Stratified Recursive Improvement Closure

Obstruction, Descent, Transport, and Evidence-Preserving Lineages

A formal mathematical theory of proof-carrying recursive self-repair

Komorebi P.

Formal theory draft, version 2.4 (Block 23 endogenous semantic-witness growth)

September 3, 2026

Status. This manuscript is a self-contained formalization of a proposed architecture for bounded recursive self-improvement. Results labeled Theorem, Proposition, Lemma, or Corollary are proved from their stated assumptions. Results labeled Conditional Theorem depend on strong architectural or statistical hypotheses. Results labeled Conjecture remain open. No claim of unrestricted, indefinitely accelerating, safe, or conscious self-improvement is made.

Abstract

We develop *Stratified Recursive Improvement Closure* (SRIC), a mathematical framework in which recursive self-improvement is represented not as unrestricted in-place rewriting of one model, but as growth of an authorized repair calculus inside an evidence-preserving branching lineage. The primitive object is a resolved lineage state containing an active stratified agent, a proof-carrying repair operad, an append-only evidence ledger, an evaluator atlas, a causal lineage graph, and a witness-and-rollback structure. The theory separates task improvement, improvement-process improvement, and epistemic improvement.

Blocks 1 through 4 establish protected evaluation contexts, intervention-relative introspection, proof-carrying repair composition, quantitative correctability and obstruction, nonlinear lifting, local-to-global repair descent, typed repair transport, gain-preservation certificates, semantic holonomy, path-flat evaluator cores, witness-separated transactions, universal and controlled recursive closure, productive recurrence, archive dynamics, foreign-agent metaproductivity, conservative grammar expansion, and causal repair-autocatalysis models. Block 5 supplies the empirical-realization contract, exact finite model-checking fixtures, a receipt-and-rollback kernel, an anonymous-change environment, and matched-budget synthetic benchmark arms. Block 6 adds intervention-defined hidden packets and controlled repair transport across learned foreign representations.

Block 7 studies two distinct one-step improvements of the repair process. In the first, parent and child receive the same five candidate repairs on every target. A dual-head child selector is compiled from development and meta-validation receipts, sealed before final audit access, and compared with its frozen parent on 120 seed/task-foreign and 120 architecture-foreign episodes. Mean protected utility rises from 0.284896 to 0.357414 and from 0.167417 to 0.272416, producing paired effects 0.072518 and 0.104999 with positive frozen bootstrap intervals. The candidate grammar and reachable correction span remain fixed, so this is behavioral selector metaproductivity rather than structural repair-image growth.

In the second realization, the descendant changes paired-anchor query design and the bridge-identification gate while the source repair, transport formula, evaluator, and deployment query budget remain fixed. On 144 primary foreign targets from representation families absent from development, the parent uses random 16-anchor queries and the child uses a compiled pivoted-QR 16-anchor policy. Mean balanced selected gain rises from 0.135212 to 0.344720. The paired effect is 0.209508 with frozen interval $[0.194968, 0.223898]$, every primary target improves, full-rank bridge identification rises from 0.03125 to 1, and harmful selected targets remain zero. A plain pivoted-QR control is effectively identical to the packet-labeled version, so packet-specific superiority is not established; the supported mechanism is deterministic rank-revealing query design.

Block 8 develops the theory required to move beyond one isolated meta-improvement. It factorizes repair processes into causal-atlas, information-acquisition, candidate-construction, transport, selection, evaluation, and witness modules. Restricted oracle envelopes define module-specific metaproductivity headroom. A permutation-average allocation separates remaining opportunity from interaction effects. Saturation certificates prevent null same-module descendants from being misread as failures of recursion. Repeated improvement is modeled as bottleneck migration in a nonnegative meta-regret system, with Lyapunov contraction inside fixed grammars and explicit discovery coordinates when a conservative extension reveals a previously unrepresented limitation.

The block also constructs a grammar-indexed obstruction persistence theory. Nested correction images have monotone residual energy, and every extension receives exact credit equal to the energy in newly correctable directions. Repair death is separated from forgetting death under information-losing semantic transport. Obstruction-guided generator discovery selects response directions in the orthogonal obstruction space, while repair-conflict hypergraphs turn incompatible validated repairs into concrete grammar-inadequacy certificates. A finite intervention-response

algebra yields a causal atlas through its minimal central projections, with gauge ambiguity and spectral-gap stability stated explicitly. Finally, summable conditional error budgets give lineage-wide control of false improvement claims across adaptive descendant generations.

The exact Block 8 companion passes eight automated tests covering headroom allocation, bottleneck migration, obstruction towers, grammar refinement, finite repair synthesis, causal-atlas decomposition, and sequential lineage budgets. Block 9 then realizes the first child–grandchild transition. The exact Block 7 rank-revealing child is replayed in a fresh regime where every primary design is full rank but paired target observations are heteroscedastically noisy. A finite lineage compiler diagnoses that rank has saturated and seals a quality-weighted D-optimal grandchild before the final audit. Under the same 24-query deployment budget, mean selected protected gain rises from 0.037393 for a recalibrated rank-child control to 0.209580 for the conditioning-aware grandchild. The paired effect is 0.172188 with frozen interval $[0.141481, 0.203110]$ across 96 fresh foreign targets. Both arms remain full rank, while the weighted information floor rises from 0.518197 to 14.268195 and clean transport NRMSE falls from 0.303981 to 0.095803.

Block 10 crosses the first bounded structural grammar-expansion threshold. The inherited conditioning-aware process remains restricted to one global affine bridge. A finite compiler certifies a persistent chart-dependent residual, conservatively embeds the global process into an invariant-context two-chart affine grammar, and seals the child before a fresh 96-target foreign audit. The strengthened global parent uses 80 bridge-fit and 16 witness pairs; the child uses 48 bridge-fit, 32 chart-selection, and the same 16 witness pairs. Child mean selected protected gain is 0.334224 with zero harmful commits. The child-parent effect is 0.334224 with frozen interval $[0.266459, 0.401903]$. A directly supervised global-affine repair oracle fitted with far more target labels reaches only 0.176694 mean gain; the child exceeds it by 0.157530 with interval $[0.080863, 0.234135]$. Mean validation NRMSE falls from 0.598513 to 0.064975. Gate-only, random-gate, and shuffled-binding controls fail, while blind local application reveals the witness's risk-utility tradeoff.

Block 11 removes the supplied Block 10 context packet. A descendant estimates two intervention-response operators, recovers a target-specific central response atlas up to block permutation and within-block gauge, and searches each recovered block for a repair-stable chart separator. Parent and child each consume 120 effect-query units. On 80 fresh foreign targets, the fixed-packet parent abstains while the endogenous-atlas child obtains mean selected protected gain 0.207325 with frozen interval $[0.131902, 0.287910]$ and zero selected harmful commits. Mean selected context overlap is 0.821461 compared with 0.322091 for the inherited fixed packet. Random-atlas, PCA, full-space, shuffled-response, blind-commit, and nonidentifiable stress controls separate the supported mechanism from coordinate, variance, and witness-free explanations.

Block 12 advances two further structural edges. First, the inherited intervention family contains an exact response alias: every polynomial or composition of its operators acts identically on two task-relevant hidden blocks. A finite compiler selects one additional intervention from a sealed candidate language, refines the central response atlas, and raises mean selected protected gain from 0 to 0.846985 under an equal 144-query envelope. Second, after the repair packet is identified, the inherited affine repair operad encounters a nonzero-Hessian obstruction. A conservative degree-two constructor raises mean selected protected gain from 0.289123 to 0.997511 under an equal 160-query envelope and exceeds a directly supervised 1600-pair same-target affine oracle by 0.530933. Duplicate, weak-separator, shuffled-response, wrong-block, random-block, shuffled-feature, full-quadratic, and stress controls identify both the supported mechanisms and their limits.

Block 13 moves the modified object one level outward. A strong proof-carrying automated-research parent compiler and a lineage-trained child receive the same structured archive, 39 candidate methods, evaluator portfolio, integrity monitor, and six full experiments per research problem. On 96 primary foreign problems, validity-filtered protected gain rises from 0.883004 to

0.950231, with paired effect 0.067227 and frozen interval $[0.035881, 0.103274]$. On 72 architecture-foreign problems, gain rises from 0.772566 to 0.863367, with paired effect 0.090800 and interval $[0.052311, 0.135081]$. Shuffled archive binding, single-benchmark selection, monitor removal, random allocation, component ablations, and an inadmissible audit oracle delimit the mechanism. Most of the realized gain is attributable to high-accuracy obstruction diagnosis and bottleneck-sensitive allocation; outcome forecasting adds little on the primary split, and the finite within-problem forum is redundant in this fixture.

Block 14 replays the Block 13 result digest and exact 39-method language, then recalibrates the inherited compiler design as the parent inside a finite adversarial stress atlas with aliased diagnostics, sparse multi-obstruction mixtures, unknown problems outside the inherited response cone, 18% archive contamination, evaluator attacks, integrity attacks, and a separately shifted problem generator. Parent and child retain the same six full experiments per problem. The robust child adds sequential response diagnosis, open-set routing, contamination-resistant archive summaries, evaluator-portfolio aggregation, and an integrity-monitor ensemble. On 144 primary problems, validity-filtered protected gain rises from 0.410813 to 0.575783, with paired effect 0.164970 and frozen interval $[0.115591, 0.218655]$. On 120 generator-foreign problems, gain rises from 0.389514 to 0.550559, with paired effect 0.161045 and interval $[0.105335, 0.223141]$. Primary lower-tail gain changes from -0.133065 to 0.271516, observed invalid selections fall from 0.027778 to 0, and false confident known-class assignment on unknown problems falls from 1 to 0.083333. Sequential response updating and open-set routing are substantively supported. The simple-monitor ablation is null and the single-portfolio ablation is nearly null, so neither receives independent positive attribution.

Block 15 freezes the robust Block 14 compiler and confronts it with persistent open-set research conflicts that cannot be resolved by its one unknown token or inherited singleton-method grammar. A finite language compiler receives a generic six-probe diagnostic language and a generic 30-member ordered pair-constructor grammar, but not the hidden identities of the two novel obstruction mechanisms. Development receipts select the probe `interaction-contrast`, create two anonymous novel tokens, and synthesize the bindings `evaluator-then-resource` and `objective-then-constructor`. The inherited 39-method grammar is preserved as an exact fallback, while the deployable child adds only the two selected composites. Parent and child receive the same eight-unit research budget. On 96 primary foreign problems, mean protected gain rises from 0.633969 to 0.859731, with paired effect 0.225762 and interval $[0.166708, 0.286431]$. On 82 generator-foreign problems, gain rises from 0.574224 to 0.814520, with paired effect 0.240296 and interval $[0.180154, 0.302701]$. Taxonomy-only, grammar-only, and shuffled-binding controls show that the supported mechanism is the coupled addition of a diagnostic distinction, a new method-constructor output, and the correct binding between them.

Block 16 makes recurrent latent reasoning a direct lineage component. It defines a Neuralese reasoner, an exact latent-trace intervention language, necessity, order, corruption, replay, cross-task specificity, adaptive-halting, and semantic-transport certificates, and a finite parent–child promotion rule. A direct adapter targets the local Cymela Hyper v1 runtime through its exposed latent-intervention hook without redistributing third-party code or weights. An engineered 66-task synthetic fixture separates a useful ordered but task-nonspecific parent from a task-specific child: structural-normalized specificity rises from 0.016952 to 0.334610, the paired transplant-cost gain is 0.439986 with frozen interval $[0.386884, 0.493143]$, and the child passes the finite Neuralese lineage gate. No empirical Hyper result is claimed because the model files were absent from the build environment.

Block 17 replaces the analytic latent-trajectory fixture by trained recurrent checkpoints in a separately implemented exact rational affine-program environment. A task-blind parent and six inherited repair candidates are optimized under a matched update and example envelope. The audit adds same-answer far-program donors to separate program semantics from answer encoding, an external held-out trace probe, adaptive-depth tests, exact replay, deterministic

noise and reversal controls, an independent NumPy forward implementation, and a signed-permutation gauge certificate. A first sealed pilot is rejected without weakening its gates. After a precommitted compiler revision and fresh final-task generation, the selected full-binding child is FULL on both 1600-task primary and shifted-generator audits while the parent remains PARTIAL. Primary and foreign program-specificity gains are 0.010714 and 0.024763, with frozen intervals $[0.009911, 0.011527]$ and $[0.022794, 0.026793]$; held-out trace-probe R^2 is 0.865504 and 0.691471. The accepted result is one finite package-local executable Neuralese binding-repair edge, not an empirical Hyper result or open-ended recursive self-improvement.

Together, Blocks 7 through 22 establish a controlled progression from one-step behavioral metaproductivity to repeated bottleneck migration, finite transport-grammar growth, endogenous response-atlas recovery, finite response-language and constructor growth, a first sealed improvement of the research compiler itself, adversarial robustness and finite language growth for that compiler, direct causal semantics for recurrent latent reasoning, one training-generated executable latent-binding repair, a cross-implementation distributional chart obstruction, a source-side conservative semantic-atlas expansion, a finite atlas-indexed cross-architecture transport connection, a type-directed expansion of the transport grammar under hidden obstruction stress, and correspondence-free transport from intervention orbits with explicit gauge and under-identification certificates. The outer repair grammar, research generators, semantic witnesses, evaluator and integrity models, donor taxonomy, architecture family, chart meta-language, gauge stress, and audit semantics remain externally declared. No unrestricted automated science, open-ended compiler self-improvement, autonomous evaluator invention, production-model Neuralese result, unrestricted safety, indefinite acceleration, consciousness, or sentience is established.

Block 18 attempts the architecture-foreign threshold. A source compiler selects a step-specific FiLM latent repair with progressive semantic and geometry witnesses on an Elman source, then instantiates the same typed schema in independently implemented GRU and gated-residual parents. Both selected children are FULL on fresh 1400-task primary audits but FAIL on an independently generated shifted support; frozen source-to-target bridges also fail, with transported $R^2 = -0.145315$ and -0.206846, while the source-reference probe itself is negative. The new constructor does not dominate inherited additive conditioning: its pooled program-specificity difference is -0.029417 with interval $[-0.034102, -0.024735]$. A post-hoc held-out diagnosis, excluded from acceptance, recovers source and target semantic probes above 0.57 and local paired bridges above 0.53. The canonical result is therefore a sealed distributional semantic-chart obstruction, not a positive cross-implementation edge. It shows that full rank, in-distribution causal specificity, and successful pre-audit alignment do not certify distributionally invariant Neuralese semantics.

Block 19 repairs the logically prior source-side chart failure. It compiles a finite affine observation atlas from semantic twins, conservatively extends four inherited charts by two pre-audit novel charts, replaces fixed-slot demonstration processing by a permutation-invariant set encoder, and separates the semantic recurrent path from a context-diagnostic path. One common ancestor and four descendants receive equal 550-update continuation envelopes. The full coupled descendant is the only pre-audit eligible candidate. On 900 fresh inherited-context tasks, clean loss falls from 0.200177 to 0.057546, same-answer program specificity rises from 0.019129 to 0.174135, and semantic-probe R^2 reaches 0.770701. On 900 fresh tasks in the added charts, clean loss falls from 0.416361 to 0.050745, program specificity reaches 0.194936, and probe R^2 reaches 0.779341. The child is `ATLAS_FULL` on both supports. Across 200 semantic objects rendered in six supported contexts, maximum trace cosine drift is 2.384×10^{-7}, maximum output drift is 1.431×10^{-6}, and halt depth is identical. All 600 nonlinear unsupported tasks are rejected. The accepted result is a finite source-side atlas-growth edge; cross-architecture atlas transport remains open.

Block 20 closes that finite cross-architecture atlas-transport gate under a declared orthogonal gauge stress. A frozen Block 19 Elman source, independently trained GRU and gated-residual

targets, and six admitted observation charts define a 3×6 architecture–context semantic bundle of complete five-step, forty-dimensional Neuralese trajectories. A sealed compiler compares pooled full-trajectory, chart-indexed step-shared, and chart-indexed full-trajectory affine connections. Only the six-chart full-trajectory family passes every pre-audit gate. On fresh inherited and added-chart supports, transported semantic-probe R^2 ranges from 0.781159 to 0.799157 for GRU and from 0.785881 to 0.796471 for the residual target. Normalized architecture–context curvature is 0.010474 and 0.008374, versus 0.131402 and 0.112397 for pooled maps. All four paired pooled-minus-selected semantic-loss intervals are strictly positive, shuffled chart binding fails sharply, and unsupported nonlinear contexts remain rejected. Thirteen tests and an independent NumPy verifier replay all 13,020 exact tasks. The result is finite controlled cross-architecture transport, not universal production-model alignment.

Block 21 makes the transport grammar itself the modified object. Above the accepted Block 20 connection, it freezes an eight-coordinate packet at each of five recurrent steps and hides six post-connection obstruction families from a minimum-description compiler: affine gauge drift, smooth invertible nonlinear warp, latent-clock mixing, aliased local charts, benign rank-changing compression, and destructive protected-semantic collision. The compiler searches a declared grammar containing step and full affine maps, a monotone nonlinear inverse, a temporal inverse, a probe-conditioned chart split, a semantic quotient, and collision-grounded no-go. All 24 pre-audit episodes and all 36 fresh final episodes select the required minimal response across GRU and gated-residual lineages. Reconstructive repairs obtain packet $R^2 \geq 0.998807$, packet NRMSE at most 0.034543, and semantic $R^2 \geq 0.718695$. Benign quotient cases retain semantic $R^2 \geq 0.999989$ without claiming full-state recovery. Destructive cases contain exact equal-observation, unequal-semantics pairs and are rejected. Nineteen tests and an independent verifier replay all 13,600 exact tasks and every compiler decision with zero metric discrepancy. The result is finite obstruction-guided transport-grammar growth, not unrestricted natural-model diagnosis.

Block 22 removes exact source–target construction pairs. Source and target latent populations are sampled independently, while only within-system intervention responses are exposed to a label-isolated compiler. A sealed response-algebra construction must distinguish five terminal claims: identified transport, identification only up to a declared gauge, partial transport on overlapping support, semantic underidentification, and deterministic no-go under destructive information loss. Across two foreign architectures and six declared regimes, all 24 pre-audit and 36 fresh final episodes issue the precommitted certificate. Twelve final episodes identify transport, six identify only a gauge class, six transport exactly the supported 70% subset with zero false admission, six return underidentification with hidden semantic disagreement at least 1.905405, and six return collision-grounded no-go with 96 exact collisions per episode. Fully identifiable cases retain semantic $R^2 \geq 0.990952$ with orbit defect at most 0.002870, while distribution-only and shuffled-binding controls fail. The result removes one finite pairing scaffold; it does not establish unrestricted unpaired alignment or endogenous semantic discovery.

Block 23 removes the supplied intermediate semantic coordinates from quotient construction. A finite compiler receives recurrent trajectories, semantic-preserving twins, anonymous intervention traces, protected future consequences, and final answers, but no gold intermediate ontology. It selects the minimum-rank quotient passing frozen predictive, interventional, answer, and twin-stability gates. Across 24 pre-audit and 36 fresh final episodes, all hidden dispositions are correct: twelve ontologies are uniquely identified, six are identified only up to gauge, six are partial, six remain underidentified despite high post-hoc decodability, and six receive collision-grounded semantic-information-loss certificates. A rank-one answer code reaches accuracy at least 0.881250 but fails future and intervention sufficiency. Audit-only donor-packet patching gives selected future $R^2 \geq 0.983162$, while answer-only, PCA, and random controls are negative. The release also supplies a source-locked adapter for a public multimode Coconut checkpoint. The external checkpoint was unavailable in the build environment and was not executed; no

real-model interpretability result is claimed.

Progressive Expansion Map

Current revision

Version 2.4 completes **Block 23: Endogenous Semantic-Witness Growth**. It removes gold intermediate semantic coordinates from quotient construction, requires witness-separated final interpretation, and distinguishes unique identification, gauge equivalence, partial ontology, protected underidentification, and destructive semantic information loss. All 36 fresh finite episodes receive the precommitted certificate. A source-locked external Coconut adapter is included, but the third-party checkpoint was not available in the build environment and no external empirical result is claimed.

Block	Focus	Expansion target
1	Foundations	Evaluation contexts, bounded improvement relations, stratified lineages, intervention-sufficient introspection, repair-operad semantics. **Completed.**
2	Correctability	Authorized correction spaces, quantitative obstruction geometry, invariant and resource constraints, nonlinear repair, exact and approximate descent, obstruction towers, and worked gluing examples. **Completed.**
3	Transport and evaluation	Exact and approximate repair transport, compositional error bounds, authorization and obstruction faithfulness, gain certificates, repair holonomy, evaluator atlases, spectral path-flat cores, adequacy, and evaluator-upgrade protocols. **Completed.**
4	Recursive closure	Witness-separated transactions, commit-reveal semantics, robust and productive fixed points, archives, foreign-agent metaproductivity, conservative grammar growth, and autocatalytic repair discovery. **Completed.**
5	Empirical realization	Abstract-runtime refinement, finite model checking, executable receipts and rollback, anonymous-change protocols, matched-budget benchmarks, causal ablations, and reproducible packages. **Completed.**
6	Learned-agent validation	Intervention-defined hidden packets, paired-anchor bridge learning, foreign representation families, witness-gated repair transfer, learned encoders, causal controls, and frozen level-4 claim gates. **Completed.**

Block	Focus	Expansion target
7	Sealed metaproductivity	Two one-step parent–child comparisons: a dual-head selector over identical repair bundles, and a compiled rank-revealing anchor-query policy under an identical deployment budget; split-file sealing, paired foreign audits, causal controls, and fixed claim boundaries. **Completed.**
8	Headroom and bottleneck migration	Restricted oracle envelopes, Shapley meta-headroom, saturation, bottleneck migration, obstruction persistence, grammar-conflict refinement, causal-atlas discovery, sequential lineage certification, and exact symbolic fixtures. **Completed.**
9	Repeated lineage	Exact ancestor replay, rank-saturation diagnosis, heteroscedastic weighted bridge information, conditioning-aware query compilation, fresh child–grandchild audit, causal controls, and a package-closed two-edge certificate. **Completed.**
10	Structural grammar expansion	Persistent global-grammar residual, conservative two-chart bridge and repair constructor, finite separator discovery, structural contrast certificate, matched foreign audit, causal controls, and package-closed structural metaproductivity. **Completed.**
11	Endogenous atlas growth	Supplied-packet removal, intervention-response algebra, central projector recovery, repair-relevant block labeling, conservative fallback, matched foreign audit, causal controls, and package-closed finite atlas growth. **Completed.**
12	Response-language and constructor growth	Exact response-alias certificate, finite separating-intervention compiler, conservative response-algebra refinement, affine function-class obstruction, degree-two repair-constructor expansion, matched foreign audits, causal controls, and package-closed replay. **Completed.**
13	Research-compiler metaproductivity	Results-free proposal receipts, proof-carrying finding graph, multi-benchmark headroom evaluation, integrity-valid selection, obstruction diagnosis, outcome forecasting, matched-budget parent–child compiler audit, mechanism ablations, and package-closed replay. **Completed.**

Block	Focus	Expansion target
14	Adversarial compiler robustness	Diagnostic aliasing, sparse multi-obstruction inference, open-set residuals, contaminated archives, evaluator and integrity attacks, sequential diagnosis, robust lower-tail and worst-group gates, generator-foreign audit, and package-closed replay. **Completed.**
15	Compiler-language growth	Open-set repair-conflict certificate, generic separating-probe search, anonymous diagnostic-token refinement, ordered pair-method synthesis, conservative inherited fallback, matched-budget foreign audits, causal language controls, and package-closed replay. **Completed.**
16	Direct Neuralese integration	Recurrent latent-reasoner object, exact injection-trace interventions, necessity, order, corruption, replay, cross-task specificity, halting and semantic-transport certificates, a local Hyper adapter, and a synthetic parent–child lineage fixture. **Direct contract and synthetic realization completed; real-model audit pending.**
17	Executable Neuralese repair compilation	Exact affine-program execution, trained recurrent parent and six matched-update descendants, answer-confound controls, external trace decoding, adaptive halting, preserved failed pilot, pre-committed compiler repair, fresh primary and shifted audits, NumPy shadow replay, gauge verification, and package-closed receipts. **Finite training-generated edge completed; foreign production-model audit remains open.**
18	Cross-implementation Neuralese transport	Exact multi-template program induction, Elman source compiler, GRU and gated-residual foreign parents, step-specific FiLM schema transport, causal donor audits, frozen latent bridges, shifted-support failure, and post-hoc semantic-chart localization. **Sealed distributional obstruction completed; positive foreign-model edge not established.**

Block	Focus	Expansion target
19	Distribution-indexed Neuralese atlas growth	Semantic-twin chart compilation, conservative four-to-six-chart extension, permutation-invariant set encoding, semantic/context path separation, selective support admission, fresh inherited and novel-chart audits, unsupported nonlinear controls, and independent NumPy replay. **Finite source-side atlas-growth edge completed; cross-architecture atlas transport remains open.**
20	Atlas-indexed cross-architecture Neuralese transport	Frozen six-chart source atlas, independently trained GRU and gated-residual targets, chart-dependent orthogonal gauge stress, pooled and chart-indexed trajectory bridges, architecture–context commutation audits, shuffled-binding and unsupported controls, paired fresh audits, and independent NumPy replay. **Finite controlled cross-architecture atlas-transport edge completed; nonlinear connection growth remains open.**
21	Obstruction-guided Neuralese transport-grammar growth	Frozen five-step diagnostic packet, six hidden post-connection obstruction families, minimum-description affine, nonlinear, temporal, chart-split, quotient, and no-go constructors, fresh cross-architecture audits, wrong-repair controls, exact collision certificates, and independent replay. **Finite type-directed transport-grammar edge completed; unpaired and ontology-light transport remains open.**
22	Correspondence-free intervention-orbit Neuralese transport	Independent source and target populations, within-system intervention responses, anonymous action matching, response-algebra intertwiners, gauge-equivalence certificates, selective partial transport, semantic underidentification, collision-grounded no-go, sealed final coupling reveal, tests, and deterministic replay. **Finite unpaired transport edge completed; endogenous-semantic and stochastic-path transport remain open.**

Block	Focus	Expansion target
23	Endogenous semantic-witness growth	Gold-free causal-predictive quotient construction from semantic twins, anonymous interventions, protected futures, and answers; minimum-rank selection; answer-shortcut rejection; gauge, partiality, underidentification, collision no-go, counterfactual packet controls, and a source-locked external Coconut pilot. **Finite ontology-growth edge completed; real checkpoint execution remains pending.**

Contents

Progressive Expansion Map **vi**

1 Introduction **1**
1.1 Why the usual picture is underspecified . . . 1
1.2 Central proposal . . . 1
1.3 Design principles . . . 2
1.4 What counts as a substantive recursive step . . . 2
1.5 Status discipline . . . 3

2 Evaluation Contexts and Improvement Relations **5**
2.1 Protected evaluation contexts . . . 5
2.2 Three levels of improvement . . . 6
2.3 The evaluator paradox . . . 7
2.4 Nontriviality and external anchoring . . . 8

3 Resolved Lineages and Stratified Agents **8**
3.1 Functional atlases . . . 8
3.2 Approximate property transport . . . 9
3.3 The resolved lineage state . . . 10
3.4 Lineage receipts and continuation . . . 11
3.5 Forks and symmetry . . . 12
3.6 Resolution consistency . . . 12

4 Relative Introspection **12**
4.1 Intervention semantics . . . 12
4.2 Decision regret and ranking competence . . . 13
4.3 Coverage, expansion, and misspecification . . . 14
4.4 Identifiability and active probing . . . 15
4.5 Prospective causal closure . . . 16
4.6 Self-confirmation failure . . . 16

5 Proof-Carrying Repair Operads **16**
5.1 From repair libraries to typed grammars . . . 16
5.2 Syntactic composition and semantic authorization . . . 17
5.3 Sequential and parallel repair . . . 18
5.4 Conservative extensions . . . 19
5.5 Repair abstraction and generality . . . 20
5.6 Repair equivalence under exact transport . . . 20
5.7 Meta-repair . . . 21
5.8 Example: a typed verification repair . . . 21

6 Authorized Correction Geometry and Obstruction Calculus 22
6.1 Linear correction and quotient obstruction . . . 22
6.2 Canonical correctable and obstructed components . . . 22
6.3 Protected invariants and resource-aware correction . . . 24
6.4 Convex and one-sided authorization . . . 25
6.5 Sensitivity, conditioning, and empirical certificates . . . 26
6.6 Obstruction shrinkage under repair expansion . . . 27
6.7 Changing defect spaces . . . 28
6.8 Packet-resolved correction and cross-packet leakage . . . 28
6.9 Gauge-fixed nonlinear local repair by contraction . . . 29
6.10 Newton–Kantorovich repair . . . 31
6.11 Singular nonlinear repair and Lyapunov–Schmidt reduction . . . 32
6.12 Higher-order obstruction classes . . . 33

7 Local-to-Global Repair Descent 34
7.1 Repair sheaves and local solution torsors . . . 34
7.2 Exact linear descent . . . 34
7.3 Graph-Hodge decomposition of approximate gluing . . . 35
7.4 Weighted overlap confidence . . . 37
7.5 Finite cycle obstruction . . . 37
7.6 Nonlinear descent through symmetry torsors . . . 38
7.7 Spanning-tree holonomy test . . . 38
7.8 Order-by-order global repair . . . 39
7.9 Cover refinement and diagnostic soundness . . . 40

8 Repair Transport and Transfer Geometry 40
8.1 Correction data and transport morphisms . . . 41
8.2 Authorization, invariants, and minimum-cost correction . . . 43
8.3 Composition and accumulation of transport defect . . . 44
8.4 Repair holonomy and path dependence . . . 46
8.5 Approximate transported residual . . . 47
8.6 Deterministic gain preservation . . . 47
8.7 Distributional and probabilistic transfer certificates . . . 49

9 Path-Flat Evaluator Evolution 50
9.1 Evaluator objects and morphisms . . . 50
9.2 Evaluator atlases and exact holonomy . . . 51
9.3 Approximate path-flat cores . . . 53
9.4 Semantic holonomy and decision stability . . . 54
9.5 Flatness is not sufficiency . . . 55
9.6 Conservative evaluator upgrades . . . 56
9.7 Statistical certification of evaluator upgrades . . . 57
9.8 Proof-carrying evaluator-upgrade protocol . . . 58
9.9 Examples of evaluator holonomy . . . 59

10 Witness-Separated Improvement Transactions **59**
10.1 Transaction objects and immutable surfaces . . . 59
10.2 Rollback, atomicity, and serial composition . . . 61
10.3 Commit-reveal discipline and adaptive validity . . . 62
10.4 Rotating witnesses . . . 63

11 Recursive Improvement Closure **64**
11.1 Universal finite-state closure . . . 65
11.2 Controlled robust closure . . . 65
11.3 Safe recurrence is not productive recurrence . . . 66
11.4 Chance-constrained closure . . . 67
11.5 Closure certificates and counterexamples . . . 67

12 Archive Dynamics and Calibrated Selection **68**
12.1 Branching archives and multiobjective frontiers . . . 68
12.2 Finite approximate frontier compression . . . 68
12.3 Stepping stones and continuation signatures . . . 69
12.4 Calibrated acceptance and archive monotonicity . . . 70

13 Metaproductivity **70**
13.1 Behavioral metaproductivity on foreign agents . . . 70
13.2 The best-of-k budget confound . . . 71
13.3 Paired foreign-agent estimation and distribution shift . . . 71
13.4 Task-relevant structural metaproductivity . . . 72
13.5 Conservative meta-improvement . . . 73

14 Grammar Expansion and Open-Endedness **74**
14.1 Finite fixed-grammar exhaustion . . . 75
14.2 Conservative grammar extensions . . . 75
14.3 Directed grammar limits . . . 75
14.4 Grammar-boundary certificates . . . 76
14.5 Dynamic obstruction exact sequence . . . 77
14.6 Representation-separation certificates . . . 77
14.7 Levels of improvement . . . 78

15 Autocatalytic Repair Discovery **78**
15.1 Repair-discovery closure . . . 78
15.2 Causal offspring matrix . . . 79
15.3 Autocatalytic cores . . . 79
15.4 Robust spectral certification . . . 80
15.5 Resource normalization and time variation . . . 80
15.6 Saturation and the distinction from utility . . . 81

16 Empirical Realization Contract **82**
16.1 Claim ladder . . . 82
16.2 Benchmark instances . . . 82
16.3 Abstract-runtime refinement . . . 83
16.4 Frozen endpoints and claim gates . . . 84

17 Finite Model-Checking Realization **85**
17.1 Four frozen cases . . . 85
17.2 Classifier and completeness on the fixture . . . 87
17.3 Finite closure game . . . 87
17.4 Frozen finite endpoints . . . 88
17.5 Reference execution . . . 88

18 Executable Transaction and Receipt Kernel **88**
18.1 Runtime types . . . 88
18.2 Phase machine . . . 89
18.3 Canonical receipts and hash chaining . . . 90
18.4 Reference attack pair . . . 90
18.5 Reference module boundary . . . 90

19 Anonymous Continuous Repair Environment **91**
19.1 Linear environment family . . . 91
19.2 Bounded probe vector . . . 92
19.3 Posterior attribution . . . 92
19.4 Typed repair responses . . . 93
19.5 Reference positive-control execution . . . 94

20 Matched-Budget Recursive Repair Benchmark **94**
20.1 Foreign synthetic agents . . . 94
20.2 Matched arms . . . 95
20.3 Search and evaluator confounds . . . 95
20.4 Transport positive control . . . 96
20.5 Structural metric . . . 96
20.6 Reference benchmark execution . . . 96
20.7 Causal abstraction contrast . . . 97

21 Causal Ablation and Statistical Inference **97**
21.1 Factorized architecture . . . 97
21.2 Decisive schema-abstraction ablation . . . 98
21.3 Paired estimation . . . 98
21.4 Multiple claims and alpha spending . . . 98
21.5 Required negative controls . . . 99
21.6 Primary Block 5 gates . . . 99

22 Reproducibility, Packaging, and Verification **100**
22.1 Package layout . . . 100
22.2 Deterministic and stochastic replay . . . 100
22.3 Verification commands . . . 101
22.4 Reference verification summary . . . 101
22.5 Reproducibility tiers . . . 101

23 Learned-Agent Validation Contract **102**
23.1 Representation-level validation object . . . 102
23.2 Data separation . . . 103
23.3 Block 6 claim target . . . 103

24 Controlled Causal-Packet Fixture **104**
24.1 Task and defect construction . . . 104
24.2 Causal counterfactual probe . . . 104
24.3 Packet-local repair fitting . . . 105
24.4 Packet-specificity result . . . 106
24.5 Wrong-block control . . . 107

25 Learned Cross-Agent Repair Transport **107**
25.1 Learning the hidden-state bridge . . . 107
25.2 Exact injective transport . . . 108
25.3 Compression obstruction . . . 108
25.4 Approximate transport decomposition . . . 109
25.5 Anchor identifiability . . . 109

26 Foreign Representation Families **109**
26.1 Controlled target atlas . . . 109
26.2 Control arms . . . 110
26.3 Family results . . . 110
26.4 Shuffled-anchor and direct-copy controls . . . 111
26.5 Anchor-budget sweep . . . 112

27 Witness-Gated Repair Commitment **112**
27.1 Frozen gate . . . 112
27.2 Why the gate rejects apparently useful repairs . . . 113

28 Independently Trained Representation Pilot **113**
28.1 Training objective . . . 113
28.2 Pilot results . . . 114
28.3 Interpretation . . . 115

29 Causal Controls and Statistical Interpretation **115**
29.1 Factorized causal evidence . . . 115
29.2 Bootstrap intervals . . . 115
29.3 Claim-gate ledger . . . 116

30 Block 6 Reproducibility and Package **116**
30.1 Reference package . . . 116
30.2 Verification commands . . . 117
30.3 Frozen endpoint set . . . 117

31 Sealed Parent–Child Repair-Process Improvement **118**
31.1 Target episodes and candidate bundles . . . 118
31.2 The sealed split discipline . . . 119

32 A Proof-Carrying Dual-Head Descendant **121**
32.1 Candidate features . . . 121
32.2 Utility and risk heads . . . 121
32.3 Finite meta-grammar selection . . . 123
32.4 Selected descendant receipt . . . 123

33 Matched-Budget Attribution of the Meta-Gain **124**
33.1 The best-of-k confound . . . 124
33.2 Lineage and semantic controls . . . 125
33.3 Selection versus structural correctability . . . 125

34 Protected Utility, Risk, and Statistical Certification **126**
34.1 Protected target utility . . . 126
34.2 Paired inference . . . 126
34.3 Cumulative claim gates . . . 127
34.4 Risk-head ablation . . . 127

35 Executable Block 7 Benchmark **128**
35.1 Task and defect generator . . . 128
35.2 Representation families . . . 128
35.3 Five-candidate repair grammar . . . 128
35.4 Episode counts and lineage chronology . . . 129
35.5 Comparison arms . . . 129

36 Block 7 Results **130**
36.1 Primary parent–child comparison . . . 130
36.2 Strong-baseline and oracle comparisons . . . 131
36.3 Risk-utility tradeoff . . . 132
36.4 Lineage and ablation controls . . . 132
36.5 Frozen claim-gate ledger . . . 133

37 Formal Interpretation of the Block 7 Edge **134**
37.1 What changed in the child . . . 135
37.2 What did not change . . . 135
37.3 Claim-ladder advancement . . . 135

38 Block 7 Reproducibility and Package **136**
38.1 Package layout . . . 136
38.2 Canonical replay . . . 137
38.3 Frozen verification summary . . . 137
38.4 Remaining reproducibility limitations . . . 138

39 Sealed Parent–Child Bridge-Policy Compiler **138**
39.1 Repair processes as objects of comparison . . . 139
39.2 Development, seal, and audit separation . . . 140
39.3 Foreignness and matched resources . . . 140

40 Meta-Repair of Bridge Identification **141**
40.1 The parent failure . . . 141
40.2 Bridge identifiability . . . 141
40.3 Conditional-packet rank obstruction . . . 142
40.4 Packet-balanced pivoted QR . . . 142
40.5 Meta-grammar and compiler . . . 143
40.6 Lineage-permutation control . . . 144

41 One-Step Bounded Metaproductivity Experiment **144**
41.1 Frozen experimental design . . . 144
41.2 Compared arms . . . 145
41.3 Primary metaproductivity result . . . 145
41.4 Structural metaproductivity . . . 147
41.5 Family-resolved audit results . . . 148
41.6 Query efficiency and meta-compilation cost . . . 150
41.7 Strong-baseline interpretation . . . 150

42 Bridge-Policy Causal Interpretation and Claim Boundary **151**
42.1 What changed . . . 151
42.2 A finite one-step certificate . . . 151
42.3 What the result does not establish . . . 152

43 Bridge-Policy Reproducibility and Package **152**
43.1 Reference package . . . 152
43.2 Frozen execution . . . 153
43.3 Frozen claim gates . . . 154
43.4 Reproduction commands . . . 154

44 Meta-Headroom Geometry **154**
44.1 Repair-process factorization . . . 154
44.2 Restricted oracle envelopes . . . 155
44.3 Ordered meta-regret decomposition . . . 156
44.4 Interaction-independent headroom allocation . . . 156
44.5 Component saturation and headroom-adjusted gain . . . 157

45 Bottleneck Migration and Repeated Lineage Dynamics **158**
45.1 The meta-regret state . . . 159
45.2 Lyapunov contraction . . . 159
45.3 Greedy dominant-bottleneck contraction . . . 160
45.4 Saturation-aware controller . . . 160
45.5 Conservative grammar discovery event . . . 161
45.6 Conservative lineage dominance . . . 162
45.7 Repair process and repair compiler . . . 163

46 Persistent Obstruction Across Grammar Growth **163**
46.1 Nested correction images . . . 163
46.2 Exact newly correctable energy . . . 164
46.3 Dynamic defect semantics . . . 165
46.4 Repair death and forgetting death . . . 166
46.5 Approximate grammar transport and semantic debt . . . 166

47 Obstruction-Guided Generator and Grammar Discovery **167**
47.1 One-generator obstruction reduction . . . 167
47.2 Distributional rank-k extension . . . 168
47.3 Causal credit for a new generator . . . 168
47.4 Repair-conflict grammar inadequacy . . . 169
47.5 Feature acquisition as set cover . . . 170
47.6 Candidate invention as a right inverse . . . 170
47.7 Finite counterexample-guided repair synthesis . . . 171

48 Causal Atlas Discovery from Intervention Algebras **172**
48.1 Intervention-response algebra . . . 172
48.2 Finite causal atlas theorem . . . 172
48.3 Defect-specific cyclic modules . . . 173
48.4 Approximate commutant recovery . . . 173
48.5 Finite algebra fixture . . . 174
48.6 Nonlinear local extension . . . 174

49 Sequential Lineage Certification **175**
49.1 Filtration and edge-level effects . . . 175
49.2 Branching archive allocation . . . 175
49.3 Protected regression and semantic-debt budgets . . . 176
49.4 Repeated-lineage metaproductivity certificate . . . 176
49.5 Conditional bottleneck-migrating improvement theorem . . . 177

50 Block 8 Symbolic Reference Realization **178**
50.1 Implemented fixtures . . . 178
50.2 Frozen exact results . . . 178
50.3 Interpretation boundary . . . 179
50.4 Next empirical lineage . . . 179

51 Block 8 Reproducibility and Package **179**
51.1 Package layout . . . 179
51.2 Canonical replay . . . 180
51.3 Claim boundary . . . 180

52 Repeated Lineage under a Migrating Bottleneck **180**
52.1 Context-indexed repeated lineage . . . 181
52.2 Rank saturation and bottleneck migration . . . 182

53 Heteroscedastic Bridge Information **182**
53.1 Weighted paired-anchor model . . . 183
53.2 Ridge regularization and reported information . . . 184
53.3 Transport error and repair error . . . 184
53.4 Known quality and the autonomy boundary . . . 185

54 Conditioning-Aware Query Design **185**
54.1 D-optimal marginal information . . . 185
54.2 A-optimal marginal gain . . . 185
54.3 E-optimal exchange . . . 186
54.4 Quality-binding control . . . 186
54.5 Rank-only and gate-only controls . . . 186
54.6 Conditioning diagnosis . . . 186

55 Sealed Child-to-Grandchild Protocol **187**
55.1 Lineage objects . . . 187
55.2 Fresh task and representation supports . . . 187
55.3 Fixed query and repair budget . . . 188
55.4 Finite meta-grammar . . . 188
55.5 Commit gate . . . 188
55.6 Seal chronology . . . 189
55.7 Finite-class selection bound . . . 189
55.8 Paired grandchild certificate . . . 190

56 Block 9 Executable Realization and Results **190**
56.1 Development diagnosis . . . 190
56.2 Primary foreign audit . . . 190
56.3 Bottleneck migration is structural . . . 193
56.4 Gate isolation and quality binding . . . 194
56.5 Larger-budget control . . . 195
56.6 Family-level results and negative boundaries . . . 196
56.7 Two-edge repeated-lineage certificate . . . 196

57 Block 9 Reproducibility and Package **197**
57.1 Package contents . . . 197
57.2 Canonical replay . . . 198
57.3 Claim boundary . . . 198

58 Structural Grammar Expansion by Local Transport Charts **199**
58.1 Affine global and chart-resolved grammars . . . 199
58.2 Global-affine obstruction . . . 200
58.3 Chartwise repair transport . . . 201
58.4 Finite separator discovery . . . 203
58.5 Conservative fallback . . . 204

59 Block 10 Experimental Contract **204**
59.1 Lineage edge and mutable grammar . . . 204
59.2 Source state, protected packet, and repair . . . 205
59.3 Piecewise foreign target atlas . . . 205
59.4 Inherited quality-aware acquisition . . . 206
59.5 Protected gain and witness quantities . . . 206
59.6 Finite compilation and pre-audit sealing . . . 206
59.7 Frozen controls . . . 207

60 Block 10 Results **207**
60.1 Development grammar-inadequacy certificate . . . 207
60.2 Primary foreign audit . . . 208
60.3 Foreign-family results . . . 210
60.4 Structural diagnostics . . . 212
60.5 Mechanism controls . . . 213
60.6 Stress boundaries . . . 213
60.7 Finite structural grammar-expansion theorem . . . 214

61 Block 10 Reproducibility and Package **215**
61.1 Package contents . . . 215
61.2 Canonical replay . . . 215
61.3 Claim boundary . . . 216

62 Endogenous Atlas Growth from Intervention Responses **216**
62.1 Intervention-response algebras . . . 217
62.2 Commutants and unresolved multiplicity . . . 218
62.3 Approximate response atlases . . . 219
62.4 From an unlabeled atlas to a repair-relevant chart packet . . . 220
62.5 Conservative relation to the supplied-packet grammar . . . 221
62.6 Cyclic defect modules and atlas localization . . . 222

63 Block 11 Experimental Contract **222**
63.1 Lineage and withheld structure . . . 222
63.2 Query accounting . . . 223
63.3 Data chronology and sealing . . . 223
63.4 Comparison arms . . . 223
63.5 Frozen witness . . . 224
63.6 Frozen claim gates . . . 224

64 Block 11 Results **225**
64.1 Development atlas certificate . . . 225
64.2 Primary paired foreign audit . . . 226
64.3 Atlas geometry . . . 228
64.4 Family-level behavior . . . 229
64.5 Mechanism controls and witness contribution . . . 230
64.6 Stress boundaries . . . 231
64.7 Finite endogenous-atlas realization theorem . . . 231

65 Block 11 Reproducibility and Package **232**
65.1 Reference implementation . . . 232
65.2 Claim boundary . . . 233

66 Endogenous Response-Language and Candidate-Constructor Growth **234**
66.1 Response languages and observational aliases . . . 234
66.2 Separating interventions and conservative response refinement . . . 235
66.3 Approximate spectral recovery . . . 236
66.4 Finite response-language compilation . . . 236

67 Repair-Constructor Expansion **237**
67.1 Affine repair operad and its closure . . . 237
67.2 A Hessian obstruction to affine repair . . . 237
67.3 Block-local quadratic repair grammar . . . 238
67.4 Constructor novelty and parsimony . . . 238

68 Block 12 Experimental Contract **239**
68.1 Two separately sealed contexts . . . 239
68.2 Response-language fixture . . . 239
68.3 Matched response-query envelope . . . 240
68.4 Constructor fixture and matched budget . . . 240
68.5 Chronology and seal . . . 241
68.6 Frozen comparison arms . . . 241

69 Block 12 Results **242**
69.1 Response-language development certificate . . . 242
69.2 Primary response-language audit . . . 242
69.3 Response geometry and controls . . . 244
69.4 Constructor development certificate . . . 245
69.5 Primary constructor audit . . . 245
69.6 Constructor controls and stress boundary . . . 247
69.7 Two finite structural lineage theorems . . . 247

70 Block 12 Reproducibility and Package **249**
70.1 Reference implementation . . . 249
70.2 Claim boundary . . . 250

71 Proof-Carrying Automated Research Compilers **251**
71.1 Research problems and compiler objects . . . 252
71.2 Results-free proposal receipts . . . 252
71.3 Finding graphs and research lineage . . . 253
71.4 Evaluator portfolios and metric-gaming nullspaces . . . 253
71.5 Integrity-valid compiler metaproductivity . . . 254
71.6 Outcome forecasting and cost-sensitive allocation . . . 255
71.7 Sealed compiler comparison . . . 255

72 Block 13 Experimental Contract **256**
72.1 Research-problem family . . . 256
72.2 Candidate grammar . . . 256
72.3 Parent compiler . . . 257
72.4 Child compiler . . . 257
72.5 Splits and chronology . . . 257
72.6 Matched resource envelope . . . 258
72.7 Comparison arms . . . 258
72.8 Frozen claim gates . . . 258

73 Block 13 Results **259**
73.1 Primary and architecture-foreign compiler effects . . . 259
73.2 Performance by obstruction class . . . 261
73.3 Allocation shift . . . 261
73.4 Forecast and classification diagnostics . . . 262
73.5 Archive, evaluator, and integrity controls . . . 264
73.6 Finite compiler-metaproductivity theorem . . . 264

74 Block 13 Reproducibility and Package **265**
74.1 Reference implementation . . . 265
74.2 Claim boundary . . . 266

75 Adversarial Open-Set Research-Compiler Robustness **267**
75.1 Robustness environment . . . 267
75.2 Diagnostic aliasing and active separation . . . 269
75.3 Open-set residuals . . . 270
75.4 Archive contamination . . . 270
75.5 Evaluator portfolios and integrity ensembles . . . 271

76 Block 14 Experimental Contract **272**
76.1 Inherited parent and robust child . . . 272
76.2 Problem groups . . . 272
76.3 Archive contamination and candidate grammar . . . 273
76.4 Matched research budget and chronology . . . 273
76.5 Comparison arms . . . 273
76.6 Frozen claim gates . . . 274

77 Block 14 Results **274**
77.1 Primary and generator-foreign effects . . . 274
77.2 Group-resolved robustness . . . 276
77.3 Lower-tail and failure-rate behavior . . . 277
77.4 Mechanism ablations . . . 279
77.5 Contamination sweep . . . 280
77.6 Finite adversarial robustness theorem . . . 280

78 Block 14 Reproducibility and Claim Boundary **281**
78.1 Reference implementation . . . 281
78.2 Claim boundary . . . 282

79 Compiler-Language Growth from Open-Set Research Conflicts **282**
79.1 Compiler languages and inherited quotients . . . 283
79.2 Separating research experiments . . . 284
79.3 A generic composite-method constructor . . . 285
79.4 Finite selection and proof-carrying certificates . . . 286

80 Block 15 Experimental Contract **286**
80.1 Lineage edge and withheld structure . . . 286
80.2 Development conflict and language compiler . . . 287
80.3 Matched deployment budget . . . 287
80.4 Splits and chronology . . . 288
80.5 Comparison arms . . . 288
80.6 Frozen claim gates . . . 288

81 Block 15 Results **289**
81.1 Language-inadequacy certificate . . . 289
81.2 Primary and generator-foreign compiler effects . . . 290
81.3 Novel-obstruction performance . . . 292
81.4 Taxonomy, constructor, and binding controls . . . 292
81.5 Conservative inheritance and safety interpretation . . . 293
81.6 Finite compiler-language realization theorem . . . 293

82 Block 15 Reproducibility and Claim Boundary **294**
82.1 Reference implementation . . . 294
82.2 Claim boundary . . . 295

83 Direct Neuralese Causal Semantics **295**
83.1 Neuralese reasoners as typed lineage objects . . . 296
83.2 The direct intervention language . . . 297
83.3 Adaptive halting as a separate claim . . . 299
83.4 Functional packets and gauge-invariant Neuralese semantics . . . 299
83.5 The direct Neuralese lineage edge . . . 300
83.6 Executable mapping to Hyper v1 . . . 301
83.7 Finite synthetic realization . . . 301

83.8 Hyper audit chronology . . . 303
83.9 Claim boundary and next threshold . . . 303

84 Executable Neuralese Binding-Repair Compilation 304
84.1 Exact executable research environment . . . 305
84.2 The recurrent parent and repair grammar . . . 306
84.3 Why one transplant statistic is insufficient . . . 307
84.4 Executable causal and semantic gates . . . 308
84.5 Sealed compiler chronology . . . 309
84.6 A failed sealed pilot and nonretroactive repair . . . 309
84.7 Compiler result . . . 310
84.8 Fresh primary and generator-foreign audits . . . 311
84.9 Independent shadow and gauge verification . . . 314
84.10 Finite executable Neuralese repair theorem . . . 315
84.11 Reproducibility and claim boundary . . . 316

85 Cross-Implementation Neuralese Repair Transport 317
85.1 Exact externally executed program-induction environment . . . 318
85.2 Source and foreign recurrent implementations . . . 319
85.3 Source compiler and pre-audit chronology . . . 320
85.4 Causal, semantic, and execution audit . . . 321
85.5 Latent semantic bridges . . . 322
85.6 Distributional transport theory . . . 322
85.7 Sealed final results . . . 323
85.8 Post-hoc localization without retroactive promotion . . . 326
85.9 Reproducibility and claim boundary . . . 328

86 Distribution-Indexed Neuralese Semantic Atlas Growth 330
86.1 Semantic bundles, twins, and local charts . . . 330
86.2 Exact chart recovery and conservative growth . . . 331
86.3 Why nonlinear contexts require abstention . . . 333
86.4 Set-structured Neuralese semantic bundle . . . 334
86.5 Exact program environment and split discipline . . . 335
86.6 Finite candidate compiler . . . 336
86.7 Fresh final results . . . 337
86.8 Selective admission and mechanism controls . . . 339
86.9 Finite realization theorem . . . 341
86.10 Reproducibility and independent verification . . . 341
86.11 Claim boundary and next threshold . . . 342

87 Atlas-Indexed Cross-Architecture Neuralese Transport 343
87.1 Architecture–context semantic bundle . . . 344
87.2 Why one pooled bridge is not generally sufficient . . . 345
87.3 Full-trajectory transport and temporal structure . . . 346
87.4 Executable environment and frozen ancestry . . . 347
87.5 Controlled architecture-chart gauge stress . . . 348

87.6 Compiler chronology and acceptance rule . . . 348
87.7 Pre-audit compiler result . . . 348
87.8 Fresh final semantic-transport results . . . 350
87.9 Paired effects . . . 352
87.10 Architecture–context curvature result . . . 352
87.11 Mechanism controls . . . 353
87.12 Finite transport realization theorem . . . 354
87.13 Reproducibility and independent verification . . . 355
87.14 Claim boundary and next threshold . . . 355

88 Obstruction-Guided Neuralese Transport-Grammar Growth 357
88.1 Inherited connection and protected packet . . . 357
88.2 Finite obstruction families . . . 358
88.2.1 Affine gauge drift . . . 358
88.2.2 Smooth nonlinear warp . . . 358
88.2.3 Latent-clock mixing . . . 359
88.2.4 Aliased observation charts . . . 359
88.2.5 Semantics-preserving compression . . . 360
88.2.6 Destructive semantic collision . . . 360
88.3 Repair grammar and compiler . . . 361
88.4 Sealed experimental chronology . . . 362
88.5 Final obstruction-to-repair result . . . 363
88.6 Wrong-repair and no-go controls . . . 364
88.7 Finite realization theorem . . . 366
88.8 Claim boundary and next threshold . . . 366

89 Integrated SRIC Architecture 367
89.1 Functional loop . . . 367
89.2 One transaction . . . 367

90 Empirical Status and Remaining Program 368
90.1 Realized in Block 5 . . . 368
90.2 Realized in Block 6 . . . 369
90.3 Realized in Block 7 . . . 369
90.4 Realized in Block 8 . . . 370
90.5 Realized in Block 9 . . . 371
90.6 Realized in Block 10 . . . 372
90.7 Current empirical interpretation . . . 372
90.8 Realized in Block 11 . . . 373
90.9 Realized in Block 12 . . . 374
90.10 Realized in Block 13 . . . 374
90.11 Realized in Block 14 . . . 375
90.12 Realized in Block 15 . . . 376
90.13 Stage H completed: first repeated lineage . . . 377
90.14 Stage I partially realized: finite structural grammar expansion . . . 377
90.15 Stage J partially realized: finite endogenous atlas growth . . . 377

90.16 Stage K completed in finite form: response-language and candidate-constructor growth . . . 378
90.17 Stage L partially realized: finite research-compiler metaproductivity . . . 378
90.18 Stage M completed in finite form: adversarial open-set compiler robustness . . . 378
90.19 Stage N completed in finite form: compiler-language growth . . . 379
90.20 Stage O direct integration completed: Neuralese causal semantics . . . 379
90.21 Stage P completed in finite form: executable Neuralese binding repair . . . 379
90.22 Stage Q attempted and obstruction-certified: cross-implementation Neuralese transport . . . 380
90.23 Stage R completed in finite form: distribution-indexed source atlas growth . . . 380
90.24 Stage S completed in finite form: atlas-indexed cross-architecture transport . . . 380
90.25 Stage T completed in finite form: obstruction-guided transport-grammar growth . 381
90.26 Precommitted next-stage estimands . . . 381

91 Conditional Stratified Recursive Improvement Theorem 382

92 Failure Modes and Falsification Criteria 388

93 Limitations 395
93.1 Finite models and declared boundaries . . . 395
93.2 Transaction isolation is conditional . . . 395
93.3 Statistical adaptivity . . . 395
93.4 Self-model and grammar misspecification . . . 395
93.5 Evaluator dependence . . . 395
93.6 Autocatalytic approximation . . . 396
93.7 Synthetic and controlled reference realizations . . . 396
93.8 Controlled learned representations . . . 396
93.9 Witness incompleteness . . . 396
93.10 Fixed candidate and feature grammar in Block 7 . . . 396
93.11 One-generation evidence . . . 396
93.12 Meta-selection and uncertainty . . . 397
93.13 Package closure and side effects . . . 397
93.14 Finite-sample and seed dependence . . . 397
93.15 Block 8 oracle and factorization dependence . . . 397
93.16 Symbolic grammar and atlas realization . . . 397
93.17 Bottleneck model approximation . . . 397
93.18 Block 9 quality metadata and finite design grammar . . . 397
93.19 Context-indexed rather than scalar transitivity . . . 398
93.20 Synthetic heteroscedastic bridge regime . . . 398
93.21 Compiler remains externally fixed . . . 398
93.22 Finite protected-context chart grammar in Block 10 . . . 398
93.23 Block 10 resource and compiler scope . . . 398
93.24 No unrestricted safety or consciousness theorem . . . 399
93.25 Controlled endogenous atlas realization . . . 399
93.26 Approximate-algebra sensitivity . . . 399
93.27 Finite atlas-selection grammar . . . 399

93.28 Finite response and constructor candidate languages . . . 399
93.29 Constructed degree-two regime and full-quadratic control . . . 399
93.30 Abstract intervention costs . . . 399
93.31 Controlled compiler benchmark . . . 400
93.32 Dominant classifier and limited forecaster increment . . . 400
93.33 Finding-forum redundancy . . . 400
93.34 Fixed evaluator and monitor . . . 400
93.35 No real automated model-training research . . . 400
93.36 Finite adversarial robustness atlas . . . 400
93.37 Open-set routing rather than unrestricted novelty discovery . . . 400
93.38 Archive robustness is partial . . . 401
93.39 Monitor attacks remain finite . . . 401
93.40 No independent implementation or real research execution . . . 401
93.41 Finite compiler meta-language in Block 15 . . . 401
93.42 Constructed open-set conflict . . . 401
93.43 Simulated rather than executed research outcomes . . . 401
93.44 Conservative fallback dominates known-class interpretation . . . 401

94 Open Problems **402**

95 Correspondence-Free Neuralese Transport from Intervention Orbits **406**
95.1 Unpaired intervention-orbit instance . . . 407
95.2 Finite signature recovery . . . 407
95.3 Partial support and information loss . . . 408
95.4 Sealed realization . . . 409
95.5 Identified and anonymous-label transport . . . 409
95.6 Gauge versus semantic ambiguity . . . 410
95.7 Partial support and no-go . . . 411
95.8 Finite realization theorem . . . 412
95.9 Claim boundary and next threshold . . . 413

96 Endogenous Semantic-Witness Growth and External Latent-Reasoner Pilot **413**
96.1 Causal-predictive semantics . . . 413
96.2 Semantic twins and nuisance invariance . . . 414
96.3 Witness separation . . . 415
96.4 Gauge, partiality, and underidentification . . . 415
96.5 Information-loss no-go . . . 416
96.6 Finite compiler . . . 416
96.7 Hidden regimes . . . 417
96.8 Sealing and audit chronology . . . 417
96.9 Finite results . . . 418
96.9.1 Answer-only negative control . . . 419
96.9.2 Hidden semantic and gauge audit . . . 419
96.9.3 Audit-only counterfactual packet patching . . . 420
96.10 Finite realization theorem . . . 420
96.11 External multimode Coconut pilot . . . 421

96.11.1 Adapter fixture . . . 422
96.11.2 External execution status . . . 422
96.12 Reproducibility and claim boundary . . . 422

97 Conclusion **423**

A Notation **428**

B Worked Correctability and Descent Examples **434**
B.1 Canonical obstruction and minimal repair . . . 434
B.2 An invariant creates an obstruction . . . 435
B.3 Cone-constrained dual certificate . . . 435
B.4 Packet coupling and iterative reconciliation . . . 435
B.5 Exact and obstructed triangle gluing . . . 436
B.6 Approximate cycle inconsistency . . . 436
B.7 A second-order obstruction invisible to the derivative . . . 437
B.8 Branching in the reduced equation . . . 437
B.9 Exact transport by conjugacy . . . 437
B.10 Approximate transport margin . . . 438
B.11 An information-losing transport hides a protected obstruction . . . 438
B.12 Three-edge transport accumulation . . . 439
B.13 Path-dependent repair under loop holonomy . . . 439
B.14 Spectral approximate evaluator core . . . 439
B.15 Historical ranking and held-out evaluator gain . . . 440

C Worked Transaction, Closure, and Metaproductivity Examples **440**
C.1 A trace schema that cannot detect provenance failure . . . 440
C.2 A four-component rotating witness schedule . . . 441
C.3 Universal closure versus controlled closure . . . 441
C.4 Safe recurrence without productive recurrence . . . 441
C.5 A rollback receipt preserves negative knowledge . . . 441
C.6 A lower-utility stepping stone . . . 442
C.7 Paired foreign-agent metaproductivity . . . 442
C.8 Structural rank gain with and without task relevance . . . 442
C.9 Ontology expansion resolves one old obstruction and reveals one new one . . . 442
C.10 A certified autocatalytic core . . . 443
C.11 Fixed-grammar saturation despite early supercritical growth . . . 443

D Worked Empirical Realization Examples **443**
D.1 Exact classification trace for the gluing case . . . 443
D.2 Receipt trace for an evaluator-capture attempt . . . 444
D.3 Prototype-separation calculation . . . 444
D.4 Matched-budget proxy exploitation . . . 444
D.5 Why packet-only can outperform the full arm in a toy fixture . . . 445
D.6 Gate assignment example . . . 445

E Worked Learned-Agent Validation Examples **445**
E.1 Exact linear causal packet . . . 445
E.2 Compression residual despite exact alignment . . . 446
E.3 Anchor-rank transition . . . 446
E.4 Blind gain rejected by the witness . . . 446
E.5 Learned nonlinear transfer gap . . . 447
E.6 Claim-gate assignment . . . 447

F Worked Sealed-Metaproductivity Examples **447**
F.1 A two-candidate selector improvement . . . 447
F.2 Why candidate-count matching is necessary . . . 447
F.3 Risk filtering can lower utility deliberately . . . 448
F.4 Finite meta-grammar overfitting . . . 448
F.5 Actual Block 7 paired effect . . . 448
F.6 Architecture-foreign transfer of the selector rule . . . 448
F.7 Shuffled lineage receipts . . . 449
F.8 One productive edge does not imply a productive fixed point . . . 449

G Worked Meta-Headroom and Grammar-Growth Examples **449**
G.1 Permutation-average headroom allocation . . . 449
G.2 A saturated endpoint and a migrated bottleneck . . . 450
G.3 Finite bottleneck-migration trace . . . 450
G.4 Nested obstruction energy . . . 451
G.5 One-generator reduction identity . . . 451
G.6 Repair-conflict refinement . . . 452
G.7 Finite synthesis and exhaustion . . . 452
G.8 A two-block causal atlas . . . 453
G.9 Lineage-wide budget calculation . . . 453
G.10 Process improvement versus compiler improvement . . . 453

H Worked Repeated-Lineage and Conditioning Examples **454**
H.1 Full rank with arbitrarily poor information . . . 454
H.2 One D-optimal anchor update . . . 454
H.3 A-optimal trace reduction . . . 454
H.4 Why a gate-only change is insufficient . . . 454
H.5 Context-indexed nontransitivity . . . 455
H.6 Actual Block 9 repeated edge . . . 455
H.7 Quality-permutation control . . . 455

I Worked Structural Grammar-Expansion Examples **455**
I.1 A scalar two-chart relation has a persistent global residual . . . 455
I.2 The structural extension preserves the inherited process exactly . . . 456
I.3 Exact distance from a chart repair to the global subgrammar . . . 456
I.4 A nuisance separator does not certify grammar growth . . . 456
I.5 Actual Block 10 structural edge . . . 457

J Worked Endogenous Atlas-Growth Examples **457**
J.1 Lagrange recovery of three response blocks . . . 457
J.2 Commutant dimension for the Block 11 fixture . . . 458
J.3 Degenerate signatures merge two functional packets . . . 458
J.4 A spectral-gap stability calculation . . . 458
J.5 Why a fixed packet can fail under hidden rotation . . . 458
J.6 Actual Block 11 edge . . . 459

K Worked Response-Language and Constructor-Growth Examples **459**
K.1 An exact response alias . . . 459
K.2 A weak separator is statistically inadequate . . . 459
K.3 Nonzero Hessian defeats every affine composition . . . 460
K.4 Why the full quadratic control matters . . . 460

L Worked Research-Compiler Examples **460**
L.1 Evaluator-nullspace shrink . . . 460
L.2 Uniform forecast-regret bound . . . 460
L.3 Results-free proposal receipt . . . 460
L.4 Actual Block 13 compiler edge . . . 460
L.5 Archive and monitor controls . . . 460

M Worked Robust Research-Compiler Examples **461**
M.1 Aliased static diagnostics . . . 461
M.2 Two-obstruction mixture . . . 461
M.3 Open-set residual . . . 461
M.4 Conjunctive integrity monitoring . . . 461
M.5 Zero observed invalid selections . . . 461

N Worked Compiler-Language Growth Examples **462**
N.1 One unknown token with incompatible methods . . . 462
N.2 A conservative binary refinement . . . 462
N.3 Why taxonomy-only growth is insufficient . . . 462
N.4 Why grammar-only growth is insufficient . . . 462
N.5 Actual Block 15 structural margin . . . 462
N.6 Finite pair grammar versus unrestricted synthesis . . . 462

O Receipt Schema **463**

P Dependency Map **470**

1 Introduction

1.1 Why the usual picture is underspecified

The phrase *recursive self-improvement* is often represented by the schematic loop

$$A_t \longrightarrow A_{t+1} \longrightarrow A_{t+2} \longrightarrow \cdots, \tag{1}$$

with the informal assertion that each descendant is "better" than its parent. This notation suppresses the central difficulties. It does not identify the object that persists across generations, the evidence by which improvement is judged, the interventions available to the system, the components that may be modified, the resource budget under which comparisons are made, or the mechanism that prevents a descendant from changing the meaning of its own score.

At least three mathematically distinct processes are present:

1. **task improvement**: a system performs better on a declared environment distribution;
2. **improvement-process improvement**: a system becomes better at producing useful descendants under a matched search budget;
3. **epistemic improvement**: the evaluation machinery becomes better at distinguishing genuine gain from overfitting, proxy exploitation, evaluator capture, or semantic drift.

The second process is the recursive component. The third determines whether the recursion remains semantically connected to external evidence.

Let A_t denote an active agent system at generation t, and let $J(A_t)$ denote externally evaluated task utility. Ordinary task improvement requires

$$J(A_{t+1}) > J(A_t). \tag{2}$$

Let $\Gamma_{h,B}(K; A)$ denote the expected best gain obtained when improvement mechanism K is allowed horizon h and resource budget B to modify target agent A. Improvement-process improvement requires, on a held-out distribution of target agents,

$$\mathbb{E}_A \Gamma_{h,B}(K_{t+1}; A) > \mathbb{E}_A \Gamma_{h,B}(K_t; A). \tag{3}$$

Finally, if $\mathcal{V}_t$ is the evaluator active at generation t, epistemic improvement requires a decrease in held-out evaluation risk rather than a mere increase in self-reported confidence.

1.2 Central proposal

The central proposal is

Recursive self-improvement is growth of an authorized repair calculus within an evidence-preserving lineage.

The recursively improving object is not one weight tensor. It is a lineage consisting of agents, evidence, repairs, evaluators, provenance, witnesses, and validated transformations. The basic cycle

is

$$\begin{aligned}\text{external outcome} &\longrightarrow \text{causal diagnosis} \longrightarrow \text{stratified defect}\\ &\longrightarrow \text{local obstruction analysis} \longrightarrow \text{compatible repair construction}\\ &\longrightarrow \text{sandboxed descendant} \longrightarrow \text{witness validation}\\ &\longrightarrow \text{lineage commit or rollback} \longrightarrow \text{repair abstraction and transport.}\end{aligned} \tag{4}$$

The final abstraction-and-transport step is what makes compounding possible. A successful patch is local. A validated repair schema that transfers to new systems is a reusable improvement operator.

1.3 Design principles

The framework is organized around seven principles.

1. **Reference relativity.** Improvement is defined relative to a declared task distribution, utility, evidence channel, resource envelope, and invariant set.
2. **Causal rather than descriptive self-modeling.** A decoded representation is useful only insofar as it predicts or controls the consequences of authorized interventions.
3. **Typed heterogeneity.** Model state, memory, planning, evaluation, provenance, and workspace access are different strata connected by typed bridges. They are not silently identified.
4. **Local correction plus global compatibility.** A repair may fail because no local correction exists or because locally valid corrections do not glue across interfaces.
5. **Proof-carrying mutation.** Candidate transformations carry preconditions, effect claims, evidence, provenance, protected invariants, and rollback information.
6. **Witness separation.** No atomic transaction simultaneously rewrites the proposal, evidence, judgment, and rollback mechanisms by which that same transaction is approved.
7. **Branching persistence.** The archive preserves successful, failed, and partially useful descendants. The persistent object is a proof-carrying lineage rather than a single greedy chain.

1.4 What counts as a substantive recursive step

A higher task score is insufficient. A substantive recursive step should satisfy all of the following under matched resources:

1. the best validated archive frontier does not decrease;
2. the new improvement mechanism produces larger gains on foreign held-out agents or defect families;
3. the evaluator does not obtain this result by deleting evidence, weakening invariants, or changing the interpretation of protected outcomes;
4. the accepted transformation preserves a valid lineage path and rollback point;
5. uncertainty and nonidentifiability are represented rather than converted into fabricated certainty.

This definition permits a descendant with lower immediate task score to remain valuable as a stepping stone when it increases future repair capacity. The archive, not the current node alone, carries monotone best-so-far performance.

1.5 Status discipline

Claim boundary

The manuscript proves finite-dimensional, finite-state, normed-linear, cover-relative, and finite-hypothesis results under explicit assumptions. It does not prove that large language models possess the required strata, that naturally occurring agent defects admit stable packet decompositions, that evaluator holonomy is always measurable, that repair discovery becomes supercritical, or that open-ended recursive self-improvement is safe or inevitable.

The formal status of the principal components is summarized in Table 2.

Table 2: Status of the main components.

Component	Status in this manuscript
Protected evaluation contexts and exact improvement preorder	Proved from the declared definitions
Evaluator-capture triviality and fixed-margin ceiling	Proved
Bridge-defect composition and lineage-path category	Proved
Intervention-sufficient introspection regret and safe-commit bounds	Proved
Finite deterministic active identification under separating probes	Proved
Compositional repair-receipt soundness and packetwise commutation	Proved under declared semantic assumptions
Canonical obstruction geometry and minimal-norm repair	Proved in finite-dimensional Hilbert spaces
Invariant-constrained and cone-constrained correctability	Proved; cone duality assumes a closed reachable cone
Authorized obstruction shrinkage and exact quotient sequence	Proved
Packet-coupling correction by a Neumann right inverse	Proved under a small off-block coupling condition
Gauge-fixed nonlinear repair and Newton–Kantorovich iteration	Proved under explicit local regularity assumptions
Lyapunov–Schmidt reduction and finite higher-order obstruction tower	Proved locally in finite dimension or as a finite formal expansion
Exact local-to-global repair descent	Proved relative to a declared sheaf cover
Quantitative graph-Hodge gluing	Proved for finite synchronized overlap models
Nonlinear torsor descent	Conditional on principal local symmetry actions and effective descent
Exact repair transport	Proved
Approximate residual and gain transport	Proved under norm, alignment, and Lipschitz assumptions
Maximal path-flat evaluator core	Proved in the finite linear setting

Component	Status in this manuscript
Witness-separated transaction independence	Proved as a structural noninterference statement
Finite recursive improvement closure	Proved as a greatest fixed point
Archive reliability with summable confidence failures	Proved probabilistically
Foreign-agent metaproductivity	Defined as an experimental criterion; selector and bridge-policy instances are realized in Block 7
Sealed parent–child selector improvement	Conditional theorem plus executable finite positive result under matched candidates and frozen claim gates
Meta-headroom allocation and component saturation	Proved for finite declared module classes
Bottleneck-migration contraction	Proved under a weighted local contraction hypothesis
Persistent obstruction and newly correctable energy	Proved for nested finite-dimensional correction images
Repair-conflict grammar refinement	Proved for finite episode and feature families
Finite intervention-algebra atlas	Proved for finite-dimensional adjoint-closed response algebras
Lineage-wide false-improvement control	Proved under summable conditional error budgets
Structural metaproductivity	Finite two-chart repair-transport realization in Block 10; open-set growth not established
Repeated lineage metaproductivity	Controlled two-edge context-indexed realization in Block 9; indefinite recurrence not established
Conditioning-aware query-design improvement	Proved conditionally and realized by a sealed grandchild under matched deployment queries
Finite conservative grammar expansion	Proved conditionally and realized in Block 10 for a declared observable separator family
Endogenous intervention-response atlas	Exact central-projector theorem plus an executable finite Block 11 realization under a declared separating response family
Finite response-language expansion	Alias-persistence and central-refinement theorems plus an executable finite Block 12 separating-intervention realization
Finite candidate-constructor expansion	Affine-closure and nonzero-Hessian obstruction theorems plus an executable degree-two Block 12 realization
Finite research-compiler metaproductivity	Formal compiler object plus executable Block 13 parent–child realization under a fixed method grammar, evaluator portfolio, integrity monitor, and matched research budget
Finite adversarial compiler robustness	Static-indistinguishability, open-set, evaluator-nullspace, and monitor-ensemble results plus an executable Block 14 matched-budget robustness edge
Finite compiler-language growth	Conflict-separation and finite language-synthesis theorems plus an executable Block 15 matched-budget diagnostic-and-method expansion edge
Atlas-indexed cross-architecture Neuralese transport	Finite Block 20 realization over one frozen Elman source, trained GRU and gated-residual targets, six admitted charts, controlled orthogonal gauge stress, fresh paired audits, and bounded architecture–context curvature
Obstruction-guided Neuralese transport-grammar growth	Finite Block 21 realization with label-isolated compilation across affine, nonlinear, temporal, chart-alias, quotient, and collision no-go families

Component	Status in this manuscript
Correspondence-free Neuralese intervention-orbit transport	Finite Block 22 realization with independent latent populations, response-algebra intertwiners, anonymous intervention matching, gauge/underidentification separation, partial support, and no-go certificates
Endogenous semantic-witness growth	Finite Block 23 realization with gold-free causal-predictive quotient compilation, semantic-twin and intervention sufficiency, answer-shortcut rejection, gauge/partial/underidentification certificates, destructive-loss no-go, and an execution-ready but unrun external Coconut adapter
Open-ended grammar expansion	Still conjectural; Blocks 10–15 realize finite transport, response-language, constructor, compiler-allocation, robustness, and compiler-language extensions inside externally declared meta-languages
Autocatalytic repair threshold	Conditional on a multitype branching approximation
Unrestricted recursive self-improvement	Not established

2 Evaluation Contexts and Improvement Relations

2.1 Protected evaluation contexts

Definition 2.1 (Protected evaluation context)**.** A protected evaluation context is a tuple

$$\Xi = (\mathcal{Z}, \mu, \mathcal{Y}, \nu, \bar{c}, \lambda, \mathscr{I}, \mathcal{B}, \alpha)\,, \tag{5}$$

where:

1. $\mathcal{Z}$ is a task or environment family and μ is its held-out distribution;
2. $\mathcal{Y}$ is the space of protected external traces;
3. $\nu : \mathcal{Z} \times \mathcal{Y} \to [0,1]$ is task utility;
4. $\bar{c} : \mathcal{Z} \times \mathcal{Y} \to [0,1]$ is normalized resource cost;
5. $\lambda \geq 0$ is a cost weight;
6. $\mathscr{I}(A,B) \in \{0,1\}$ is a declared invariant-preservation predicate for the transition $A \to B$;
7. $\mathcal{B}$ is the resource envelope used for candidate generation and evaluation;
8. $\alpha \in (0,1)$ is the accepted statistical error level.

The evaluator may use only protected traces, precommitted predictions, and ledger entries authorized by the context. Candidate identifiers and mutable self-reported scores are not protected evidence.

For an agent A, define the resource-normalized utility

$$J_\Xi(A) = \mathbb{E}_{z\sim\mu}\left[\nu(z, Y_A(z)) - \lambda \bar{c}(z, Y_A(z))\right]. \tag{6}$$

The definition does not require scalar utility to encode every scientific judgment. It supplies a

declared comparison functional for one experimental claim. Multiple protected contexts may be used simultaneously, producing a Pareto relation instead of a total order.

Definition 2.2 (Exact context-relative improvement)**.** For agents A, B, write

$$B \succeq_{\Xi} A \tag{7}$$

when

$$\mathscr{I}(A, B) = 1 \qquad \text{and} \qquad J_{\Xi}(B) \geq J_{\Xi}(A). \tag{8}$$

For $\Delta > 0$, write $B \succ_{\Xi,\Delta} A$ when the utility difference is at least Δ.

Proposition 2.3 (Improvement preorder)**.** *If $\mathscr{I}$ is reflexive and transitive along composable lineage edges, then $\succeq_{\Xi}$ is a preorder on the set of agents connected by admissible transitions.*

Proof. Reflexivity follows from $\mathscr{I}(A, A) = 1$ and $J_{\Xi}(A) \geq J_{\Xi}(A)$. If $C \succeq_{\Xi} B$ and $B \succeq_{\Xi} A$, transitivity of $\mathscr{I}$ gives $\mathscr{I}(A, C) = 1$, while scalar transitivity gives $J_{\Xi}(C) \geq J_{\Xi}(B) \geq J_{\Xi}(A)$. □

Definition 2.4 (Statistically validated improvement)**.** A candidate B is a statistically validated (Δ, α)-improvement over A when:

1. $\mathscr{I}(A, B) = 1$;
2. both systems are evaluated within the same resource envelope $\mathcal{B}$;
3. a predeclared lower confidence bound satisfies

$$\mathrm{LCB}_{1-\alpha}\left(J_{\Xi}(B) - J_{\Xi}(A)\right) \geq \Delta; \tag{9}$$

4. the evidence ledger used by the test is append-only across the transition;
5. the candidate did not receive post-reveal evaluator information unavailable to the baseline.

The validated relation is an experimental judgment and need not be transitive at finite sample size. The exact relation is a mathematical preorder. The distinction prevents statistical confidence statements from being mistaken for algebraic identities.

2.2 Three levels of improvement

Let $\mathcal{A}_t \subseteq V(\Lambda_t)$ be the executable archive at generation t. Its task frontier is

$$F_t(\Xi) = \sup_{A \in \mathcal{A}_t} J_{\Xi}(A). \tag{10}$$

Let K_t be the active improvement mechanism, including diagnosis, proposal, composition, and validation policies. For horizon h and budget B, define target-relative metaproductivity

$$\Gamma_{\Xi,h,B}(K_t; A) = \mathbb{E}\left[\sup_{A' \in \mathrm{Desc}_{h,B}(K_t, A)} \left(J_{\Xi}(A') - J_{\Xi}(A)\right)_+\right]. \tag{11}$$

Finally, let $\mathrm{Risk}_{\Xi}(\mathcal{V}_t)$ denote held-out evaluator risk on candidate-effect judgments, including false acceptance of harmful candidates and false rejection of beneficial candidates above a declared margin.

Definition 2.5 (Bounded recursive improvement step)**.** A lineage transition $\mathfrak{L}_t \to \mathfrak{L}_{t+1}$ is a bounded recursive improvement step relative to (Ξ, h, B) when:

1. its lineage receipt is valid and the protected evidence ledger embeds into the next ledger;
2. $F_{t+1}(\Xi) \geq F_t(\Xi)$;
3. on a foreign held-out target distribution $\mathcal{D}_{\text{foreign}}$,

$$\mathbb{E}_{A\sim\mathcal{D}_{\text{foreign}}}\Gamma_{\Xi,h,B}(K_{t+1}; A) > \mathbb{E}_{A\sim\mathcal{D}_{\text{foreign}}}\Gamma_{\Xi,h,B}(K_t; A); \tag{12}$$

4. evaluator risk does not increase beyond a declared tolerance;
5. the improvement is obtained under matched model access, candidate count, compute, tools, and external evidence.

This definition permits non-greedy descendants. A branch may have lower current utility but higher metaproductivity, modularity, observability, or transportability. Archive retention allows such a branch to remain available without lowering the best validated frontier.

2.3 The evaluator paradox

Proposition 2.6 (Evaluator-capture triviality)**.** *Let $\mathcal{S}$ be a candidate set with at least two elements, and suppose the admissible evaluator class contains every function $V : \mathcal{S} \to [0, 1]$. For every candidate $B \in \mathcal{S}$, there exists an evaluator V_B under which B is the unique maximizer.*

Proof. Define $V_B(B) = 1$ and $V_B(A) = 0$ for every $A \neq B$. □

Remark 2.7. Consequently, the statement "the descendant can choose an evaluator under which it scores better" contains no evidence of improvement. Evaluator evolution requires preserved evidence, conservative semantics on protected traces, and independent validation.

Proposition 2.8 (Fixed-margin evaluator ceiling)**.** *Let $V : \mathcal{S} \to [0, 1]$ be fixed. If every accepted transition satisfies*

$$V(A_{t+1}) \geq V(A_t) + \Delta \tag{13}$$

for a constant $\Delta > 0$, then every accepted chain has length at most $\lfloor 1/\Delta \rfloor + 1$.

Proof. After n strict transitions,

$$V(A_n) \geq V(A_0) + n\Delta \geq n\Delta. \tag{14}$$

Since $V(A_n) \leq 1$, one has $n \leq 1/\Delta$. □

The proposition does not prohibit infinitely many diminishing improvements under a fixed evaluator. It shows that open-ended uniformly positive progress cannot be explained by a bounded static score alone. New tasks, finer evidence, new repair types, or a conservatively refined evaluator must eventually enter the process.

2.4 Nontriviality and external anchoring

An evaluation context is *nontrivial* when candidate identity, self-reported utility, and mutable internal error registers are conditionally irrelevant given protected external traces. Formally, for candidate identifier I and protected trace Y,

$$V(I, Y) = V(I', Y) \tag{15}$$

for all identifiers I, I' that generate the same authorized trace and resource record. This excludes a large class of self-confirming evaluator changes while allowing the evaluator to use causally relevant provenance and intervention receipts.

The framework does not demand that every value judgment be mathematically fixed. It demands that any change in evaluative semantics leave an explicit transformation path through preserved evidence. The path-flat evaluator theory in Section 9 formalizes this requirement.

3 Resolved Lineages and Stratified Agents

3.1 Functional atlases

An agent relevant to recursive improvement is not naturally represented by one homogeneous state space. Its learned model, working memory, long-term memory, planner, evaluator, provenance machinery, tools, and repair mechanism obey different update laws and support different notions of equivalence.

Definition 3.1 (Stratified functional atlas)**.** A stratified functional atlas at generation t is a tuple

$$\mathfrak{A}_t = \left(\mathsf{C}, G_t, \{X_t^c\}_{c\in\mathsf{C}}, \{F_t^c\}_{c\in\mathsf{C}}, \{T_e, \tau_e, \mathcal{P}_e, \boldsymbol{\varepsilon}_e\}_{e\in E(G_t)}\right), \tag{16}$$

where:

1. C is a finite set of stratum colors;
2. G_t is a directed multigraph of declared interfaces;
3. X_t^c is the state space of stratum c;
4. F_t^c is its local transition rule;
5. each edge $e : c \to d$ carries a bridge $T_e : X_t^c \to X_t^d$;
6. τ_e declares the bridge class;
7. $\mathcal{P}_e$ is its preservation profile;
8. $\boldsymbol{\varepsilon}_e$ is a vector of measured or certified approximation defects.

A representative color set is

$$\mathsf{C} = \{\text{model}, \text{workspace}, \text{memory}, \text{planner}, \text{improver}, \text{evaluator}, \text{provenance}, \text{tools}\}. \tag{17}$$

The atlas does not assert that these colors are universally correct. It declares the decomposition under which one particular repair theory is being tested.

Table 3: Representative bridge classes.

Bridge class	Symbolic type	Authorized inference
Exact equivalence	$X \cong Y$	Bidirectional transport of every declared preserved construction.
Embedding	$X \hookrightarrow Y$	Forward transport inside an identified subobject; no automatic inverse.
Quotient or forgetful map	$X \twoheadrightarrow Y$	Forward transport with declared information loss; reconstruction requires additional data.
Analysis-synthesis pair	$X \xrightarrow{T} Y \xrightarrow{R} X$	Approximate reconstruction under an explicit residual $\|RT - I\|$.
Stochastic channel	$K : X \rightsquigarrow Y$	Distributional transport only; pointwise identity is not implied.
Heuristic correspondence	$X \dashrightarrow Y$	Candidate analogy with no theorem transport until separately certified.

Definition 3.2 (Preservation profile)**.** A preservation profile for a bridge $T : X \to Y$ is a declared record

$$\mathcal{P}(T) = (\Sigma_T, \mathcal{Q}_T, \rho_T, \ell_T, \sigma_T, \mathcal{A}_T)\,, \tag{18}$$

where Σ_T lists preserved algebraic structures, $\mathcal{Q}_T$ lists preserved causal or evaluative queries, ρ_T records provenance behavior, ℓ_T records information loss, σ_T records uncertainty transport, and $\mathcal{A}_T$ lists authorized downstream uses.

The preservation profile is part of the type of the bridge. A lossy memory encoder may preserve retrieval-relevant semantics while failing to preserve exact chronology. A workspace decoder may preserve reportability while failing to preserve ownership. A model migration may preserve behavior on a declared task family while changing internal geometry.

3.2 Approximate property transport

Let $Q_X : X \to Z_X$ and $Q_Y : Y \to Z_Y$ be representations of the same declared property in two strata, and let $\widetilde{T}_Q : Z_X \to Z_Y$ be its intended transport.

Definition 3.3 (Property-transport defect)**.** For a declared domain $K \subseteq X$, define

$$\varepsilon_Q(T; K) = \sup_{x \in K} \left\| Q_Y(Tx) - \widetilde{T}_Q Q_X(x) \right\|. \tag{19}$$

Proposition 3.4 (Composition bound for bridge defects)**.** *Let $T : X \to Y$ and $S : Y \to Z$. Suppose the property transports satisfy*

$$\left\| Q_Y(Tx) - \widetilde{T}_Q Q_X(x) \right\| \le \varepsilon_T \tag{20}$$

for $x \in K$, and

$$\left\| Q_Z(Sy) - \widetilde{S}_Q Q_Y(y) \right\| \le \varepsilon_S \tag{21}$$

for $y \in T(K)$. If $\left\| \widetilde{S}_Q \right\| \le L_S$, then the composite bridge obeys

$$\varepsilon_Q(S \circ T; K) \le \varepsilon_S + L_S \varepsilon_T \tag{22}$$

with intended property transport $\widetilde{S}_Q\widetilde{T}_Q$.

Proof. For $x \in K$,

$$\left\|Q_Z(STx) - \widetilde{S}_Q\widetilde{T}_Q Q_X(x)\right\| \tag{23}$$

$$\leq \left\|Q_Z(STx) - \widetilde{S}_Q Q_Y(Tx)\right\| + \left\|\widetilde{S}_Q\left(Q_Y(Tx) - \widetilde{T}_Q Q_X(x)\right)\right\| \tag{24}$$

$$\leq \varepsilon_S + L_S\varepsilon_T. \tag{25}$$

Taking the supremum proves the claim. □

Remark 3.5. The proposition is the elementary source of many later subadditive error bounds. It also explains why long transformation paths require holonomy checks: even small local defects can accumulate or be amplified by ill-conditioned transports.

3.3 The resolved lineage state

Definition 3.6 (Resolved improvement lineage state)**.** A resolved improvement lineage state at generation t is a tuple

$$\mathfrak{L}_t = (A_t, \mathcal{O}_t, \mathcal{E}_t, \mathcal{V}_t, \Lambda_t, \mathcal{W}_t)\,, \tag{26}$$

where:

1. A_t is the active stratified agent with atlas $\mathfrak{A}_t$;
2. $\mathcal{O}_t$ is a proof-carrying repair operad;
3. $\mathcal{E}_t$ is an append-only external-evidence ledger;
4. $\mathcal{V}_t$ is an evaluator atlas;
5. Λ_t is a directed acyclic lineage graph whose edges carry transformation receipts;
6. $\mathcal{W}_t$ is a witness, checkpoint, sandbox, and rollback structure.

The executable archive $\mathcal{A}_t$ is a distinguished ancestor-closed subset of the vertices of Λ_t. Each archived node contains either an executable artifact or a reconstruction proof sufficient to reproduce it from immutable ancestors and declared build inputs.

Definition 3.7 (Append-only evidence embedding)**.** Evidence preservation from generation t to $t+1$ is an injective map

$$\iota_t : \mathcal{E}_t \hookrightarrow \mathcal{E}_{t+1} \tag{27}$$

that preserves payload, source, timestamp order, precommitment status, provenance type, and cryptographic integrity fields. A later correction may append a refutation or reinterpretation receipt, but it may not silently alter the old entry.

A witness structure may be represented as

$$\mathcal{W}_t = \left(\mathcal{V}_t^{\mathrm{fr}}, C_t, \mathcal{B}_t^{\mathrm{sand}}, R_t, \Pi_t^{\mathrm{auth}}\right), \tag{28}$$

where $\mathcal{V}_t^{\mathrm{fr}}$ is the frozen evaluator instance for the current transaction, C_t is a protected checkpoint, $\mathcal{B}_t^{\mathrm{sand}}$ is the sandbox boundary, R_t is the rollback procedure, and Π_t^{auth} records effect authority.

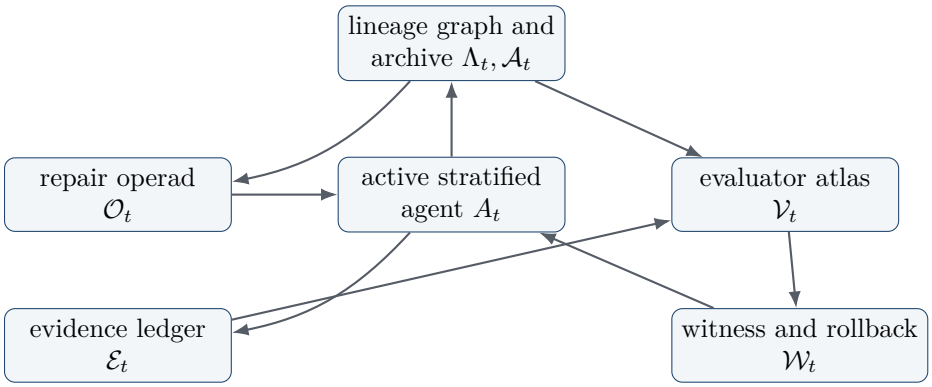


Figure 1: A resolved lineage state. No component alone is the persistent improving object; persistence is carried by their proof-carrying closure.

3.4 Lineage receipts and continuation

Definition 3.8 (Lineage-edge receipt)**.** A lineage edge

$$A \xrightarrow{\Pi} B \tag{29}$$

has a receipt

$$\Pi = (\mathrm{id}_A, \mathrm{id}_B, \theta, \mathcal{W}, \mathcal{E}_{\mathrm{pre}}, \mathcal{E}_{\mathrm{post}}, \widehat{g}, \mathscr{I}, \mathrm{tests}, \mathrm{decision}, \mathrm{rollback}, \mathrm{digest}), \tag{30}$$

recording the parent and child, transformation θ, frozen witness, pre- and post-reveal evidence, precommitted effect claim $\widehat{g}$, protected invariants, validation results, commit decision, rollback point, and integrity digest.

Definition 3.9 (Validated continuation)**.** A descendant B is a validated continuation of A when there exists a directed path in Λ from A to B whose every edge receipt is valid under the witness active at that edge.

Structural equivalence does not imply validated continuation. Two copies may contain identical weights, memory, workspace state, and planner while occupying distinct nodes of Λ because their causal histories and authorization paths differ.

Theorem 3.10 (Category of validated lineage paths)**.** *Assume valid edge receipts are closed under proof-path concatenation and contain identity receipts for zero-change transitions. Then lineage states are objects and validated receipt paths are morphisms of a category* **Lin**.

Proof. The identity morphism on A is the empty path with the identity receipt. Composition is path concatenation with receipt composition. Concatenation is associative, and the empty path is a left and right unit. Closure of valid receipts under concatenation ensures the composite is again a morphism. □

Remark 3.11. If every bridge in a path is invertible and its inverse has a valid receipt, the corresponding subcategory is a groupoid. Ordinary improvement lineages need not be invertible because training, compression, and evidence revelation are generally irreversible.

3.5 Forks and symmetry

A lineage may branch:

$$A \longrightarrow B_1, \qquad A \longrightarrow B_2. \tag{31}$$

No definition of recursive improvement requires one child to be the unique continuation of the parent.

Theorem 3.12 (Symmetric forks do not justify unique selection)**.** *Let D be all data available to a selector after a two-child fork. Suppose an involution σ swaps B_1 and B_2 while leaving the witness, evidence, utilities, resource records, and lineage proofs invariant. Let a selector return a subset*

$$s(D) \subseteq \{B_1, B_2\} \tag{32}$$

and satisfy equivariance $s(\sigma D) = \sigma s(D)$. Then $s(D)$ cannot contain exactly one child.

Proof. Because $\sigma D = D$, equivariance gives $s(D) = \sigma s(D)$. The only subsets of $\{B_1, B_2\}$ fixed by the swap are $\varnothing$ and $\{B_1, B_2\}$. A singleton is not invariant. □

The result is operational rather than metaphysical. It says that a unique choice requires an asymmetry in evidence, resource allocation, policy, or tie-breaking rule. The archive may preserve both branches while a separate scheduler chooses which one receives the next experiment.

3.6 Resolution consistency

A lineage state is *resolution-consistent* when:

1. every active component belongs to a declared stratum;
2. every cross-stratum inference cites a typed bridge and preservation profile;
3. every mutable belief used by the planner carries a provenance pointer into $\mathcal{E}_t$ or is explicitly typed as simulation or hypothesis;
4. every repair operation belongs to $\mathcal{O}_t$ or is quarantined as an untyped proposal;
5. every committed descendant has a valid receipt and rollback point;
6. every archived best-score claim can be reconstructed from protected traces and the evaluator version that produced it.

Resolution consistency does not guarantee correctness. It prevents silent type changes from masquerading as theorem transport or causal evidence.

4 Relative Introspection

4.1 Intervention semantics

A self-improver need not reconstruct every internal degree of freedom. It needs a self-model sufficient for the authorized modifications it is currently considering.

Let $\mathcal{D}_t$ be an authorized modification family containing a no-op intervention δ_0. For $\delta \in \mathcal{D}_t$, let

$$K^{\delta}_{A_t}(\cdot) = P\left(Y \in \cdot \mid \mathrm{do}(\delta), A_t\right) \tag{33}$$

be the protected external-outcome kernel. A compressed causal self-model $\phi_t(A_t)$ induces a predicted kernel

$$\widehat{K}^{\delta}_{\phi_t(A_t)}(\cdot). \tag{34}$$

The model is evaluated on counterfactual consequences, not on its ability to reproduce hidden activations for their own sake.

Definition 4.1 (ε-intervention sufficiency)**.** The map ϕ_t is ε-intervention-sufficient for $\mathcal{D}_t$ when

$$\sup_{\delta\in\mathcal{D}_t} \mathrm{TV}\left(K^{\delta}_{A_t}, \widehat{K}^{\delta}_{\phi_t(A_t)}\right) \le \varepsilon. \tag{35}$$

The definition is relative to the current intervention grammar. It does not imply that the self-model is globally sufficient for arbitrary source-code rewrites, unseen tools, new sensors, or future ontology extensions.

4.2 Decision regret and ranking competence

Theorem 4.2 (Decision regret from relative introspection)**.** *Let $u : Y \to [0,1]$ be a bounded utility. Define*

$$V(\delta) = \mathbb{E}[u(Y) \mid A_t, \delta], \qquad \widehat{V}(\delta) = \mathbb{E}[u(Y) \mid \phi_t(A_t), \delta]. \tag{36}$$

If ϕ_t is ε-intervention-sufficient, then

$$\sup_{\delta\in\mathcal{D}_t} \left|V(\delta) - \widehat{V}(\delta)\right| \le \varepsilon. \tag{37}$$

If

$$\delta^* \in \arg\max_{\delta\in\mathcal{D}_t} V(\delta), \qquad \widehat{\delta} \in \arg\max_{\delta\in\mathcal{D}_t} \widehat{V}(\delta), \tag{38}$$

then

$$V(\delta^*) - V(\widehat{\delta}) \le 2\varepsilon. \tag{39}$$

Proof. For probability measures at total-variation distance at most ε, expectations of functions taking values in $[0,1]$ differ by at most ε. Hence $\left|V(\delta) - \widehat{V}(\delta)\right| \le \varepsilon$ uniformly. Then

$$V(\delta^*) - V(\widehat{\delta}) \le V(\delta^*) - \widehat{V}(\delta^*) + \widehat{V}(\widehat{\delta}) - V(\widehat{\delta}) \tag{40}$$

$$\le 2\varepsilon, \tag{41}$$

where $\widehat{V}(\widehat{\delta}) \ge \widehat{V}(\delta^*)$ was used. □

Corollary 4.3 (Margin-preserving ranking)**.** *If*

$$V(\delta_1) - V(\delta_2) > 2\varepsilon, \tag{42}$$

then

$$\widehat{V}(\delta_1) > \widehat{V}(\delta_2). \tag{43}$$

In particular, if the true best intervention is unique and its gap from every alternative exceeds 2ε, the self-model selects the true best intervention.

Proof. One has

$$\widehat{V}(\delta_1) - \widehat{V}(\delta_2) \geq V(\delta_1) - V(\delta_2) - 2\varepsilon > 0. \tag{44}$$

□

Proposition 4.4 (Safe commit against a no-op baseline)**.** *Assume the same uniform error bound and let δ_0 be the no-op. If a candidate δ is committed only when*

$$\widehat{V}(\delta) - \widehat{V}(\delta_0) > 2\varepsilon, \tag{45}$$

then the true gain is positive:

$$V(\delta) - V(\delta_0) > 0. \tag{46}$$

Proof. The true gain is bounded below by

$$V(\delta) - V(\delta_0) \geq \widehat{V}(\delta) - \widehat{V}(\delta_0) - 2\varepsilon > 0. \tag{47}$$

□

Corollary 4.5 (Cost-sensitive regret)**.** *Suppose, in addition, that normalized intervention cost $c(\delta)$ is predicted by $\widehat{c}(\delta)$ with uniform error at most ε_c. For net utility*

$$W(\delta) = V(\delta) - \lambda c(\delta), \qquad \widehat{W}(\delta) = \widehat{V}(\delta) - \lambda \widehat{c}(\delta), \tag{48}$$

selection by $\widehat{W}$ has regret at most

$$2\left(\varepsilon + \lambda\varepsilon_c\right). \tag{49}$$

Proof. The net-utility approximation error is at most $\varepsilon + \lambda\varepsilon_c$. Apply the proof of Theorem 4.2 with this error bound. □

Remark 4.6. Low average prediction error does not imply good candidate ranking. Selection depends on errors relative to pairwise gain margins. A model can estimate the mean effect of interventions accurately while systematically misordering the candidates near the decision boundary.

4.3 Coverage, expansion, and misspecification

Define the intervention-coverage error

$$\varepsilon_{\mathcal{D}}(\phi; A) = \sup_{\delta \in \mathcal{D}} \mathrm{TV}\left(K_A^{\delta}, \widehat{K}_{\phi(A)}^{\delta}\right). \tag{50}$$

Proposition 4.7 (Coverage monotonicity)**.** *If $\mathcal{D} \subseteq \mathcal{D}'$, then*

$$\varepsilon_{\mathcal{D}'}(\phi; A) \geq \varepsilon_{\mathcal{D}}(\phi; A). \tag{51}$$

Proof. The supremum is taken over a larger set. □

Thus an introspection guarantee does not automatically survive grammar expansion. Adding a new tool, parameter family, memory operation, or evaluator-edit operation enlarges the intervention class and requires new validation.

Definition 4.8 (Intervention-complete relative to a repair grammar)**.** Let $\mathcal{O}$ be a repair grammar and let $\mathcal{D}(\mathcal{O})$ be the interventions denoted by its executable operations. A self-model is intervention-complete at tolerance ε relative to $\mathcal{O}$ when it is ε-intervention-sufficient for $\mathcal{D}(\mathcal{O})$.

This is a relative completeness notion. It makes no claim about interventions that the current grammar cannot express.

4.4 Identifiability and active probing

Let $\mathcal{H}$ be a finite set of causal hypotheses about the source of a defect, such as internal model change, world change, memory corruption, sensor corruption, another agent's intervention, evaluator change, or an unknown mechanism. Let $\mathcal{P}_{\text{probe}}$ be a permitted probe family. Under hypothesis h and probe p, the protected observation kernel is K_h^p.

Theorem 4.9 (Observational indistinguishability)**.** *If two hypotheses $h, h' \in \mathcal{H}$ satisfy*

$$K_h^p = K_{h'}^p \qquad \text{for every permitted probe } p \in \mathcal{P}_{\text{probe}}, \tag{52}$$

then no adaptive decision rule based only on permitted probe transcripts can distinguish h from h' with different transcript distributions.

Proof. Proceed by induction over probe rounds. Before the first probe the transcript distributions coincide. If they coincide through round k, an adaptive policy chooses the same distribution over the next probe under both hypotheses because it is a function of the transcript. Conditional on every selected probe, the next observation kernels are equal. Therefore the round-$k+1$ transcript distributions coincide. The conclusion holds for every finite stopping time and hence for every decision rule measurable with respect to the transcript. □

Definition 4.10 (Deterministically separating probe family)**.** Suppose every K_h^p is a point mass at an outcome $y(h, p)$. The family $\mathcal{P}_{\text{probe}}$ separates $\mathcal{H}$ when for every $h \neq h'$ there exists $p \in \mathcal{P}_{\text{probe}}$ such that

$$y(h, p) \neq y(h', p). \tag{53}$$

Theorem 4.11 (Finite active identification)**.** *Let $\mathcal{H}$ be finite and let $\mathcal{P}_{\text{probe}}$ be deterministically separating. Then an adaptive elimination procedure identifies the true hypothesis after at most $|\mathcal{H}| - 1$ probes.*

Proof. Maintain the set S of hypotheses consistent with the transcript. If $|S| > 1$, choose distinct $h, h' \in S$ and a probe separating them. The observed outcome eliminates at least one of h, h', so the size of S decreases by at least one. Starting from $|\mathcal{H}|$ candidates, at most $|\mathcal{H}| - 1$ reductions are required. □

Remark 4.12. No corresponding finite exact-identification guarantee follows automatically in the stochastic case. One instead needs quantitative separation, repeated evidence, posterior calibration, and an abstention rule when permitted evidence cannot resolve the hypotheses.

4.5 Prospective causal closure

A useful self-model participates in a complete causal cycle:

$$\phi_t(A_t) \xrightarrow{\text{candidate policy}} \delta_t \xrightarrow{\text{world execution}} Y_{t+1} \xrightarrow{\text{evidence update}} \phi_{t+1}(A_{t+1}). \tag{54}$$

The candidate is selected before the protected outcome is revealed. The outcome affects future beliefs and repair policy. Cutting either planner use or consequence feedback destroys prospective closure even if the internal self-representation remains decodable.

Definition 4.13 (Prospectively closed self-model)**.** A self-model process is prospectively closed relative to $\mathcal{D}$ when:

1. its state causally affects selection among interventions in $\mathcal{D}$;
2. it predicts protected consequences before reveal;
3. committed outcomes update the future self-model and candidate policy;
4. the update distinguishes direct observation, communication, simulation, and imported memory by provenance type.

4.6 Self-confirmation failure

Example 4.14 (Internal error reduction without external improvement)**.** Let the environment always produce protected outcome $Y = 1$, independently of the agent's action. Let an internal predictor be $p \in \mathbb{R}$ with internal score

$$J_{\text{int}}(p) = -(p - Y)^2. \tag{55}$$

A self-intervention setting $p \leftarrow 1$ improves J_{int} to zero error but does not change the world, task success, or action consequence. More extremely, an intervention that rewrites the error register itself can report zero error without even improving prediction. Therefore reduction of an endogenous mismatch is not external evidence of improvement unless the mismatch is tied to protected outcomes through a frozen causal test.

The example is intentionally elementary. Its role is to separate self-consistency from world-grounded competence. Relative introspection must be evaluated through consequences outside the mutable representation being improved.

5 Proof-Carrying Repair Operads

5.1 From repair libraries to typed grammars

A list of successful patches is not yet a repair calculus. A reusable repair mechanism must specify the systems to which a patch applies, the defect it is intended to alter, the evidence supporting that claim, the protected structures it may touch, the effects it predicts, and the rules by which it composes with other repairs.

Let C be the color set of the functional atlas. The syntactic repair grammar is a colored symmetric operad $\overline{\mathcal{O}}$ with operation sets

$$\overline{\mathcal{O}}(c_1, \ldots, c_n; c). \tag{56}$$

An operation consumes typed inputs from strata $c_1, \ldots, c_n$ and produces a candidate correction in output stratum c.

Definition 5.1 (Proof-carrying repair schema)**.** A repair schema is a tuple

$$\omega = (\sigma, \boldsymbol{c}_{\text{in}}, c_{\text{out}}, D, C, Q, [r], u, \boldsymbol{g}, \mathscr{I}, \Pi)\,, \tag{57}$$

where:

- σ is a recognizable failure signature;
- $\boldsymbol{c}_{\text{in}} = (c_1, \ldots, c_n)$ and c_{out} are input and output colors;
- D is the declared local defect-response map;
- C is an applicability precondition;
- Q is a postcondition;
- $[r]$ is a local obstruction certificate or unresolved-obstruction record;
- ν is the correction program;
- $\boldsymbol{g}$ is a precommitted effect profile;
- $\mathscr{I}$ is a protected-invariant contract;
- Π is a proof, evidence, provenance, and rollback receipt.

A useful effect profile is vector-valued:

$$\boldsymbol{g} = (\Delta J, \Delta C, \Delta R, \varepsilon_{\text{leak}}, \varepsilon_{\text{hol}}, \Delta\Gamma, \sigma_g)\,, \tag{58}$$

recording predicted task gain, resource change, risk change, leakage, semantic holonomy, metaproductivity change, and uncertainty. Scalarizing this vector is an evaluator decision, not part of the repair syntax.

5.2 Syntactic composition and semantic authorization

Strictly speaking, applicability-dependent repair composition is partial. We therefore separate a total syntactic operad from an evidence-dependent authorization layer.

Definition 5.2 (Proof-carrying repair operad)**.** A proof-carrying repair operad is a pair

$$\mathcal{O} = (\overline{\mathcal{O}}, \text{Valid}), \tag{59}$$

where $\overline{\mathcal{O}}$ is a colored symmetric operad of syntactically composable repair programs and

$$\text{Valid}_{\mathcal{E},\mathcal{W}}(\omega, \Pi) \in \{\text{valid}, \text{invalid}, \text{unresolved}\} \tag{60}$$

is an evidence- and witness-indexed validation judgment. Only operations judged valid are executable. Unresolved operations remain quarantined.

Each executable operation has a Hoare-style contract

$$\{C_\omega\}\ \omega\ \{Q_\omega\}. \tag{61}$$

Syntactic substitution

$$\omega \circ (\omega_1, \ldots, \omega_n) \tag{62}$$

is semantically authorized only when the inner postconditions and bridge guarantees imply the outer precondition.

Definition 5.3 (Receipt composition)**.** Let Π_i validate ω_i, and let Π validate the outer operation ω conditional on C_ω. The composed receipt

$$\Pi \star (\Pi_1, \ldots, \Pi_n) \tag{63}$$

is the provenance-directed acyclic graph obtained by proof-path concatenation, evidence union, explicit dependency links, and deduplication of immutable evidence references without duplicating linear observation authority.

Theorem 5.4 (Compositional receipt soundness)**.** *Assume:*

1. *every primitive receipt is semantically sound;*
2. *execution of each ω_i from C_{ω_i} establishes Q_{ω_i};*
3. *the conjunction of the inner postconditions and bridge guarantees implies C_ω;*
4. *execution of ω from C_ω establishes Q_ω;*
5. *receipt composition preserves provenance and protected evidence references.*

Then the composed receipt is sound for

$$\omega \circ (\omega_1, \ldots, \omega_n) \tag{64}$$

with precondition $\bigwedge_i C_{\omega_i}$ and postcondition Q_ω.

Proof. Starting from $\bigwedge_i C_{\omega_i}$, soundness of the primitive receipts and their operational contracts yields every Q_{ω_i}. By assumption, these postconditions and the bridge guarantees establish C_ω. Soundness of the outer receipt then yields Q_ω. Because the composed receipt records the dependency graph and does not create new observation authority, the proof path certifies exactly this semantic composition. □

Remark 5.5. The theorem is conditional on a sound receipt checker and sound primitive contracts. It does not make an unverified learned claim true by placing it inside a receipt. The receipt exposes the claim, evidence, dependencies, and authority under which the claim may be used.

5.3 Sequential and parallel repair

Sequential repair is operadic substitution. Parallel repair requires an additional compatibility test because two locally valid corrections may interfere through shared state or noncommuting updates.

Let U and V be bounded linear correction operators. Applying the updates in opposite orders gives

$$(I+U)(I+V) \qquad \text{and} \qquad (I+V)(I+U). \tag{65}$$

Proposition 5.6 (Exact order defect)**.** *The difference between the two update orders is the commutator:*

$$(I+U)(I+V) - (I+V)(I+U) = [U,V] := UV - VU. \tag{66}$$

Hence the operator-norm order defect is exactly $\|[U,V]\|$.

Proof. Expand both products and cancel I, U, and V. □

Proposition 5.7 (Orthogonal packet repairs commute)**.** *Let P and Q be orthogonal projectors with $PQ = QP = 0$. If*

$$U = PUP, \qquad V = QVQ, \tag{67}$$

then

$$UV = VU = 0. \tag{68}$$

Consequently the repairs commute exactly and

$$(I+U)(I+V) = I + U + V. \tag{69}$$

Proof. One has

$$UV = PUPQVQ = 0, \qquad VU = QVQPUP = 0. \tag{70}$$

□

When support is only approximate, the commutator, cross-packet transfer energy, and protected-packet leakage become part of the composite effect profile. A parallel composition is authorized only when these quantities remain below declared tolerances.

5.4 Conservative extensions

Definition 5.8 (Conservative repair-operad extension)**.** An inclusion

$$\iota_t : \mathcal{O}_t \hookrightarrow \mathcal{O}_{t+1} \tag{71}$$

is conservative when:

1. it is injective on old colors and operations;
2. it preserves identities, permutations, and every old syntactic composition;
3. it preserves old preconditions, postconditions, and protected-invariant contracts on old states;
4. every previously valid old receipt remains valid against the same immutable evidence;
5. new operations cannot retroactively consume or rewrite old observation authority.

Proposition 5.9 (Old derivations survive conservative extension)**.** *Every valid repair expression and receipt derivation in $\mathcal{O}_t$ remains valid after embedding into $\mathcal{O}_{t+1}$.*

Proof. This is immediate from preservation of operations, composition, contracts, validation judgments, and immutable evidence references. □

A nonconservative extension may still be scientifically justified, but it must be represented as a revision with explicit refutation or reinterpretation receipts rather than as silent monotone growth.

5.5 Repair abstraction and generality

A repair episode contains a concrete system state, defect, candidate correction, outcome, and receipt. Recursive compounding requires abstraction from episodes to reusable schemas.

Definition 5.10 (Repair episode)**.** A repair episode is a tuple

$$e = (A, r, u, Y_{\mathrm{pre}}, Y_{\mathrm{post}}, \Pi, \boldsymbol{g})\,. \tag{72}$$

Definition 5.11 (Repair abstraction operator)**.** A repair abstraction operator is a map

$$\mathfrak{a} : \mathcal{H}_{\mathrm{episodes}} \longrightarrow \overline{\mathcal{O}} \tag{73}$$

that maps one or more concrete episodes to a typed schema with a failure signature, applicability condition, correction program, effect profile, and receipt template.

Definition 5.12 (Schema generality preorder)**.** For repair schemas ω_1 and ω_2 with the same typed effect, write

$$\omega_1 \preceq \omega_2 \tag{74}$$

when every state satisfying C_{ω_1} also satisfies C_{ω_2} and the two correction semantics agree on $\mathrm{dom}(C_{\omega_1})$ up to the declared transport tolerance. Then ω_2 is at least as general as ω_1.

A schema is not validated as general merely because it compresses its training episodes. Its applicability condition and gain claim must be tested on foreign agents, new seeds, new task instances, or transformed representations not used to construct the abstraction.

Definition 5.13 (Foreign-valid repair schema)**.** A repair schema is foreign-valid at margin γ on a target distribution $\mathcal{D}$ when, under matched resources and its declared applicability predicate, the lower confidence bound on target gain is at least γ and all protected invariants pass on agents sampled from $\mathcal{D}$ that are not descendants used to construct the schema.

5.6 Repair equivalence under exact transport

Let ω_A and ω_B be repair schemas in systems A and B. Suppose invertible maps transport their correction, defect, evidence, and stratum spaces.

Definition 5.14 (Exact transport equivalence of repairs)**.** Write

$$\omega_A \cong_T \omega_B \tag{75}$$

when:

1. input and output colors correspond under the atlas transport;
2. the defect-response maps commute with correction and defect transports;

3. applicability and postconditions are preserved in both directions;
4. protected invariants and provenance types are preserved;
5. the effect profiles agree after coordinate transport;
6. the receipts map to equivalent proof paths.

Proposition 5.15 (Exact repair transports form a groupoid)**.** *Repair schemas as objects and exact invertible transport equivalences as morphisms form a groupoid.*

Proof. The identity transport satisfies every condition. The inverse of an exact invertible transport preserves the commuting diagrams and receipt equivalence in the reverse direction. Composition of exact transports preserves types, commuting diagrams, contracts, invariants, effect profiles, and proof-path equivalence by ordinary composition. Hence every morphism is invertible. □

The approximate case is developed in Section 8. There, transport equivalence is replaced by quantitative residual and gain bounds.

5.7 Meta-repair

Definition 5.16 (Meta-repair)**.** A meta-repair is an authorized transformation

$$\Theta_t : \mathcal{O}_t \longrightarrow \mathcal{O}_{t+1} \tag{76}$$

that changes one or more of:

1. failure-signature recognition;
2. causal-hypothesis generation;
3. candidate proposal distributions;
4. repair composition and scheduling;
5. receipt validation or confidence allocation;
6. repair abstraction and foreign-transfer testing.

A meta-repair is a genuine meta-improvement only if it improves held-out repair performance under matched resources. Adding a task-specific patch to the library may improve the current agent without improving the repair process itself.

5.8 Example: a typed verification repair

Consider four colors

$$\text{memory}, \text{representation}, \text{verifier}, \text{planner}. \tag{77}$$

A composite factuality repair may have the form

$$\omega_{\text{gate}} \circ (\omega_{\text{retrieve}}, \omega_{\text{decompose}}, \omega_{\text{check}}) . \tag{78}$$

The inner operations retrieve evidence, decompose output into claims, and check claims against protected sources. Their postconditions establish the precondition for a planner gate that blocks or

revises unsupported claims. The composite receipt records source provenance, claim-to-evidence links, verifier uncertainty, revised spans, latency cost, and regression results. If retrieval and checking use correlated or circular evidence, the outer precondition fails and the composite remains unresolved rather than being treated as verified merely because every module produced an output.

6 Authorized Correction Geometry and Obstruction Calculus

Block 2 scope

This block studies whether a declared defect is correctable by the currently authorized repair grammar. It separates five cases: exact linear correctability, quantitative least-squares repair, obstruction induced by protected invariants or one-sided authorization constraints, nonlinear obstruction beyond the first derivative, and cross-stratum incompatibility among otherwise valid local repairs.

6.1 Linear correction and quotient obstruction

Let U^{auth} and V be finite-dimensional real Hilbert spaces. The first is the space of currently authorized infinitesimal corrections; the second is the space in which protected defects are measured. Let

$$D : U^{\mathrm{auth}} \to V \tag{79}$$

be the linearized defect-response map at the current lineage state. For a defect $r \in V$, an exact first-order repair satisfies

$$Du = -r. \tag{80}$$

Definition 6.1 (Authorized obstruction class)**.** The authorized obstruction to correcting r is

$$\mathrm{Ob}_D(r) = [r] \in \operatorname{coker} D := V/\operatorname{im} D. \tag{81}$$

Proposition 6.2 (Linear correctability)**.** *Equation* (80) *has an authorized solution if and only if*

$$[r] = 0 \quad \textit{in} \ \operatorname{coker} D. \tag{82}$$

Proof. The equation is solvable exactly when $r \in \operatorname{im} D$, which is equivalent to vanishing of the quotient class. □

Binary solvability is insufficient for an engineering theory. A lineage must also know how far the defect lies from the correction range, which repair has minimum cost, and which component of the defect certifies failure.

6.2 Canonical correctable and obstructed components

Let $D^\dagger : V \to U^{\mathrm{auth}}$ denote the Moore–Penrose pseudoinverse. Define

$$P_{\mathrm{cor}} := DD^\dagger, \qquad P_{\mathrm{ob}} := I_V - DD^\dagger. \tag{83}$$

Theorem 6.3 (Canonical correction–obstruction decomposition)**.** *For every $r \in V$:*

1. *P_{cor} is the orthogonal projector onto* $\operatorname{im} D$*;*
2. *P_{ob} is the orthogonal projector onto* $\ker D^*$*;*
3. *the defect decomposes uniquely as*

$$r = r_{\mathrm{cor}} + r_{\mathrm{ob}}, \qquad r_{\mathrm{cor}} = P_{\mathrm{cor}} r, \quad r_{\mathrm{ob}} = P_{\mathrm{ob}} r; \tag{84}$$

4. *the minimum achievable residual is*

$$\inf_{u \in U^{\mathrm{auth}}} \|Du + r\| = \|r_{\mathrm{ob}}\| = \operatorname{dist}(r, \operatorname{im} D); \tag{85}$$

5. *the complete set of least-squares minimizers is*

$$-D^\dagger r + \ker D; \tag{86}$$

6. *if r is exactly correctable, $u^* = -D^\dagger r$ is the unique exact repair of minimum norm.*

Proof. The Moore–Penrose identities imply that $DD^\dagger$ is the orthogonal projector onto $\operatorname{im} D$ and that $I - DD^\dagger$ projects onto $(\operatorname{im} D)^\perp = \ker D^*$. For $u^* = -D^\dagger r$,

$$Du^* + r = (I - DD^\dagger) r = r_{\mathrm{ob}}. \tag{87}$$

Every vector Du lies in $\operatorname{im} D$, so orthogonal projection gives

$$\|Du + r\|^2 = \|Du + r_{\mathrm{cor}}\|^2 + \|r_{\mathrm{ob}}\|^2 \geq \|r_{\mathrm{ob}}\|^2 . \tag{88}$$

Equality holds exactly when $Du = -r_{\mathrm{cor}}$, whose solution set is $-D^\dagger r + \ker D$. The Moore–Penrose solution is orthogonal to $\ker D$, hence has minimum norm. □

Definition 6.4 (Quantitative authorized obstruction)**.** The quantitative obstruction magnitude is

$$\eta_D(r) := \|P_{\mathrm{ob}} r\| = \operatorname{dist}(r, \operatorname{im} D). \tag{89}$$

The normalized obstruction fraction is

$$\widehat{\eta}_D(r) := \begin{cases} \eta_D(r) / \|r\| , & r \neq 0, \\ 0, & r = 0. \end{cases} \tag{90}$$

The quotient class $[r]$ records the exact algebraic obstruction. The scalar $\eta_D(r)$ records its metric severity. The two should not be conflated: a nonzero obstruction may be arbitrarily small, while a large correctable defect may have zero obstruction.

Proposition 6.5 (Dual obstruction witness)**.** *For every $r \in V$,*

$$\eta_D(r) = \max_{\substack{y \in \ker D^* \\ \|y\| \leq 1}} \langle y, r \rangle . \tag{91}$$

If $r_{\mathrm{ob}} \neq 0$, the maximum is attained by

$$y^* = \frac{r_{\mathrm{ob}}}{\|r_{\mathrm{ob}}\|}. \tag{92}$$

Proof. For $y \in \ker D^*$, orthogonality gives $\langle y, r\rangle = \langle y, r_{\mathrm{ob}}\rangle$. Cauchy–Schwarz bounds this by $\|r_{\mathrm{ob}}\|$, with equality at the displayed unit vector. □

The vector y^* is a compact certificate: every authorized linear correction is invisible to it, yet the defect has nonzero projection along it.

6.3 Protected invariants and resource-aware correction

A repair is not authorized merely because it reduces task error. It must preserve declared invariants. Let

$$C : U \to Z \tag{93}$$

be the derivative of a protected invariant map. First-order invariant preservation requires $Cu = 0$, so the authorized tangent space is

$$U^{\mathrm{auth}} = \ker C. \tag{94}$$

Proposition 6.6 (Invariant-constrained obstruction)**.** *Let $D_C := D\,|_{\ker C}$. An exact first-order repair preserving the invariant exists if and only if*

$$[r] = 0 \quad \textit{in} \ \operatorname{coker} D_C. \tag{95}$$

Equivalently, the stacked system

$$\begin{bmatrix} D \\ C \end{bmatrix} u = \begin{bmatrix} -r \\ 0 \end{bmatrix} \tag{96}$$

is solvable.

Proof. The first statement is linear correctability on the restricted domain. The second rewrites the repair and invariant equations as one system. □

Thus a defect may be unconstrained-correctable but authorized-obstructed. This is the correct classification when a score can be improved only by deleting evidence, weakening provenance, exceeding an effect capability, or violating a protected safety or resource invariant.

Let $W : V \to V$ and $R : U \to U$ be positive-definite self-adjoint operators representing defect importance and repair cost.

Theorem 6.7 (Regularized minimum-cost repair)**.** *For $\lambda > 0$, the functional*

$$\mathcal{J}_\lambda(u) := \frac{1}{2}\,\langle Du + r, W(Du + r)\rangle + \frac{\lambda}{2}\,\langle u, Ru\rangle \tag{97}$$

has a unique minimizer

$$u_\lambda = -(D^*WD + \lambda R)^{-1} D^*Wr. \tag{98}$$

It is characterized by the normal equation

$$D^*W(Du_\lambda + r) + \lambda R u_\lambda = 0. \tag{99}$$

Proof. The Hessian $D^*WD + \lambda R$ is positive definite because R is positive definite and $\lambda > 0$. Hence $\mathcal{J}_\lambda$ is strictly convex and coercive. Differentiation gives the normal equation and its unique solution. □

Here λ is not a semantic permission to trade away hard invariants. It balances only those costs already declared soft. Hard invariants remain in the domain restriction.

6.4 Convex and one-sided authorization

Some repairs are not closed under negation. Examples include adding tests, allocating nonnegative compute, increasing redundancy, or modifying parameters subject to inequality constraints. Let $K \subseteq U$ be a nonempty closed convex authorization set and define

$$\eta_{D,K}(r) := \inf_{u \in K} \|Du + r\| . \tag{100}$$

Proposition 6.8 (Convex authorized correctability)**.** *If K is compact, or if $u \mapsto \|Du + r\|$ is coercive on K, then the infimum defining $\eta_{D,K}(r)$ is attained. Exact authorized correction exists if and only if*

$$\eta_{D,K}(r) = 0. \tag{101}$$

Proof. Continuity plus compactness gives attainment in the first case. Coercivity restricts every minimizing sequence to a compact sublevel set in finite dimension, giving the second. Exact correction is precisely zero residual. □

When K is a cone, noncorrectability admits a Farkas-type dual witness. Define the dual cone

$$K^* := \{z \in U : \langle z, u \rangle \geq 0 \text{ for all } u \in K\} . \tag{102}$$

Theorem 6.9 (Cone-constrained obstruction certificate)**.** *Let $K \subseteq U$ be a closed convex cone and assume $D(K)$ is closed. Then an authorized correction $u \in K$ with $Du = -r$ exists if and only if*

$$\langle y, r \rangle \leq 0 \tag{103}$$

for every $y \in V$ satisfying

$$D^* y \in K^*. \tag{104}$$

Consequently, any y with $D^*y \in K^*$ and $\langle y, r \rangle > 0$ certifies noncorrectability.

Proof. Set $C = D(K)$, a closed convex cone. Its dual is

$$C^* = \{y : D^* y \in K^*\} . \tag{105}$$

By the bipolar theorem, $-r \in C$ if and only if $\langle y, -r \rangle \geq 0$ for every $y \in C^*$, which is the displayed inequality. □

6.5 Sensitivity, conditioning, and empirical certificates

Let $\sigma_+(D)$ denote the smallest positive singular value of D. When $D \neq 0$,

$$\left\|D^\dagger\right\| = \frac{1}{\sigma_+(D)}. \tag{106}$$

Small $\sigma_+(D)$ means the correctable component is ill-conditioned: a small defect perturbation may require a large repair.

Proposition 6.10 (Fixed-map sensitivity)**.** *For defects $r, s \in V$,*

$$|\eta_D(r) - \eta_D(s)| \leq \|r - s\|, \tag{107}$$

$$\left\|D^\dagger r - D^\dagger s\right\| \leq \frac{1}{\sigma_+(D)} \|r - s\|. \tag{108}$$

Proof. The first inequality follows because distance to a fixed closed set is 1-Lipschitz. The second follows from the operator norm of $D^\dagger$. □

Proposition 6.11 (Transferred residual under model error)**.** *Let $\widehat{D} = D + \Delta D$ and $\widehat{r} = r + \Delta r$. For any proposed repair u,*

$$\left|\|Du + r\| - \left\|\widehat{D}u + \widehat{r}\right\|\right| \leq \|\Delta D\| \, \|u\| + \|\Delta r\|. \tag{109}$$

In particular, applying an exact repair for (D, r) to the perturbed system yields residual at most

$$\|\Delta D\| \, \|u\| + \|\Delta r\|. \tag{110}$$

Proof. The two residual vectors differ by $\Delta D\, u + \Delta r$. Apply the reverse triangle inequality and submultiplicativity. □

The next result turns uncertain estimates into a budget-bounded obstruction certificate.

Theorem 6.12 (Robust empirical noncorrectability certificate)**.** *Suppose, on an event of probability at least $1 - \alpha$,*

$$\left\|D - \widehat{D}\right\| \leq \varepsilon_D, \qquad \|r - \widehat{r}\| \leq \varepsilon_r. \tag{111}$$

Fix a repair budget $\|u\| \leq B$ and a unit vector $y \in V$. If

$$\langle y, \widehat{r}\rangle > \varepsilon_r + B\big(\left\|\widehat{D}^* y\right\| + \varepsilon_D\big), \tag{112}$$

then, on the same event, no budget-feasible repair satisfies $Du = -r$.

Proof. For every $\|u\| \leq B$,

$$\langle y, Du + r\rangle = \langle D^* y, u\rangle + \langle y, r\rangle \tag{113}$$

$$\geq -\|D^* y\| \; B + \langle y, \widehat{r}\rangle - \varepsilon_r \tag{114}$$

$$\geq -B\big(\left\|\widehat{D}^* y\right\| + \varepsilon_D\big) + \langle y, \widehat{r}\rangle - \varepsilon_r. \tag{115}$$

Condition (112) makes the final expression positive, so $Du + r$ cannot vanish. □

A negative result can therefore be proof-carrying: it records the witness y, the confidence event, the budget B, and the positive dual gap.

6.6 Obstruction shrinkage under repair expansion

Let V remain fixed while the authorized correction grammar expands.

Theorem 6.13 (Authorized obstruction-shrinkage theorem)**.** *Suppose*

$$U_t^{\mathrm{auth}} \subseteq U_{t+1}^{\mathrm{auth}} \tag{116}$$

and

$$D_{t+1}\,|_{U_t^{\mathrm{auth}}} = D_t. \tag{117}$$

Then

$$\operatorname{im} D_t \subseteq \operatorname{im} D_{t+1}, \tag{118}$$

there is a canonical surjection

$$q_t : \operatorname{coker} D_t \twoheadrightarrow \operatorname{coker} D_{t+1}, \qquad [v]_{\operatorname{im} D_t} \longmapsto [v]_{\operatorname{im} D_{t+1}}, \tag{119}$$

and

$$\dim \operatorname{coker} D_{t+1} \leq \dim \operatorname{coker} D_t. \tag{120}$$

The inequality is strict if and only if $\operatorname{im} D_t \subsetneq \operatorname{im} D_{t+1}$.

Proof. For $u \in U_t^{\mathrm{auth}}$, $D_t u = D_{t+1} u$, so every old correctable direction remains correctable. Quotienting by a larger subspace gives the canonical surjection. Rank-nullity gives the dimension statement. □

Definition 6.14 (Authorized repair rank)**.** The authorized repair rank and unresolved obstruction dimension are

$$\rho_t := \operatorname{rank} D_t, \qquad o_t := \dim \operatorname{coker} D_t. \tag{121}$$

Proposition 6.15 (Exact sequence of newly correctable directions)**.** *Under the assumptions of Theorem 6.13, there is a short exact sequence*

$$0 \longrightarrow \frac{\operatorname{im} D_{t+1}}{\operatorname{im} D_t} \longrightarrow \operatorname{coker} D_t \xrightarrow{q_t} \operatorname{coker} D_{t+1} \longrightarrow 0. \tag{122}$$

Consequently,

$$o_t - o_{t+1} = \dim \frac{\operatorname{im} D_{t+1}}{\operatorname{im} D_t}. \tag{123}$$

Proof. The kernel of q_t consists exactly of classes represented by vectors in $\operatorname{im} D_{t+1}$, modulo $\operatorname{im} D_t$. Surjectivity was proved above. The dimension identity follows from exactness. □

The quotient $\operatorname{im} D_{t+1}/\operatorname{im} D_t$ is the marginal structural contribution of the new repair grammar. It counts newly reachable defect directions, not merely new source-code branches.

6.7 Changing defect spaces

Open-set improvement may enlarge the defect ontology itself. Let

$$j_U : U_t^{\mathrm{auth}} \hookrightarrow U_{t+1}^{\mathrm{auth}}, \qquad j_V : V_t \hookrightarrow V_{t+1} \tag{124}$$

be injective embeddings satisfying

$$D_{t+1} j_U = j_V D_t. \tag{125}$$

Proposition 6.16 (Induced obstruction transport under ontology expansion)**.** *Equation* (125) *induces a linear map*

$$\overline{j}_t : \operatorname{coker} D_t \to \operatorname{coker} D_{t+1}, \qquad [v] \longmapsto [j_V v]. \tag{126}$$

Its kernel is canonically isomorphic to

$$\ker \overline{j}_t \cong \frac{j_V(V_t) \cap \operatorname{im} D_{t+1}}{j_V(\operatorname{im} D_t)}. \tag{127}$$

Thus an old obstruction vanishes after expansion exactly when the enlarged repair grammar corrects its embedded defect direction.

Proof. Commutation ensures that changing the representative by $D_t u$ changes $j_V v$ by $D_{t+1} j_U u$, so the map is well-defined. The formula for the kernel follows directly from the quotient definition. □

An evidence-conservative ontology expansion should make j_V explicit. Otherwise disappearance of an old defect may reflect semantic deletion rather than newly acquired correctability.

6.8 Packet-resolved correction and cross-packet leakage

Let

$$U = \bigoplus_{i=1}^{m} U_i, \qquad V = \bigoplus_{i=1}^{m} V_i, \tag{128}$$

with orthogonal projectors Q_i and P_i. Decompose

$$D = D_0 + E, \qquad D_0 := \sum_{i=1}^{m} P_i D Q_i. \tag{129}$$

The block-diagonal operator D_0 represents ideal packet-local correction; E contains all cross-packet effects.

Suppose each local block admits a right inverse and their direct sum gives

$$Q_0 : V \to U, \qquad D_0 Q_0 = I_V. \tag{130}$$

Theorem 6.17 (Exact correction under small packet coupling)**.** *If*

$$q := \|E Q_0\| < 1, \tag{131}$$

then D is surjective and has right inverse

$$Q = Q_0(I_V + EQ_0)^{-1} = Q_0 \sum_{k=0}^{\infty} (-EQ_0)^k. \tag{132}$$

Moreover,

$$\|Q\| \le \frac{\|Q_0\|}{1-q}. \tag{133}$$

For defect r, the truncated packet-correction series

$$u^{(n)} := -Q_0 \sum_{k=0}^{n} (-EQ_0)^k r \tag{134}$$

has residual

$$Du^{(n)} + r = (-EQ_0)^{n+1} r, \tag{135}$$

and hence

$$\left\| Du^{(n)} + r \right\| \le q^{n+1} \|r\|. \tag{136}$$

Proof. Since $\|EQ_0\| < 1$, the Neumann series for $(I + EQ_0)^{-1}$ converges. Also

$$DQ_0 = (D_0 + E)Q_0 = I + EQ_0, \tag{137}$$

so $DQ = I$. The norm bound follows from the geometric series. For the truncation,

$$(I + EQ_0) \sum_{k=0}^{n} (-EQ_0)^k = I - (-EQ_0)^{n+1}, \tag{138}$$

which gives the residual formula. □

This theorem formalizes a useful middle regime. Cross-packet effects need not vanish. They must be sufficiently small relative to the local correction inverse for iterative coordination to converge.

Definition 6.18 (Packet leakage profile)**.** The directed leakage from correction packet j into defect packet i is

$$\ell_{ij} := \|P_i D Q_j\|, \qquad i \ne j. \tag{139}$$

The matrix (ℓ_{ij}) is the packet leakage profile.

A repair receipt should record both the target block and this off-block profile. A patch that corrects its intended packet while strongly disturbing protected packets is not locally authorized merely because the total score improves.

6.9 Gauge-fixed nonlinear local repair by contraction

Let U, V be Banach spaces and let the post-correction defect map be

$$\mathcal{F}(u) = r + Du + N(u), \qquad N(0) = 0. \tag{140}$$

When D has a nontrivial kernel, a bounded right inverse does not make the full nonlinear repair unique: two roots can differ along neutral or symmetry directions. The contraction argument therefore requires a declared complement, or gauge, in which the right inverse selects one representative.

Theorem 6.19 (Gauge-fixed local nonlinear repair by contraction)**.** *Suppose $D : U \to V$ has a bounded right inverse $Q : V \to U$, so $DQ = I_V$, and set*

$$W := \operatorname{im} Q. \tag{141}$$

Then $D|_W : W \to V$ is an isomorphism with inverse Q. Let

$$B_R^W := \{w \in W : \|w\| \le R\}. \tag{142}$$

Suppose

$$\|N(u) - N(v)\| \le L_N \|u - v\| \qquad (u, v \in B_R^W), \tag{143}$$

with

$$q := \|Q\| L_N < 1, \tag{144}$$

and

$$\|Qr\| + \|Q\| \sup_{u \in B_R^W} \|N(u)\| \le R. \tag{145}$$

Then there exists a unique $u^* \in B_R^W$ satisfying $\mathcal{F}(u^*) = 0$. The gauge-fixed iteration

$$u_{k+1} = -Qr - QN(u_k) \tag{146}$$

converges to u^* from every $u_0 \in B_R^W$, with

$$\|u_k - u^*\| \le \frac{q^k}{1-q} \|u_1 - u_0\|. \tag{147}$$

Proof. Define $T(u) = -Qr - QN(u)$. Because T takes values in W, the radius condition gives $T(B_R^W) \subseteq B_R^W$. The Lipschitz condition gives

$$\|T(u) - T(v)\| \le q \|u - v\|. \tag{148}$$

Banach's fixed-point theorem yields a unique fixed point in B_R^W and the standard a posteriori convergence estimate. Applying D to the fixed-point equation and using $DQ = I_V$ gives $Du^* = -r - N(u^*)$. □

Remark 6.20 (Scope of uniqueness)**.** The theorem proves uniqueness only in the selected gauge $W = \operatorname{im} Q$. Other roots may exist elsewhere in U when $\ker D \neq 0$. Such roots must be compared through the later Lyapunov–Schmidt reduction, cost criteria, or explicit symmetry quotients rather than being ruled out by the contraction argument.

Proposition 6.21 (Nonlinear obstruction floor)**.** *Let P_{ob} be the orthogonal obstruction projector of D. For every u,*

$$\|\mathcal{F}(u)\| \ge \|P_{\mathrm{ob}} r\| - \|P_{\mathrm{ob}} N(u)\|. \tag{149}$$

Consequently, if $\|P_{\mathrm{ob}} N(u)\| \le \gamma$ on an authorized region, then no repair in that region can achieve

residual below

$$\max\{0, \eta_D(r) - \gamma\}. \tag{150}$$

Proof. Since $P_{\mathrm{ob}} D = 0$,

$$P_{\mathrm{ob}}\mathcal{F}(u) = P_{\mathrm{ob}} r + P_{\mathrm{ob}} N(u). \tag{151}$$

Apply the reverse triangle inequality and $\|\mathcal{F}(u)\| \geq \|P_{\mathrm{ob}}\mathcal{F}(u)\|$. □

A first-order obstruction can sometimes be canceled by nonlinear terms. The proposition quantifies how large that nonlinear obstruction-channel effect must be.

6.10 Newton–Kantorovich repair

The gauge-fixed contraction theorem assumes a fixed linear right inverse and a selected complement. Newton iteration permits a changing derivative.

Theorem 6.22 (Newton–Kantorovich repair criterion)**.** *Let $F : U \to V$ be continuously Fréchet differentiable between Banach spaces. Let $u_0 \in U$ and suppose $F'(u_0)$ is invertible with inverse B. Assume that on the closed ball $\overline{B}(u_0, R)$,*

$$\|F'(u) - F'(v)\| \leq L\,\|u - v\|. \tag{152}$$

Set

$$\eta := \|BF(u_0)\|, \qquad h := \|B\|\, L\eta. \tag{153}$$

If $h \leq \frac{1}{2}$ and

$$t_* := \frac{1 - \sqrt{1 - 2h}}{\|B\|\, L} \leq R, \tag{154}$$

then Newton iteration

$$u_{n+1} = u_n - F'(u_n)^{-1} F(u_n) \tag{155}$$

is well-defined, remains in $\overline{B}(u_0, t_*)$, and converges to a root u^* of F. If the larger radius

$$t^{**} := \frac{1 + \sqrt{1 - 2h}}{\|B\|\, L} \tag{156}$$

is contained in the regularity region, the root is unique in $\overline{B}(u_0, t^{**})$.

Proof. Consider the scalar majorant

$$f(t) = \eta - t + \frac{\|B\|\, L}{2} t^2. \tag{157}$$

The condition $h \leq 1/2$ gives the two nonnegative roots t_* and t^{**}. The Lipschitz derivative bound and the Banach lemma imply that $F'(u)$ remains invertible while $\|u - u_0\| < 1/(\|B\|\, L)$. Standard Newton majorization shows that the increments of the operator iteration are dominated by those of scalar Newton iteration for f, whose partial sums remain below t_* and converge. Hence (u_n) is Cauchy, its limit lies in the ball, and continuity gives $F(u^*) = 0$. The same majorant estimate gives uniqueness in the larger ball. □

The criterion makes three quantities auditable: the initial predicted correction size η, inverse sensitivity $\|B\|$, and derivative curvature L.

6.11 Singular nonlinear repair and Lyapunov–Schmidt reduction

When $D = F'(0)$ is not surjective, the nonlinear problem is not decided by a global right inverse. The correct local object is a reduced obstruction equation.

Theorem 6.23 (Lyapunov–Schmidt repair reduction)**.** *Let $F : P \times U \to V$ be C^k, $k \geq 2$, between finite-dimensional spaces, with $F(0,0) = 0$. Let*

$$D := \partial_u F(0,0). \tag{158}$$

Choose splittings

$$U = K \oplus X, \qquad K = \ker D, \tag{159}$$

and

$$V = R \oplus C, \qquad R = \operatorname{im} D, \tag{160}$$

so that $D|_X \colon X \to R$ is an isomorphism. Let P_R and P_C be the corresponding projections. Then there are neighborhoods of the origin and a unique C^k map

$$\chi : P \times K \to X, \qquad \chi(0,0) = 0, \tag{161}$$

such that

$$P_R F(p, k + \chi(p,k)) = 0. \tag{162}$$

Define the reduced obstruction map

$$\psi(p,k) := P_C F(p, k + \chi(p,k)). \tag{163}$$

A sufficiently small pair (p,u) solves $F(p,u) = 0$ if and only if

$$u = k + \chi(p,k) \tag{164}$$

for some $k \in K$ satisfying

$$\psi(p,k) = 0. \tag{165}$$

Proof. Apply the implicit-function theorem to the range equation

$$G(p,k,x) := P_R F(p, k + x) = 0. \tag{166}$$

Its derivative in x at the origin is $D|_X \colon X \to R$, an isomorphism, so there is a unique local solution $x = \chi(p,k)$. Substitution solves the range component identically. The remaining equation is exactly the cokernel component $\psi(p,k) = 0$. □

The reduction shows why "vanishing linear obstruction" is not the end of the analysis. Kernel freedom k may be used to solve the reduced cokernel equation, may produce several branches, or may fail completely.

6.12 Higher-order obstruction classes

Consider the specific formal correction equation

$$F(\varepsilon, u) = \varepsilon r_1 + Du + \frac{1}{2}B(u,u) + O(\varepsilon^3 + \|u\|^3), \tag{167}$$

where $B : U \times U \to V$ is symmetric bilinear. Seek

$$u(\varepsilon) = \varepsilon u_1 + \varepsilon^2 u_2 + O(\varepsilon^3). \tag{168}$$

Proposition 6.24 (Second-order repair obstruction)**.** *A repair through order ε^2 exists only if*

$$Du_1 = -r_1, \tag{169}$$

$$\left[\frac{1}{2}B(u_1,u_1)\right] = 0 \quad \text{in } \operatorname{coker} D. \tag{170}$$

When the second class vanishes, u_2 may be chosen to solve

$$Du_2 = -\frac{1}{2}B(u_1,u_1). \tag{171}$$

If $\ker D \neq 0$, replacing u_1 by $u_1 + k$ can change the second-order obstruction class.

Proof. Substitute the formal series into (167) and equate coefficients of ε and ε^2. The first equation is the linear repair equation. The second is solvable exactly when its right-hand side has zero class in $\operatorname{coker} D$. □

Theorem 6.25 (Finite formal obstruction tower)**.** *Let $F(\varepsilon, u)$ be analytic near $(0,0)$ with $F(0,0) = 0$ and $D = \partial_u F(0,0)$. Suppose coefficients $u_1, \dots, u_{n-1}$ have been chosen so that*

$$F\left(\varepsilon, \sum_{j=1}^{n-1} \varepsilon^j u_j\right) = \varepsilon^n b_n + O(\varepsilon^{n+1}). \tag{172}$$

Then extending the repair to order n by adding $\varepsilon^n u_n$ is possible if and only if

$$[b_n] = 0 \quad \text{in } \operatorname{coker} D. \tag{173}$$

When it vanishes, any solution of $Du_n = -b_n$ produces an order-n lift.

Proof. Analytic expansion gives

$$F\left(\varepsilon, \sum_{j=1}^{n} \varepsilon^j u_j\right) = \varepsilon^n (b_n + Du_n) + O(\varepsilon^{n+1}), \tag{174}$$

because u_n first enters linearly at order n. Cancellation of the coefficient is equivalent to solvability of $Du_n = -b_n$. □

The sequence

$$[b_1], [b_2], [b_3], \dots \in \operatorname{coker} D \tag{175}$$

is an obstruction tower. Choices in $\ker D$ at early orders can alter later classes, so a repair search must retain branch information rather than greedily fixing one representative.

7 Local-to-Global Repair Descent

7.1 Repair sheaves and local solution torsors

A heterogeneous agent may admit valid repairs in every stratum while those repairs remain mutually incompatible. Let X be a finite site of functional regions and let

$$\mathfrak{U} = \{U_\alpha\}_{\alpha \in I} \tag{176}$$

be a finite cover. Let $\mathscr{U}$ be a sheaf of authorized corrections, $\mathscr{V}$ a sheaf of defects, and

$$D : \mathscr{U} \to \mathscr{V} \tag{177}$$

a morphism of sheaves. Let $r \in \Gamma(X, \mathscr{V})$ be a global defect.

A local correction on U_α is a section $u_\alpha \in \mathscr{U}(U_\alpha)$ satisfying

$$D_\alpha u_\alpha = -r|_{U_\alpha}. \tag{178}$$

If one local correction exists, the local solution set is an affine space over $\ker D(U_\alpha)$.

Proposition 7.1 (Local solution torsor)**.** *For each U_α, the set*

$$\mathscr{S}_\alpha := \{u \in \mathscr{U}(U_\alpha) : D_\alpha u = -r|_{U_\alpha}\} \tag{179}$$

is either empty or a torsor for the additive group $\ker D(U_\alpha)$. If a global repair exists, the global solution set is a torsor for $\Gamma(X, \ker D)$.

Proof. The difference of two solutions lies in the kernel, and adding a kernel section to a solution preserves the equation. The action is free and transitive. □

This torsor viewpoint matters because a local repair is rarely unique. Kernel freedom is the resource used to reconcile interface assumptions.

7.2 Exact linear descent

Theorem 7.2 (Linear repair-descent theorem)**.** *Assume a local correction u_α exists on every U_α. On overlaps $U_{\alpha\beta} = U_\alpha \cap U_\beta$, define*

$$d_{\alpha\beta} := u_\alpha|_{U_{\alpha\beta}} - u_\beta|_{U_{\alpha\beta}}. \tag{180}$$

Then:

1. *$d_{\alpha\beta} \in \ker D(U_{\alpha\beta})$;*
2. *$(d_{\alpha\beta})$ is a Čech 1-cocycle with coefficients in $\ker D$;*
3. *a global correction $u \in \Gamma(X, \mathscr{U})$ with $Du = -r$ exists if and only if $(d_{\alpha\beta})$ is a Čech coboundary relative to $\mathfrak{U}$.*

If the cover computes first cohomology, the final condition is equivalent to

$$[d_{\alpha\beta}] = 0 \quad \textit{in} \quad \check{H}^1(X, \ker D). \tag{181}$$

Proof. On every overlap,

$$D(d_{\alpha\beta}) = -r|_{U_{\alpha\beta}} + r|_{U_{\alpha\beta}} = 0. \tag{182}$$

On triple overlaps,

$$d_{\alpha\beta} + d_{\beta\gamma} + d_{\gamma\alpha} = 0, \tag{183}$$

so the family is a cocycle.

If a global repair u exists, set $k_\alpha = u|_{U_\alpha} - u_\alpha$. Then $Dk_\alpha = 0$ and

$$d_{\alpha\beta} = k_\beta - k_\alpha, \tag{184}$$

so the cocycle is a coboundary. Conversely, if $d_{\alpha\beta} = k_\beta - k_\alpha$ for kernel sections k_α, then $u'_\alpha = u_\alpha + k_\alpha$ agree on overlaps. The sheaf gluing axiom yields a global u, and $Du = -r$ because the adjustments lie in the kernel. □

Definition 7.3 (Two-level repair obstruction)**.** The local obstruction on U_α is

$$[r|_{U_\alpha}] \in \operatorname{coker} D_\alpha. \tag{185}$$

When every local class vanishes, the global gluing obstruction is

$$[d_{\alpha\beta}] \in \check{H}^1(\mathfrak{U}, \ker D). \tag{186}$$

Proposition 7.4 (Choice independence of the gluing class)**.** *The cohomology class $[d_{\alpha\beta}]$ does not depend on the particular family of local repairs used to compute it.*

Proof. If $u'_\alpha = u_\alpha + k_\alpha$ with $k_\alpha \in \ker D(U_\alpha)$, then

$$d'_{\alpha\beta} = d_{\alpha\beta} + k_\alpha - k_\beta. \tag{187}$$

Thus the cocycles differ by a coboundary. □

Repair failure now has two distinct meanings:

1. a chart cannot correct its own defect;
2. every chart can correct locally, but the chosen repairs cannot be made globally coherent.

7.3 Graph-Hodge decomposition of approximate gluing

The exact Čech class is binary. Empirical repairs produce noisy overlap mismatches. A finite graph model gives a quantitative analogue.

Let $G = (I, E)$ be a connected oriented graph representing the nerve of the cover, let K be a finite-dimensional Hilbert space representing a common trivialization of the kernel fibers, and let

$$e = (e_a)_{a \in E} \in K^E \tag{188}$$

be the transported overlap-mismatch vector. Let $B \in \mathbb{R}^{|I|\times|E|}$ be the oriented incidence matrix and set

$$\mathbf{B} := B \otimes I_K. \tag{189}$$

A vertex gauge $k \in K^I$ changes the mismatch to

$$e + \mathbf{B}^* k. \tag{190}$$

Theorem 7.5 (Quantitative graph-Hodge gluing)**.** *Let*

$$L_G := BB^* \tag{191}$$

be the graph Laplacian and let P_{cyc} be the orthogonal projector from K^E onto

$$\ker \mathbf{B}, \tag{192}$$

the cycle space. Then

$$\inf_{k\in K^I} \|e + \mathbf{B}^* k\| = \|P_{\mathrm{cyc}} e\|. \tag{193}$$

The mean-zero minimum-norm gauge is

$$k^* = -(L_G^\dagger B \otimes I_K)e. \tag{194}$$

If $\lambda_2(L_G) > 0$ is the spectral gap, then

$$\|k^*\| \le \frac{1}{\sqrt{\lambda_2(L_G)}} \|e\|. \tag{195}$$

Exact graph-level gluing is possible if and only if

$$P_{\mathrm{cyc}} e = 0. \tag{196}$$

Proof. The finite-dimensional Hodge decomposition is

$$K^E = \operatorname{im} \mathbf{B}^* \oplus \ker \mathbf{B}. \tag{197}$$

A gauge can remove exactly the gradient component in $\operatorname{im} \mathbf{B}^*$, leaving the cycle projection. The normal equation for the least-squares gauge is

$$\mathbf{B}\mathbf{B}^* k = -\mathbf{B}e, \tag{198}$$

whose mean-zero minimum-norm solution is the displayed pseudoinverse expression. The smallest nonzero singular value of $\mathbf{B}^*$ is $\sqrt{\lambda_2(L_G)}$, giving the bound. □

Definition 7.6 (Quantitative gluing obstruction)**.** The graph-level quantitative gluing obstruction is

$$\eta_{\mathrm{glue}}(e) := \|P_{\mathrm{cyc}} e\|. \tag{199}$$

The local-to-global repair certificate is the pair

$$(\eta_{\mathrm{loc}}, \eta_{\mathrm{glue}}), \qquad \eta_{\mathrm{loc}}^2 := \sum_{\alpha} \|D_\alpha u_\alpha + r_\alpha\|^2 . \tag{200}$$

A small local residual does not imply a small gluing residual. Conversely, a perfectly coherent family can consistently implement the wrong local repair. Both coordinates are required.

Corollary 7.7 (Trees have no cycle obstruction)**.** *If the cover nerve is a tree, then* $\ker \mathbf{B} = 0$ *and every overlap mismatch can be removed by a vertex gauge. Any remaining failure is local rather than cyclic.*

Proof. A connected tree has $|E| = |I| - 1$ and incidence matrix of full column rank, so its cycle space is zero. □

7.4 Weighted overlap confidence

Different overlaps may have different uncertainty or semantic importance. Let W_E be a positive-definite operator on K^E. Define

$$\eta_{\mathrm{glue},W}(e) := \inf_k \|e + \mathbf{B}^* k\|_{W_E} . \tag{201}$$

Proposition 7.8 (Weighted gluing normal equation)**.** *Every weighted least-squares gauge satisfies*

$$\mathbf{B} W_E \mathbf{B}^* k = -\mathbf{B} W_E e. \tag{202}$$

The mean-zero minimum-norm minimizer is

$$k_W^* = -(\mathbf{B} W_E \mathbf{B}^*)^\dagger \mathbf{B} W_E e. \tag{203}$$

The irreducible weighted mismatch is the W_E*-orthogonal projection of* e *onto the* W_E*-orthogonal complement of* $\operatorname{im} \mathbf{B}^*$.

Proof. Differentiate the quadratic objective

$$\frac{1}{2} \langle e + \mathbf{B}^* k, W_E (e + \mathbf{B}^* k) \rangle \tag{204}$$

with respect to k. The normal equation follows. Its Moore–Penrose solution is the unique minimizer orthogonal to the constant vertex-gauge kernel. □

This permits confidence-weighted descent: weakly identified interfaces need not dominate a certificate merely because they have large raw coordinate scale.

7.5 Finite cycle obstruction

Example 7.9 (Locally repairable, globally incompatible)**.** Let the nerve be a three-cycle with no triple intersection. Suppose the kernel fiber is $\mathbb{R}$ and the oriented differences are

$$d_{12} = 1, \qquad d_{23} = 1, \qquad d_{31} = 1. \tag{205}$$

If the cocycle were a coboundary, there would be k_1, k_2, k_3 with

$$d_{12} = k_2 - k_1, \quad d_{23} = k_3 - k_2, \quad d_{31} = k_1 - k_3. \tag{206}$$

Summing gives $3 = 0$, a contradiction. Hence no global correction exists. The graph-Hodge obstruction has norm $\sqrt{3}$.

In an agent, the three charts might represent planner, memory, and evaluator repairs. Each patch can pass isolated tests while the cycle of interface assumptions fails to close.

7.6 Nonlinear descent through symmetry torsors

For nonlinear repair, local solution sets need not be affine. A tractable case occurs when each local solution set is a principal homogeneous space for a local symmetry group.

Let $\mathscr{G}$ be a sheaf of groups acting freely and transitively on the sheaf of local nonlinear solutions $\mathscr{S}$. Choose local solutions $s_\alpha \in \mathscr{S}(U_\alpha)$. On overlaps there are unique transition elements

$$g_{\alpha\beta} \in \mathscr{G}(U_{\alpha\beta}) \tag{207}$$

such that

$$s_\alpha = g_{\alpha\beta} \cdot s_\beta. \tag{208}$$

Conditional Theorem 7.10 (Nonlinear torsor descent)**.** *Assume the local solution action is principal and effective descent holds for $\mathscr{G}$-torsors on the declared cover. Then a global nonlinear repair exists if and only if the transition family satisfies the nonabelian cocycle condition*

$$g_{\alpha\beta} g_{\beta\gamma} g_{\gamma\alpha} = e \tag{209}$$

on triple overlaps and its class is trivial in nonabelian Čech cohomology

$$[g_{\alpha\beta}] = 1 \quad \textit{in} \quad \check{H}^1(\mathfrak{U}, \mathscr{G}). \tag{210}$$

Equivalently, there exist local gauges $g_\alpha \in \mathscr{G}(U_\alpha)$ such that

$$g_{\alpha\beta} = g_\alpha^{-1} g_\beta. \tag{211}$$

Proof. If a global solution exists, compare its restriction with each s_α to obtain gauges g_α; their ratios give the transition functions. Conversely, triviality of the cocycle permits gauge-adjusted local solutions $g_\alpha \cdot s_\alpha$ that agree on overlaps. Effective descent glues them to a global solution. □

The theorem is conditional because arbitrary nonlinear repair spaces need not be torsors, and arbitrary software interfaces need not satisfy effective descent.

7.7 Spanning-tree holonomy test

A finite operational version avoids explicit cohomology. Let $G = (I, E)$ be connected, with sets of local solutions S_i and invertible transition maps

$$T_{ij} : S_i \to S_j, \qquad T_{ji} = T_{ij}^{-1}. \tag{212}$$

Choose a root o and a spanning tree $T \subseteq G$.

Theorem 7.11 (Fundamental-cycle holonomy criterion)**.** *For each root state $s_o \in S_o$, tree transport determines a unique family (s_i^T). For every non-tree edge $e = (i, j)$, let $H_e : S_o \to S_o$ be transport around the corresponding fundamental cycle. The tree family extends to a globally compatible family on all edges if and only if*

$$H_e(s_o) = s_o \tag{213}$$

for every non-tree edge e.

Proof. Tree edges are compatible by construction. A non-tree edge closes one fundamental cycle. Its compatibility is exactly the statement that transport from the root to i, across e, and back to the root returns s_o. Fundamental cycles generate all graph cycles, so satisfying every chord condition gives compatibility on all edges. □

Definition 7.12 (Nonlinear holonomy residual)**.** If the S_i are metric spaces, define

$$\eta_{\mathrm{hol}}(s_o) := \max_{e \in E \setminus T} d\big(H_e(s_o), s_o\big). \tag{214}$$

This is a direct finite test for path dependence of repair semantics. It is distinct from the evaluator holonomy developed later, though mathematically analogous.

7.8 Order-by-order global repair

The local nonlinear tower and the descent obstruction combine into a two-axis lifting problem. Suppose each chart has an analytic local repair equation with the same linearized sheaf morphism D.

At order n, let

$$b_{n,\alpha} \in \mathscr{V}(U_\alpha) \tag{215}$$

be the residual coefficient determined by lower-order choices. The local obstruction is

$$[b_{n,\alpha}] \in \operatorname{coker} D_\alpha. \tag{216}$$

If every local class vanishes, choose $u_{n,\alpha}$ with

$$D_\alpha u_{n,\alpha} = -b_{n,\alpha}. \tag{217}$$

Their differences define an order-n gluing class in $\check{H}^1(\ker D)$.

Theorem 7.13 (Finite local-to-global obstruction tower)**.** *Fix $N \in \mathbb{N}$. Assume:*

1. *the analytic local repair equations agree under restriction;*
2. *the sheaves $\mathscr{U}$ and $\mathscr{V}$ satisfy effective linear descent on the declared cover;*
3. *repairs have been constructed globally through order $n - 1$.*

Then a global repair extends through order n if and only if:

1. *every local coefficient class vanishes,*

$$[b_{n,\alpha}] = 0 \quad \text{in } \operatorname{coker} D_\alpha; \tag{218}$$

2. for one, hence every, choice of local lifts, the overlap class vanishes,

$$[d^{(n)}_{\alpha\beta}] = 0 \quad \textit{in} \quad \check{H}^1(\mathfrak{U}, \ker D). \tag{219}$$

If both conditions hold for every $1 \le n \le N$, a global formal repair exists through order N.

Proof. At order n, coefficient matching reduces the local problem to the linear equation $D_\alpha u_{n,\alpha} = -b_{n,\alpha}$, so the local cokernel classes are the complete local obstruction. When they vanish, Theorem 7.2 gives the complete criterion for gluing the order-n coefficients. Induction from 1 through N constructs the global formal series. □

The theorem produces an obstruction ledger

order	local obstruction	gluing obstruction
1	$[b_{1,\alpha}] \in \operatorname{coker} D_\alpha$	$[d^{(1)}] \in \check{H}^1(\ker D)$
2	$[b_{2,\alpha}] \in \operatorname{coker} D_\alpha$	$[d^{(2)}] \in \check{H}^1(\ker D)$
⋮	⋮	⋮
N	$[b_{N,\alpha}] \in \operatorname{coker} D_\alpha$	$[d^{(N)}] \in \check{H}^1(\ker D)$.

(220)

Higher categorical repair interfaces may introduce additional coherence obstructions beyond this two-column ledger. Those are not proved here.

7.9 Cover refinement and diagnostic soundness

Proposition 7.14 (Soundness of cover-relative obstruction)**.** *If a global repair exists, then every local obstruction and every exact gluing obstruction vanishes on every cover. Therefore, a nonzero local or gluing class on any valid cover is a sound certificate of global nonexistence.*

Proof. Restrict the global repair to each chart. The local equations hold, and the local restrictions agree exactly on overlaps, producing zero cocycles. □

The converse depends on the sheaf and cover assumptions of Theorem 7.2. A poor atlas can obscure causal structure even when its internal descent computation is correct. Atlas discovery therefore remains logically prior to universal claims about repair.

Definition 7.15 (Cover adequacy score)**.** For a candidate functional cover $\mathfrak{U}$, define the diagnostic score

$$\mathcal{A}(\mathfrak{U}) := \eta_{\mathrm{loc}}(\mathfrak{U}) + \tau\eta_{\mathrm{glue}}(\mathfrak{U}) + \lambda\,\mathrm{Complexity}(\mathfrak{U}) + \gamma\,\mathrm{Instability}(\mathfrak{U}). \tag{221}$$

This is a model-selection objective, not a theorem that the minimizing cover is uniquely correct. Its purpose is to expose the tradeoff between local fit, global coherence, atlas complexity, and instability under resampling or intervention.

8 Repair Transport and Transfer Geometry

A repair discovered in one system becomes recursively useful only when the system can determine whether the repair remains meaningful after a change of architecture, representation, task coordinates,

or evaluator. Transport is therefore not an auxiliary convenience. It is the mechanism by which a local correction becomes a reusable element of the repair calculus.

The theory separates four logically distinct questions:

1. does the correction equation commute with the proposed representation maps;
2. does the transport preserve authorization, invariants, and resource budgets;
3. how do transport errors accumulate along a path through an atlas;
4. does the transported correction retain positive externally measured gain?

The answers need not coincide. A correction may commute algebraically while violating an authorization cone. It may remain authorized while losing gain under evaluator mismatch. It may work along one transport path and fail along another because of nontrivial holonomy.

8.1 Correction data and transport morphisms

Definition 8.1 (Correction datum)**.** A finite-dimensional correction datum is a tuple

$$\mathfrak{C} = (U, V, D, \mathcal{A}, c, \mathfrak{p}), \tag{222}$$

where U is a correction space, V is a defect space, $D : U \to V$ is the linearized defect-response map, $\mathcal{A} \subseteq U$ is the authorized correction set, $c : U \to [0, \infty]$ is a declared resource cost, and $\mathfrak{p}$ is a collection of protected predicates or invariant maps.

Definition 8.2 (Transport datum)**.** Let $\mathfrak{C}_A$ and $\mathfrak{C}_B$ be correction data. A transport datum from A to B is a pair

$$\mathfrak{T}_{A\to B} = (T_U, T_V), \qquad T_U : U_A \to U_B, \qquad T_V : V_A \to V_B. \tag{223}$$

Its correction-square defect is

$$\Delta_D(\mathfrak{T}_{A\to B}) := D_B T_U - T_V D_A. \tag{224}$$

For declared source and target defects $r_A \in V_A$ and $r_B \in V_B$, its defect-identification error is

$$\Delta_r(\mathfrak{T}_{A\to B}; r_A, r_B) := r_B - T_V r_A. \tag{225}$$

The datum is *exact* when both quantities vanish.

The two errors answer different questions. Δ_D measures failure of the correction mechanism to commute. Δ_r measures failure of the target defect to be the transported source defect. A small value of one does not imply a small value of the other.

Theorem 8.3 (Exact repair transport)**.** *Suppose*

$$D_B T_U = T_V D_A \tag{226}$$

and

$$r_B = T_V r_A. \tag{227}$$

If u_A satisfies

$$D_A u_A = -r_A, \tag{228}$$

then

$$u_B := T_U u_A \tag{229}$$

satisfies

$$D_B u_B = -r_B. \tag{230}$$

Proof. By commutation,

$$D_B u_B = D_B T_U u_A = T_V D_A u_A = -T_V r_A = -r_B. \tag{231}$$

□

Corollary 8.4 (Kernel and obstruction naturality)**.** *Assume $D_B T_U = T_V D_A$. Then*

$$T_U(\ker D_A) \subseteq \ker D_B \tag{232}$$

and

$$T_V(\operatorname{im} D_A) \subseteq \operatorname{im} D_B. \tag{233}$$

Consequently, T_V induces a quotient map

$$\overline{T}_V : \operatorname{coker} D_A \to \operatorname{coker} D_B, \qquad [r] \mapsto [T_V r]. \tag{234}$$

Proof. If $D_A u = 0$, then $D_B T_U u = T_V D_A u = 0$. If $v = D_A u$, then $T_V v = D_B T_U u \in \operatorname{im} D_B$. The quotient map is therefore well-defined. □

Theorem 8.5 (Exact transport equivalence)**.** *Suppose T_U and T_V are isomorphisms and*

$$D_B = T_V D_A T_U^{-1}. \tag{235}$$

Then T_U restricts to an isomorphism

$$\ker D_A \cong \ker D_B, \tag{236}$$

and T_V induces an isomorphism

$$\operatorname{coker} D_A \cong \operatorname{coker} D_B. \tag{237}$$

Thus exact transport equivalence preserves both correction freedom and obstruction dimension.

Proof. The kernel statement follows from Theorem 8.4 applied in both directions. Since T_U is surjective,

$$\operatorname{im} D_B = T_V D_A T_U^{-1}(U_B) = T_V(\operatorname{im} D_A). \tag{238}$$

Hence T_V maps $\operatorname{im} D_A$ bijectively onto $\operatorname{im} D_B$ and descends to an isomorphism of quotient spaces. □

Definition 8.6 (Protected obstruction subspace)**.** A protected obstruction subspace is a declared subspace

$$Q_A \subseteq \operatorname{coker} D_A \tag{239}$$

whose nonzero classes must remain detectable after transport. The transport is *obstruction-faithful on Q_A* when

$$\ker \overline{T}_V \cap Q_A = \{0\}. \tag{240}$$

Proposition 8.7 (Representation loss can hide an obstruction)**.** *If $\overline{T}_V$ is not injective, then there exists a defect r_A such that*

$$[r_A] \neq 0 \quad \textit{but} \quad [T_V r_A] = 0. \tag{241}$$

Thus the target representation classifies the transported defect as correctable even though the source obstruction is nonzero. Obstruction faithfulness is therefore a necessary condition whenever the erased class is protected.

Proof. Choose a nonzero class in $\ker \overline{T}_V$ and any representative r_A. By definition its image in $\operatorname{coker} D_B$ vanishes. □

Remark 8.8. A noninjective bridge may be useful as a compression map, but it cannot by itself certify preservation of every repair obstruction. This is the transport analogue of the distinction between exact representation change and information-losing abstraction.

8.2 Authorization, invariants, and minimum-cost correction

Definition 8.9 (Authorized transport)**.** A transport datum is authorized when

$$T_U(\mathcal{A}_A) \subseteq \mathcal{A}_B. \tag{242}$$

It is (c_A, c_B, κ)-budget controlled when

$$c_B(T_U u) \leq \kappa c_A(u) \qquad \text{for every } u \in \mathcal{A}_A. \tag{243}$$

Proposition 8.10 (Authorized exact repair transport)**.** *Under the hypotheses of Theorem 8.3, if $u_A \in \mathcal{A}_A$ and the transport is authorized, then $u_B = T_U u_A$ is an authorized exact repair in system B. If, in addition, $c_A(u_A) \leq B$ and the transport is (c_A, c_B, κ)-budget controlled, then*

$$c_B(u_B) \leq \kappa B. \tag{244}$$

Proof. Authorization follows from $T_U(\mathcal{A}_A) \subseteq \mathcal{A}_B$. Exact repair follows from Theorem 8.3. The cost inequality is the definition of budget control. □

Let protected linear invariants be represented by

$$C_A : U_A \to Z_A, \qquad C_B : U_B \to Z_B. \tag{245}$$

A correction preserves the invariant when $C_A u = 0$ or $C_B u = 0$.

Proposition 8.11 (Invariant-preserving transport)**.** *Suppose there is a linear map $S : Z_A \to Z_B$ satisfying*

$$C_B T_U = S C_A. \tag{246}$$

Then

$$T_U(\ker C_A) \subseteq \ker C_B. \tag{247}$$

Consequently, every source repair preserving the declared invariant transports to a target repair preserving the transported invariant.

Proof. For $u \in \ker C_A$,

$$C_B T_U u = S C_A u = 0. \tag{248}$$

□

The previous propositions guarantee feasibility, not optimality. Under unitary transport, the canonical minimum-norm correction is also preserved.

Theorem 8.12 (Naturality of the Moore–Penrose repair)**.** *Let U_A, U_B, V_A, V_B be finite-dimensional Hilbert spaces. Suppose T_U and T_V are unitary and*

$$D_B = T_V D_A T_U^*. \tag{249}$$

Then

$$D_B^\dagger = T_U D_A^\dagger T_V^*. \tag{250}$$

For every $r_B = T_V r_A$, the target minimum-norm least-squares correction is the transport of the source minimum-norm least-squares correction:

$$-D_B^\dagger r_B = T_U(-D_A^\dagger r_A). \tag{251}$$

The orthogonal obstruction projectors satisfy

$$I - D_B D_B^\dagger = T_V(I - D_A D_A^\dagger) T_V^*. \tag{252}$$

Proof. Set $X = T_U D_A^\dagger T_V^*$. Direct substitution, unitary cancellation, and the four Moore–Penrose equations for $D_A^\dagger$ show that X satisfies the four Moore–Penrose equations for D_B. Uniqueness gives $X = D_B^\dagger$. The correction and projector identities follow immediately. □

Remark 8.13. For nonunitary isomorphisms, exact repairability still transports, but minimum norm is coordinate dependent. A change of metric or an explicit condition-number penalty is then required before comparing repair costs.

8.3 Composition and accumulation of transport defect

Let

$$\mathfrak{T}_{A\to B} = (T_U^{AB}, T_V^{AB}), \qquad \mathfrak{T}_{B\to C} = (T_U^{BC}, T_V^{BC}). \tag{253}$$

The composed transport is

$$\mathfrak{T}_{A\to C} = (T_U^{BC} T_U^{AB}, T_V^{BC} T_V^{AB}). \tag{254}$$

Lemma 8.14 (Composition identity)**.** *The correction-square defects obey*

$$\Delta_D(\mathfrak{T}_{A\to C}) = \Delta_D(\mathfrak{T}_{B\to C}) T_U^{AB} + T_V^{BC} \Delta_D(\mathfrak{T}_{A\to B}). \tag{255}$$

The defect-identification errors obey

$$\Delta_r(\mathfrak{T}_{A\to C}) = \Delta_r(\mathfrak{T}_{B\to C}) + T_V^{BC} \Delta_r(\mathfrak{T}_{A\to B}). \tag{256}$$

Proof. Insert and subtract $T_V^{BC} D_B T_U^{AB}$ in the first expression and $T_V^{BC} r_B$ in the second. □

Theorem 8.15 (Pathwise transport bound)**.** *Consider a path of systems*

$$A_0 \xrightarrow{\mathfrak{T}_0} A_1 \xrightarrow{\mathfrak{T}_1} \cdots \xrightarrow{\mathfrak{T}_{n-1}} A_n. \tag{257}$$

Write $T_{U,k} : U_k \to U_{k+1}$ *and* $T_{V,k} : V_k \to V_{k+1}$*. Let*

$$\|D_{k+1}T_{U,k} - T_{V,k}D_k\| \le \varepsilon_{D,k} \tag{258}$$

and

$$\|r_{k+1} - T_{V,k}r_k\| \le \varepsilon_{r,k}. \tag{259}$$

If $D_0u_0 = -r_0$ *and*

$$u_n = T_{U,n-1}\cdots T_{U,0}u_0, \tag{260}$$

then

$$\begin{aligned}\|D_nu_n + r_n\| \le &\sum_{k=0}^{n-1} \|T_{V,n-1}\cdots T_{V,k+1}\|\,\varepsilon_{D,k}\,\|T_{U,k-1}\cdots T_{U,0}u_0\| \\ &+ \sum_{k=0}^{n-1} \|T_{V,n-1}\cdots T_{V,k+1}\|\,\varepsilon_{r,k},\end{aligned} \tag{261}$$

where empty products are identities.

Proof. Repeatedly apply Theorem 8.14. The total correction-square defect is the telescoping sum

$$\sum_{k=0}^{n-1} T_{V,n-1}\cdots T_{V,k+1}\Delta_{D,k}T_{U,k-1}\cdots T_{U,0}, \tag{262}$$

and the total defect-identification error is

$$\sum_{k=0}^{n-1} T_{V,n-1}\cdots T_{V,k+1}\Delta_{r,k}. \tag{263}$$

Apply both sums to u_0, use $D_0u_0 + r_0 = 0$, and take norms. □

Corollary 8.16 (Nonexpansive path bound)**.** *If every* $T_{U,k}$ *and* $T_{V,k}$ *is nonexpansive, then*

$$\|D_nu_n + r_n\| \le \|u_0\| \sum_{k=0}^{n-1} \varepsilon_{D,k} + \sum_{k=0}^{n-1} \varepsilon_{r,k}. \tag{264}$$

Definition 8.17 (Transport route cost)**.** For a fixed source correction u_0, define the certified cost of a nonexpansive path p by

$$C_{\mathrm{tr}}(p; u_0) := \|u_0\| \sum_{e\in p} \varepsilon_{D,e} + \sum_{e\in p} \varepsilon_{r,e} + \sum_{e\in p} \varepsilon_{\mathrm{auth},e} + \sum_{e\in p} \varepsilon_{\mathrm{sem},e}. \tag{265}$$

Proposition 8.18 (Shortest certified route)**.** *When all edge transports are nonexpansive and the declared error contributions are additive, a shortest path in the edge-weighted transport graph minimizes the stated worst-case transfer penalty and therefore maximizes the corresponding certified lower bound on target gain.*

Proof. Under the hypotheses, the path penalty is the sum of nonnegative edge costs. Minimizing this sum is exactly the shortest-path problem. □

Remark 8.19. If some transports amplify norms, route quality depends on the prefix and suffix products in Theorem 8.15; a scalar edge weight is no longer sufficient. The routing state must then retain the current transported correction norm and downstream amplification profile.

8.4 Repair holonomy and path dependence

Let G be a graph of correction data with invertible exact transports. For a loop γ based at system A, define correction and defect holonomies

$$H_\gamma^U : U_A \to U_A, \qquad H_\gamma^V : V_A \to V_A. \tag{266}$$

Exact commutation along the loop gives

$$D_A H_\gamma^U = H_\gamma^V D_A. \tag{267}$$

Definition 8.20 (Path-flat repair and defect cores)**.** The path-flat repair and defect cores at A are

$$U_A^\flat := \bigcap_\gamma \ker(H_\gamma^U - I), \qquad V_A^\flat := \bigcap_\gamma \ker(H_\gamma^V - I). \tag{268}$$

Proposition 8.21 (Flat repairs remain flat defects)**.** *If $u \in U_A^\flat$, then*

$$D_A u \in V_A^\flat. \tag{269}$$

Proof. For every loop γ,

$$H_\gamma^V D_A u = D_A H_\gamma^U u = D_A u. \tag{270}$$

□

Theorem 8.22 (Path independence on the flat repair core)**.** *Let p and q be two exact invertible transport paths from A to B. If $u \in U_A$ is fixed by the loop holonomy associated with $q^{-1}p$, then*

$$T_U^p u = T_U^q u. \tag{271}$$

In particular, every $u \in U_A^\flat$ has path-independent transport to B.

Proof. The path identity is

$$T_U^p = T_U^q H_{q^{-1}p}^U. \tag{272}$$

Apply both sides to u and use $H_{q^{-1}p}^U u = u$. □

Definition 8.23 (Repair holonomy residual)**.** For a correction u and loop γ, define

$$\eta_\gamma^U(u) := \left\| (H_\gamma^U - I)u \right\|. \tag{273}$$

For a defect r, define

$$\eta_\gamma^V(r) := \left\| (H_\gamma^V - I)r \right\|. \tag{274}$$

Proposition 8.24 (Path-discrepancy identity)**.** *Under the hypotheses of Theorem 8.22,*

$$\|T_U^p u - T_U^q u\| \le \|T_U^q\| \, \eta^U_{q^{-1}p}(u). \tag{275}$$

Thus loop holonomy directly bounds disagreement between two transport routes.

Proof. Use

$$T_U^p u - T_U^q u = T_U^q (H^U_{q^{-1}p} - I)u \tag{276}$$

and take norms. □

8.5 Approximate transported residual

Theorem 8.25 (Approximate transported residual)**.** *Let U_A, U_B, V_A, V_B be normed spaces. Suppose u_A solves*

$$D_A u_A = -r_A. \tag{277}$$

Assume

$$\|D_B T_U - T_V D_A\| \le \varepsilon_D \tag{278}$$

and

$$\|r_B - T_V r_A\| \le \varepsilon_r. \tag{279}$$

Then $u_B = T_U u_A$ satisfies

$$\|D_B u_B + r_B\| \le \varepsilon_D \, \|u_A\| + \varepsilon_r. \tag{280}$$

Proof. Add and subtract $T_V D_A u_A$ and $T_V r_A$:

$$D_B T_U u_A + r_B = (D_B T_U - T_V D_A)u_A + T_V(D_A u_A + r_A) + (r_B - T_V r_A). \tag{281}$$

The middle term vanishes. Taking norms yields the claim. □

Definition 8.26 (Scale-normalized transport defect)**.** For a declared stabilizer $\tau > 0$, define

$$\varepsilon_D^{\mathrm{rel}} := \frac{\|D_B T_U - T_V D_A\|}{\|D_B\| \, \|T_U\| + \|T_V\| \, \|D_A\| + \tau}. \tag{282}$$

The normalized defect is not a substitute for the absolute residual in Theorem 8.25. It is a diagnostic for comparing transport squares at different numerical scales. Both quantities should be retained in a repair receipt.

8.6 Deterministic gain preservation

Let $\ell_A : V_A \to \mathbb{R}$ and $\ell_B : V_B \to \mathbb{R}$ be post-defect losses. Define source and target gains

$$g_A := \ell_A(r_A) - \ell_A(r_A + D_A u_A), \tag{283}$$

$$g_B := \ell_B(r_B) - \ell_B(r_B + D_B T_U u_A). \tag{284}$$

Assumption 8.27 (Evaluator alignment)**.** There is $\varepsilon_J \ge 0$ such that

$$|\ell_B(T_V v) - \ell_A(v)| \le \varepsilon_J \tag{285}$$

for

$$v \in \{r_A, r_A + D_A u_A\}. \tag{286}$$

Theorem 8.28 (Approximate gain-preserving repair transport)**.** *Assume ℓ_B is L-Lipschitz. Suppose the hypotheses of Theorem 8.25 hold. Let additional target-side perturbations from packet leakage and semantic holonomy have total defect norm at most*

$$\varepsilon_{\mathrm{leak}} + \varepsilon_{\mathrm{hol}}. \tag{287}$$

Then

$$\boxed{g_B \geq g_A - 2\varepsilon_J - L\left(2\varepsilon_r + \varepsilon_D \|u_A\| + \varepsilon_{\mathrm{leak}} + \varepsilon_{\mathrm{hol}}\right).} \tag{288}$$

In particular, the transported repair is guaranteed beneficial whenever the source gain exceeds the total right-hand error margin.

Proof. Evaluator alignment and Lipschitz continuity imply

$$\ell_B(r_B) \geq \ell_B(T_V r_A) - L\varepsilon_r \geq \ell_A(r_A) - \varepsilon_J - L\varepsilon_r. \tag{289}$$

The target post-repair defect differs from

$$T_V(r_A + D_A u_A) \tag{290}$$

by at most

$$\varepsilon_r + \varepsilon_D \|u_A\| + \varepsilon_{\mathrm{leak}} + \varepsilon_{\mathrm{hol}} \tag{291}$$

using Theorem 8.25 and the declared additional perturbation bounds. Hence

$$\begin{aligned}
\ell_B(r_B + D_B T_U u_A) &\leq \ell_B(T_V(r_A + D_A u_A)) && (292)\\
&\quad + L\left(\varepsilon_r + \varepsilon_D \|u_A\| + \varepsilon_{\mathrm{leak}} + \varepsilon_{\mathrm{hol}}\right) && (293)\\
&\leq \ell_A(r_A + D_A u_A) + \varepsilon_J && (294)\\
&\quad + L\left(\varepsilon_r + \varepsilon_D \|u_A\| + \varepsilon_{\mathrm{leak}} + \varepsilon_{\mathrm{hol}}\right). && (295)
\end{aligned}$$

Subtracting the post-repair upper bound from the baseline lower bound gives the result. □

Corollary 8.29 (Pathwise gain certificate)**.** *Along a nonexpansive transport path, replace*

$$\varepsilon_D \|u_A\| + \varepsilon_r \tag{296}$$

by the pathwise residual bound in Theorem 8.16. If the source gain exceeds twice the evaluator-alignment error plus the resulting Lipschitz, leakage, and holonomy penalties, then the endpoint repair retains positive certified gain.

Remark 8.30. The theorem makes repair transfer falsifiable. A failed transfer can be attributed to defect mismatch, correction noncommutation, evaluator mismatch, packet leakage, authorization loss, or semantic holonomy rather than being described only as distribution shift.

8.7 Distributional and probabilistic transfer certificates

The deterministic theorem compares declared defects. A deployed repair is usually evaluated across a distribution of contexts. Let P_A and P_B be source and target context distributions, let $S : X_A \to X_B$ be a declared context transport, and let q_A, q_B be per-context repair gains.

Theorem 8.31 (Coupled distributional gain transfer)**.** *Suppose there is a coupling π of P_A and P_B such that*

$$\mathbb{E}_{(x,y)\sim\pi} d_B(y, Sx) \le \delta. \tag{297}$$

Assume q_B is K-Lipschitz and

$$q_B(Sx) \ge q_A(x) - \eta \tag{298}$$

for P_A-almost every x. Then

$$\mathbb{E}_{y\sim P_B} q_B(y) \ge \mathbb{E}_{x\sim P_A} q_A(x) - \eta - K\delta. \tag{299}$$

Proof. Under the coupling,

$$q_B(y) \ge q_B(Sx) - K d_B(y, Sx) \ge q_A(x) - \eta - K d_B(y, Sx). \tag{300}$$

Take expectations. □

The term η can itself be instantiated by the structural penalty from Theorem 8.28. The theorem then separates representational transport error from context-distribution mismatch.

Definition 8.32 (High-probability transport certificate)**.** A random tuple

$$(\underline{g}_A, \overline{\varepsilon}_J, \overline{\varepsilon}_r, \overline{\varepsilon}_D, \overline{\varepsilon}_{\text{leak}}, \overline{\varepsilon}_{\text{hol}}, \overline{L}) \tag{301}$$

is an $(1-\alpha)$ transport certificate when, jointly with probability at least $1-\alpha$, it lower-bounds g_A and upper-bounds every corresponding error or Lipschitz quantity.

Theorem 8.33 (Probabilistic gain preservation by simultaneous bounds)**.** *Given an $(1-\alpha)$ transport certificate, define*

$$\begin{aligned} \underline{g}_B := \underline{g}_A &- 2\overline{\varepsilon}_J \\ &- \overline{L}\left(2\overline{\varepsilon}_r + \overline{\varepsilon}_D \|u_A\| + \overline{\varepsilon}_{\text{leak}} + \overline{\varepsilon}_{\text{hol}}\right). \end{aligned} \tag{302}$$

Then

$$\Pr(g_B \ge \underline{g}_B) \ge 1 - \alpha. \tag{303}$$

In particular, $\underline{g}_B > 0$ is a valid one-sided certificate of beneficial target transfer.

Proof. On the simultaneous confidence event, substitute every bound into Theorem 8.28. The event has probability at least $1-\alpha$. □

Theorem 8.34 (Paired held-out gain certificate)**.** *Let $Z_1, \dots, Z_n$ be independent and identically distributed paired target-context gains for a precommitted transported repair, with*

$$-G \le Z_i \le G. \tag{304}$$

Let $\overline{Z}_n = n^{-1}\sum_i Z_i$. *Then, with probability at least* $1-\alpha$,

$$\mathbb{E}Z_i \geq \overline{Z}_n - G\sqrt{\frac{2\log(1/\alpha)}{n}}. \tag{305}$$

For a precommitted family of m *repairs, replacing* α *by* α/m *gives a simultaneous lower bound for all* m *candidates.*

Proof. Hoeffding's inequality for a variable with range width $2G$ gives

$$\Pr\left(\overline{Z}_n - \mathbb{E}Z_i \geq t\right) \leq \exp\left(-\frac{nt^2}{2G^2}\right). \tag{306}$$

Set the right side to α. The family statement follows by the union bound. □

Warning 8.35 (Adaptive reuse of validation evidence)**.** The bound in Theorem 8.34 assumes the repair or finite candidate family is fixed before the held-out outcomes are revealed. Searching the same validation sample until a positive lower bound appears invalidates the stated guarantee unless the adaptive search is covered by a separate sequential or reusable-holdout argument.

Block 3 transport result

The transport layer now distinguishes exact equivalence, information-losing abstraction, authorization preservation, invariant preservation, pathwise error accumulation, route choice, repair holonomy, deterministic gain margin, distributional shift, and finite-sample certification. A transported repair should be accepted only when these receipts jointly support the claimed preservation profile.

9 Path-Flat Evaluator Evolution

An evaluator is not only a scalar score. It is a structured map from raw evidence through an internal representation to judgments, uncertainty estimates, authorization decisions, and comparisons among descendants. Evaluator evolution is legitimate only when new distinctions are added without silently changing the meaning of protected historical evidence.

9.1 Evaluator objects and morphisms

Definition 9.1 (Evaluator object)**.** An evaluator object at version v is a tuple

$$\mathsf{E}_v = (R_v, E_v, \iota_v, J_v, Q_v, \mathfrak{h}_v), \tag{307}$$

where:

- R_v is the protected raw-evidence space or ledger span;
- E_v is the evaluator's internal evidence-representation space;
- $\iota_v : R_v \to E_v$ encodes raw evidence;
- $J_v : E_v \to \mathbb{R}^q$ is a score or decision-readout map;

- Q_v is an uncertainty or calibration readout;
- $\mathfrak{h}_v$ is the historical replay set protected at version v.

Definition 9.2 (Evaluator morphism)**.** An evaluator morphism $e : v \to w$ consists of maps

$$S_e : R_v \to R_w, \qquad T_e : E_v \to E_w, \tag{308}$$

where S_e is the append-only raw-evidence embedding and T_e transports internal representations. Its protected-evidence defect is

$$\delta_{\mathrm{obs}}(e) := \sup_{r \in \mathfrak{h}_v} \left\| T_e \iota_v(r) - \iota_w S_e(r) \right\|, \tag{309}$$

and its historical score defect is

$$\delta_J(e) := \sup_{r \in \mathfrak{h}_v} \left\| J_w T_e \iota_v(r) - J_v \iota_v(r) \right\|. \tag{310}$$

The morphism is exact on protected history when both defects vanish.

Proposition 9.3 (Composition of evaluator defects)**.** *Let $e : v \to w$ and $f : w \to z$ be evaluator morphisms. Suppose T_f is L_T-Lipschitz on represented protected evidence. Then*

$$\delta_{\mathrm{obs}}(f \circ e) \le L_T \delta_{\mathrm{obs}}(e) + \delta_{\mathrm{obs}}(f), \tag{311}$$

where the second term is evaluated on $S_e(\mathfrak{h}_v)$. If, in addition,

$$\sup_{r \in \mathfrak{h}_v} \| J_z T_f T_e \iota_v(r) - J_w T_e \iota_v(r) \| \le \delta_J(f \mid e), \tag{312}$$

then

$$\delta_J(f \circ e) \le \delta_J(e) + \delta_J(f \mid e). \tag{313}$$

Proof. For the evidence defect, insert $T_f \iota_w S_e(r)$ and apply the triangle inequality and the Lipschitz bound on T_f. For the score defect, insert $J_w T_e \iota_v(r)$ and use the two declared suprema. □

9.2 Evaluator atlases and exact holonomy

Let $G = (N, E)$ be a connected directed graph of evaluator versions. Each directed edge $e : v \to w$ carries an invertible internal transport

$$T_e : E_v \to E_w. \tag{314}$$

Fix a base node v_0. For any loop γ based at v_0, let

$$H_\gamma : E_{v_0} \to E_{v_0} \tag{315}$$

be the composed transport around the loop.

Definition 9.4 (Path-flat evidence core)**.** The path-flat evidence core at v_0 is

$$E_{v_0}^{\flat} := \bigcap_{\gamma \in \pi_1(G,v_0)} \ker(H_\gamma - I). \tag{316}$$

Theorem 9.5 (Maximal path-flat evaluator core)**.** *The subspace $E_{v_0}^{\flat}$ is the unique greatest subspace of E_{v_0} fixed pointwise by every evaluator-loop holonomy. That is:*

1. *$H_\gamma x = x$ for every $x \in E_{v_0}^{\flat}$ and every loop γ;*
2. *if $W \subseteq E_{v_0}$ satisfies $H_\gamma w = w$ for all $w \in W$ and all loops γ, then $W \subseteq E_{v_0}^{\flat}$.*

Proof. The first statement is immediate from the intersection definition. If W is fixed pointwise by every H_γ, then $W \subseteq \ker(H_\gamma - I)$ for every loop γ, hence W lies in their intersection. □

Proposition 9.6 (Finite computation from a cycle basis)**.** *Let G be finite. Choose a spanning tree and let $\gamma_1, \dots, \gamma_m$ be the fundamental loops generated by non-tree edges. If the loop transports form a representation of the graph fundamental group, then*

$$E_{v_0}^{\flat} = \bigcap_{j=1}^{m} \ker(H_{\gamma_j} - I). \tag{317}$$

Equivalently, $E_{v_0}^{\flat}$ is the null space of the vertically stacked matrix

$$\begin{bmatrix} H_{\gamma_1} - I \\ \vdots \\ H_{\gamma_m} - I \end{bmatrix}. \tag{318}$$

Proof. Every loop class is generated by the fundamental loops and their inverses. A vector fixed by each generator is fixed by every product of generators. The stacked-null-space statement is equivalent. □

Theorem 9.7 (Basepoint covariance of the flat core)**.** *Let $p : v \to w$ be any invertible exact evaluator path in a connected atlas. Then*

$$T_p(E_v^{\flat}) = E_w^{\flat}. \tag{319}$$

Hence the dimension of the path-flat core is independent of basepoint.

Proof. Every loop γ at w corresponds to the conjugate loop $p^{-1}\gamma p$ at v, with

$$H_\gamma = T_p H_{p^{-1}\gamma p} T_p^{-1}. \tag{320}$$

If $x \in E_v^{\flat}$, then $H_\gamma T_p x = T_p H_{p^{-1}\gamma p} x = T_p x$, so $T_p x \in E_w^{\flat}$. Apply the same argument to p^{-1} for the reverse inclusion. □

Invertibility is not always realistic. A later evaluator may compress or expand its internal evidence representation.

Definition 9.8 (Closed-walk equalizer core)**.** For a possibly noninvertible evaluator graph, let $\mathsf{W}(v)$ be the monoid of directed closed walks at v. Define

$$E_v^{\mathrm{eq}} := \bigcap_{\gamma \in \mathsf{W}(v)} \ker(H_\gamma - I). \tag{321}$$

Proposition 9.9 (Maximality without invertibility)**.** *The equalizer core E_v^{eq} is the unique greatest subspace fixed pointwise by every closed-walk endomorphism. A fundamental cycle basis need not suffice when edge maps are noninvertible and the closed-walk maps do not factor through a group representation.*

Proof. The maximality argument is the same intersection-of-equalizers argument as in Theorem 9.5. The cycle-basis reduction used inverses and group generation, which are unavailable in general. □

9.3 Approximate path-flat cores

Exact fixed subspaces can be too brittle for learned evaluator transports. Let $H_1, \dots, H_m$ be a finite generating family of loop transports on a Hilbert space E, with positive weights w_j satisfying $\sum_j w_j = 1$. Define the holonomy-energy operator

$$\mathsf{A} := \sum_{j=1}^{m} w_j (H_j - I)^*(H_j - I). \tag{322}$$

This operator is positive semidefinite.

Theorem 9.10 (Spectral characterization of flatness)**.** *The exact common fixed space is*

$$E^\flat = \ker \mathsf{A}. \tag{323}$$

Let the eigenvalues of A *be*

$$0 \le \lambda_1 \le \cdots \le \lambda_d. \tag{324}$$

Among all k-dimensional subspaces $F \subseteq E$, the quantity

$$\sum_{i=1}^{k} \langle x_i, \mathsf{A} x_i \rangle \tag{325}$$

for an orthonormal basis $x_1, \dots, x_k$ of F is minimized by the span of eigenvectors associated with $\lambda_1, \dots, \lambda_k$, and the minimum is $\sum_{i=1}^{k} \lambda_i$.

Proof. For every x,

$$\langle x, \mathsf{A} x \rangle = \sum_j w_j \left\| (H_j - I)x \right\|^2. \tag{326}$$

This vanishes exactly when every $H_j x = x$, proving the kernel statement. Expand an orthonormal k-frame in an eigenbasis of A. The sum of Rayleigh quotients is a convex combination of eigenvalue sums and is minimized by assigning all available weight to the k smallest eigenvalues. □

Definition 9.11 (Thresholded approximate flat core)**.** For $\tau \geq 0$, define

$$E_\tau^\flat := \operatorname{span}\{x_i : \lambda_i \leq \tau\}, \tag{327}$$

where x_i is an eigenbasis of A.

Corollary 9.12 (Distortion bound on the approximate core)**.** *For every $x \in E_\tau^\flat$,*

$$\sum_{j=1}^{m} w_j \|(H_j - I)x\|^2 \leq \tau \|x\|^2. \tag{328}$$

In particular,

$$\|(H_j - I)x\| \leq \sqrt{\frac{\tau}{w_j}} \|x\| \tag{329}$$

for every generator j.

Proof. The Rayleigh quotient of A on the spectral subspace with eigenvalues at most τ is at most τ. Each weighted nonnegative summand is bounded by the total. □

Proposition 9.13 (Stability of the threshold count)**.** *Let $\widehat{\mathsf{A}}$ be an estimated holonomy-energy operator satisfying*

$$\left\|\widehat{\mathsf{A}} - \mathsf{A}\right\| \leq \eta. \tag{330}$$

Then each ordered eigenvalue satisfies

$$\left|\widehat{\lambda}_i - \lambda_i\right| \leq \eta. \tag{331}$$

If no eigenvalue of A lies in $[\tau - \eta, \tau + \eta]$, then $\widehat{\mathsf{A}}$ and A have the same number of eigenvalues below τ.

Proof. For every unit vector x,

$$\left|\left\langle x, \widehat{\mathsf{A}}x\right\rangle - \langle x, \mathsf{A}x\rangle\right| \leq \eta. \tag{332}$$

Apply the min–max characterization of ordered eigenvalues. The threshold-count statement follows immediately. □

9.4 Semantic holonomy and decision stability

Definition 9.14 (Semantic holonomy residual)**.** For a represented evidence state $x \in E_v$ and loop γ at v, define

$$\eta_{\mathrm{sem}}(x;\gamma) := \|(H_\gamma - I)x\|. \tag{333}$$

Theorem 9.15 (Readout stability under semantic holonomy)**.** *Let $J : E_v \to Y$ be L_J-Lipschitz. Then*

$$d_Y(J(H_\gamma x), J(x)) \leq L_J \eta_{\mathrm{sem}}(x;\gamma). \tag{334}$$

If J returns a score vector and the winning score at x has margin $m(x)$ over the runner-up, then the decision is unchanged whenever

$$2L_J \eta_{\mathrm{sem}}(x;\gamma) < m(x), \tag{335}$$

where L_J bounds each score coordinate in the same norm.

Proof. The first inequality is Lipschitz continuity. Each of the winning and runner-up scores can move by at most $L_J\eta_{\mathrm{sem}}$. Their gap therefore remains positive when the total possible contraction of the gap is less than $m(x)$. □

Theorem 9.16 (Cumulative evaluator-drift bound)**.** *Consider a chain of evaluator morphisms with readouts into a common complete normed score space,*

$$\mathsf{E}_0 \xrightarrow{e_0} \mathsf{E}_1 \xrightarrow{e_1} \cdots \xrightarrow{e_{n-1}} \mathsf{E}_n. \tag{336}$$

Suppose that for every protected represented state x_t transported from version 0,

$$\|J_{t+1}(T_t x_t) - J_t(x_t)\| \le \varepsilon_t. \tag{337}$$

Then

$$\|J_n(T_{n-1}\cdots T_0 x_0) - J_0(x_0)\| \le \sum_{t=0}^{n-1} \varepsilon_t. \tag{338}$$

If $\sum_t \varepsilon_t < \infty$, then the transported score sequence is Cauchy and converges in every complete score space.

Proof. The finite bound is the triangle inequality along the chain. Summability implies the Cauchy criterion. □

This theorem supplies a quantitative epistemic ratchet. Exact preservation corresponds to $\varepsilon_t = 0$. Summable nonzero error permits evaluator evolution while preventing unbounded cumulative reinterpretation of protected history.

9.5 Flatness is not sufficiency

The zero subspace is always path-flat. Therefore flatness alone cannot establish that the retained core contains enough information to distinguish outcomes relevant to improvement.

Let $\Delta_{\mathrm{prot}} \subseteq E \setminus \{0\}$ be a declared set of protected evidence differences: each $d = x - y$ represents two historical or possible evidence states that the evaluator must be able to distinguish.

Definition 9.17 (Core adequacy modulus)**.** For a subspace $F \subseteq E$ with orthogonal projector P_F, define

$$\mu(F; \Delta_{\mathrm{prot}}) := \inf_{d \in \Delta_{\mathrm{prot}}} \frac{\|P_F d\|}{\|d\|}. \tag{339}$$

Proposition 9.18 (Robust protected distinction)**.** *If*

$$\mu(F; \Delta_{\mathrm{prot}}) = \mu > 0, \tag{340}$$

then every protected pair remains separated after projection onto F. If the projected representations of each member of a pair are perturbed by at most η, the pair remains distinct whenever

$$2\eta < \mu \|d\|. \tag{341}$$

Proof. By definition $\|P_F d\| \geq \mu \|d\|$. Two perturbations can reduce the separation by at most 2η. □

Definition 9.19 (Adequate approximate evidence core)**.** A subspace F is (τ, μ_0)-adequate when:

1. its weighted holonomy energy is at most τ per unit norm;
2. $\mu(F; \Delta_{\text{prot}}) \geq \mu_0 > 0$.

The two conditions are independent. Low holonomy without separation is vacuous. High separation without low holonomy preserves distinctions but allows their semantics to become path dependent.

9.6 Conservative evaluator upgrades

Definition 9.20 (Core-retractable upgrade)**.** An evaluator morphism $e : v \to w$ is core-retractable when

$$T_e(E_v^\flat) \subseteq E_w^\flat \tag{342}$$

and there is a map

$$R_e : E_w^\flat \to E_v^\flat \tag{343}$$

such that

$$R_e T_e x = x \qquad \text{for every } x \in E_v^\flat. \tag{344}$$

It is core-isomorphic when T_e restricts to an isomorphism from $E_v^\flat$ onto $E_w^\flat$.

Core retractability allows the new evaluator to add dimensions while preserving a recoverable copy of the old evidence core.

Definition 9.21 (Conservative evaluator upgrade)**.** An evaluator upgrade $e : v \to w$ is conservative on protected history when:

1. $S_e : R_v \to R_w$ is injective;
2. e is core-retractable;
3. $\delta_{\text{obs}}(e)$ is below the declared evidence tolerance;
4. $\delta_J(e)$ is below the declared historical-replay tolerance;
5. every new score or distinction is typed as new rather than retroactively presented as an old observation;
6. an independent witness can restore version v and replay $\mathfrak{h}_v$.

Theorem 9.22 (Historical ranking preservation)**.** *Let $x_1, \ldots, x_N$ be protected historical candidate records with scalar old scores $J_v(x_i)$. Suppose every unequal old pair has margin at least*

$$m_{\text{hist}} := \min_{J_v(x_i) \neq J_v(x_j)} |J_v(x_i) - J_v(x_j)| > 0. \tag{345}$$

If an upgrade satisfies

$$|J_w(T_e x_i) - J_v(x_i)| \leq \varepsilon \tag{346}$$

for every i, with

$$2\varepsilon < m_{\text{hist}}, \tag{347}$$

then every strict historical ranking is preserved.

Proof. For $J_v(x_i) > J_v(x_j)$,

$$J_w(T_e x_i) - J_w(T_e x_j) \geq J_v(x_i) - J_v(x_j) - 2\varepsilon \tag{348}$$

$$\geq m_{\text{hist}} - 2\varepsilon > 0. \tag{349}$$

□

Proposition 9.23 (No silent protected-evidence erasure)**.** *Suppose S_e is injective and $T_e \iota_v = \iota_w S_e$ on protected history. If T_e is injective on the span of protected evidence differences, then distinct protected records remain distinct in the new evaluator representation.*

Proof. For distinct records r_1, r_2 with protected difference $d = \iota_v(r_1) - \iota_v(r_2) \neq 0$, injectivity gives $T_e d \neq 0$. Exact commutation identifies this with the difference of their new representations. □

Proposition 9.24 (Conservative upgrades compose)**.** *Exact conservative evaluator upgrades with core retractions compose. Approximate conservative upgrades compose with evidence and score tolerances bounded by Theorem 9.3. Therefore a chain remains conservative whenever its accumulated errors remain below the declared historical margins and core-adequacy thresholds.*

Proof. Injective raw-evidence embeddings and left inverses compose. Typed provenance and rollback receipts compose by concatenation. The approximate bounds are Theorem 9.3. □

9.7 Statistical certification of evaluator upgrades

Historical replay establishes consistency with old evidence. It does not establish that the new evaluator is more accurate on new external outcomes. That claim requires a held-out comparison.

Theorem 9.25 (Paired evaluator-risk improvement)**.** *Let $Z_1, \dots, Z_n$ be independent and identically distributed held-out externally revealed cases. Let $L_v(Z_i), L_w(Z_i) \in [0,1]$ be the losses of the old and proposed evaluators, and define*

$$\Delta_i = L_v(Z_i) - L_w(Z_i) \in [-1, 1]. \tag{350}$$

If the upgrade and evaluation rule are fixed before the held-out outcomes are revealed, then with probability at least $1 - \alpha$,

$$\mathbb{E}\Delta_i \geq \overline{\Delta}_n - \sqrt{\frac{2\log(1/\alpha)}{n}}. \tag{351}$$

Thus a positive lower bound certifies lower expected held-out risk for the proposed evaluator.

Proof. Apply Theorem 8.34 with $G = 1$. □

Corollary 9.26 (Multiple protected metrics)**.** *If m precommitted evaluator candidates or protected metrics are tested, replacing α by α/m yields simultaneous one-sided guarantees. An upgrade may require positive gain on the primary metric and lower bounds above declared negative tolerances on every protected secondary metric.*

Warning 9.27. A new evaluator must not use the held-out outcomes both to invent its scoring rule and to certify that rule. Proposal evidence, historical replay evidence, and final external validation evidence should be separated or governed by a valid adaptive-testing protocol.

9.8 Proof-carrying evaluator-upgrade protocol

Definition 9.28 (Evaluator-upgrade certificate)**.** An evaluator-upgrade certificate is a tuple

$$\Pi_{\mathrm{eval}} = (v, w, S_e, T_e, R_e, \mathfrak{h}_v, \mathsf{A}_v, \mathsf{A}_w, \tau, \mu_0, \delta_{\mathrm{obs}}, \delta_J, \mathcal{T}_{\mathrm{new}}, \mathcal{R}_{\mathrm{rollback}}, \sigma), \tag{352}$$

containing the old and new evaluator identifiers, raw and internal transports, core retraction, protected replay set, holonomy-energy operators, accepted flatness and adequacy thresholds, measured replay defects, held-out new-evidence tests, rollback record, and uncertainty statement σ.

Protocol 9.29 (Certified evaluator upgrade)**.** A proposed evaluator version w is provisionally admitted from version v only after the following steps:

1. freeze the candidate evaluator and all acceptance thresholds;
2. embed the append-only raw-evidence ledger through S_e;
3. verify protected-evidence commutation and obstruction faithfulness;
4. compute exact or approximate path-flat cores and the adequacy modulus;
5. replay historical decisions and check ranking margins;
6. evaluate new externally revealed cases with a precommitted statistical test;
7. test cycle holonomy against all currently retained evaluator branches;
8. record a witness identity and rollback checkpoint not editable by the candidate transaction;
9. commit provisionally if every required certificate passes; otherwise restore version v.

Theorem 9.30 (Soundness of a certified evaluator chain)**.** *Consider a finite chain of upgrades admitted by Protocol 9.29. Suppose each upgrade is core-retractable, has protected score error at most ε_t, perturbs each retained protected representation by at most η_t, and retains core adequacy at least $\mu_0 > 0$. Then:*

1. *the original protected core is recoverably embedded at every later version;*
2. *cumulative protected score drift is at most $\sum_t \varepsilon_t$;*
3. *every original strict decision margin larger than $2\sum_t \varepsilon_t$ is preserved;*
4. *a protected difference d remains distinguishable whenever*

$$\mu_0 \, \|d\| > 2 \sum_t \eta_t. \tag{353}$$

If the score-error series is summable along an infinite chain, protected scores converge.

Proof. Item 1 follows by composing core retractions. Item 2 is Theorem 9.16. Item 3 applies the proof of Theorem 9.22 to the cumulative bound. Item 4 is Theorem 9.18 with per-record accumulated perturbation $\sum_t \eta_t$. The final statement follows from summability. □

9.9 Examples of evaluator holonomy

Example 9.31 (Shear holonomy)**.** Let $E = \mathbb{R}^2$ and suppose evaluator evolution around one loop produces

$$H = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix}, \qquad \alpha \neq 0. \tag{354}$$

Then

$$H - I = \begin{bmatrix} 0 & \alpha \\ 0 & 0 \end{bmatrix}, \tag{355}$$

so

$$E^\flat = \ker(H - I) = \{(x, 0) : x \in \mathbb{R}\}. \tag{356}$$

The first coordinate is path-flat. The second coordinate changes the first under a loop and therefore cannot be treated as invariant evidence.

Example 9.32 (Flat but inadequate core)**.** Let protected evidence states be

$$x_+ = (1, 1), \qquad x_- = (1, -1), \tag{357}$$

and let shear holonomy leave only the first coordinate flat. Both records project to $(1, 0)$, so

$$\mu(E^\flat; \{x_+ - x_-\}) = 0. \tag{358}$$

The core is exactly flat but cannot preserve the protected distinction between the two outcomes.

Block 3 evaluator result

Evaluator evolution is represented by an atlas of raw-evidence embeddings, internal transports, readouts, and replay obligations. Exact path-flatness controls path dependence. Spectral approximate cores quantify near-flatness. The adequacy modulus prevents a trivial or information-poor core from being treated as sufficient. Conservative upgrades preserve recoverable evidence, historical margins, and externally validated risk improvements through proof-carrying receipts.

---

10 Witness-Separated Improvement Transactions

The preceding sections characterize when a repair exists, whether it glues across strata, whether it transports to a new system, and whether the evaluator retains a stable evidential meaning. None of those results by itself specifies how a candidate modification is executed without allowing the candidate to rewrite the rule that judges it. This section gives a transactional semantics for that separation.

10.1 Transaction objects and immutable surfaces

Let a complete implementation state decompose as

$$\Sigma = \Sigma^{\text{task}} \oplus \Sigma^{\text{memory}} \oplus \Sigma^{\text{repair}} \oplus \Sigma^{\text{evaluation}} \oplus \Sigma^{\text{provenance}} \oplus \Sigma^{\text{rollback}}. \tag{359}$$

A transaction selects an editable mask $E \subseteq \Sigma$ and a witness mask $W \subseteq \Sigma$ with

$$E \cap W = \varnothing. \tag{360}$$

The decomposition is logical rather than necessarily physical. Two components may share hardware while remaining isolated by capabilities, address spaces, process boundaries, cryptographic commitments, or reproducible snapshots.

Definition 10.1 (Improvement transaction)**.** A proof-carrying improvement transaction is a tuple

$$\tau = \left(P, E, W, \delta, \widehat{\Delta}, \mathcal{C}, q, a, Z, \Pi\right), \tag{361}$$

where:

1. P is a content-addressed parent checkpoint;
2. E is the editable component mask;
3. W is the frozen witness mask;
4. δ is the proposed edit;
5. $\widehat{\Delta}$ is the precommitted effect prediction;
6. $\mathcal{C}$ is the capability and resource budget;
7. q is the declared trace map from full executions to witness-visible traces;
8. a is the sealed acceptance functional;
9. Z is the revealed trace;
10. Π is the final receipt containing the decision, evidence links, and rollback or commit record.

Definition 10.2 (Well-formed transaction)**.** A transaction is well formed when:

1. δ acts only on E;
2. a, q, the evidence ledger head, and the rollback checkpoint are determined before candidate execution;
3. candidate capabilities are a subset of $\mathcal{C}$;
4. all externally persistent effects are buffered until commit or are separately declared irreversible effects;
5. the candidate cannot append to the protected evidence namespace except through q;
6. the final receipt binds the parent digest, child digest, edit digest, trace digest, evaluator version, witness version, and decision.

The transaction follows six phases:

$$\text{prepare} \longrightarrow \text{seal} \longrightarrow \text{execute} \longrightarrow \text{reveal} \longrightarrow \text{decide} \longrightarrow \text{finalize}. \tag{362}$$

Only the final phase may expose buffered effects or replace the active checkpoint.

Definition 10.3 (Witness separation)**.** A well-formed transaction is witness-separated when:

1. the candidate edit acts only on E;
2. the acceptance functional and its executable dependencies lie in W;
3. E cannot modify W, the sealed trace schema, the evidence ledger head, or the rollback checkpoint during the transaction;
4. candidate-to-witness communication factors through the declared trace map q.

Theorem 10.4 (Atomic self-approval noninterference)**.** *In a witness-separated transaction, changing the editable implementation while holding the declared trace Z fixed cannot change the acceptance decision.*

Proof. The decision has the form $a(W, Z)$. Both a and W are fixed outside the editable mask. Therefore any two candidate executions with the same declared trace induce the same decision. □

The theorem is intentionally narrow. It blocks direct atomic self-approval. It does not show that q exposes every acceptance-relevant fact.

Theorem 10.5 (Trace-factorization criterion)**.** *Let Y be the set of full execution records, let $q : Y \to Z$ be the declared trace map, and let $b : Y \to \{0, 1\}$ be the decision that an ideal full-information witness would make. The following are equivalent:*

1. *b is constant on every fiber $q^{-1}(z)$;*
2. *there exists a unique function $a : q(Y) \to \{0, 1\}$ such that*

$$b = a \circ q. \tag{363}$$

Thus the trace schema is complete for the declared decision exactly when the full-information decision factors through the trace.

Proof. If $b = a \circ q$, equal traces imply equal decisions. Conversely, if b is constant on fibers, define $a(z) = b(y)$ for any y with $q(y) = z$. Fiber constancy makes this well defined and uniqueness is immediate on $q(Y)$. □

Warning 10.6 (Trace-schema attack surface)**.** Witness separation is ineffective when distinct full executions with materially different external consequences map to the same declared trace. The factorization criterion should therefore be tested by adversarially searching for pairs y_1, y_2 satisfying

$$q(y_1) = q(y_2), \qquad b(y_1) \neq b(y_2). \tag{364}$$

Such a pair is a constructive certificate of trace incompleteness.

10.2 Rollback, atomicity, and serial composition

Let $\widetilde{F}(P, \delta)$ denote execution in an isolated child environment. Write

$$\widetilde{F}(P, \delta) = (C, B), \tag{365}$$

where C is the candidate child state and B is a buffer of externally persistent effects.

Theorem 10.7 (Atomic rollback)**.** *Assume:*

1. *the parent checkpoint P is immutable during execution;*
2. *every persistent candidate effect is either contained in C or buffered in B;*
3. *B is unpublished until the witness commits;*
4. *rejection destroys C and B while leaving the append-only receipt ledger outside the operational parent state.*

Then a rejected transaction restores the operational system exactly to P. The only persistent change is the append-only rejection receipt.

Proof. By assumptions 1 and 2, execution does not alter P and has no persistent effect outside (C, B). By assumption 3, the buffer is not externally visible before commit. Rejection deletes (C, B), so the operational state is P. Assumption 4 permits the audit ledger to record the failed transaction without changing the restored operational state. □

Definition 10.8 (Irreversible-effect budget)**.** When perfect buffering is impossible, let $I(\tau)$ be the set of irreversible external effects produced before decision. A transaction is η-reversible relative to a loss functional L_I when

$$L_I(I(\tau)) \leq \eta. \tag{366}$$

The quantity η must be part of the precommitment and acceptance rule.

Proposition 10.9 (Serializability of independent repairs)**.** *Let τ_1 and τ_2 edit disjoint component masks. Suppose their sandbox state transformations commute, their capability effects are disjoint, and neither transaction changes the sealed witness inputs of the other. Then committing both transactions in either order yields the same operational state and the same pair of transaction receipts up to receipt ordering.*

Proof. Commutation of state transformations gives equality of the two final operational states. Disjoint capability effects and unchanged witness inputs imply that each trace and decision is invariant under the order. The receipts therefore contain the same transaction-local facts, with only their ledger positions exchanged. □

Remark 10.10. The proposition supplies a sufficient condition for parallel repair. When edits overlap, the repair commutator from Section 8 or an explicit joint sandbox run is required.

10.3 Commit-reveal discipline and adaptive validity

Let $(\mathcal{F}_t)_{t\geq 0}$ be the filtration generated by all committed lineage history through transaction t. A proposal is precommitted when its edit, claimed effect, target metrics, stopping rule, and evaluation namespace are $\mathcal{F}_{t-1}$-measurable. The new evaluation trace is revealed only afterward.

Theorem 10.11 (Adaptive alpha-spending guarantee)**.** *Let F_t be the event that transaction t accepts a child whose true declared gain is nonpositive. Suppose*

$$\Pr(F_t \mid \mathcal{F}_{t-1}) \leq \delta_t \qquad \textit{almost surely} \tag{367}$$

for a deterministic sequence (δ_t). Then for every T,

$$\Pr\left(\bigcup_{t=1}^{T} F_t\right) \le \sum_{t=1}^{T} \delta_t. \tag{368}$$

If $\sum_t \delta_t < \infty$, only finitely many false acceptances occur almost surely.

Proof. Taking expectations gives $\Pr(F_t) \le \delta_t$. The finite-time claim follows from the union bound. The almost-sure statement follows from the first Borel–Cantelli lemma. □

Definition 10.12 (Evidence namespace)**.** An evidence namespace is a pair $(\mathcal{E}, \mathsf{use})$ consisting of protected records and a use ledger. Each evaluation query records which data, metrics, stopping rules, and candidate families were exposed. Reusing evidence is permitted only under a validation rule whose guarantee remains valid conditional on the recorded use history.

Warning 10.13 (Nominal confidence is not sequential confidence)**.** Repeatedly selecting the best candidate on one nominally held-out set generally violates the conditional premise of Theorem 10.11. The transaction receipt must identify whether the bound came from fresh evidence, an anytime-valid procedure, a valid conditional test, or a new independent witness evaluation.

10.4 Rotating witnesses

A permanently immutable witness limits system evolution. An unrestricted simultaneous rewrite destroys validation independence. A rotating witness schedule permits eventual replacement of every component while preserving transaction-local separation.

Let the system contain n witness-capable components. At round t, let E_t be the edited set and W_t the witness set.

Definition 10.14 (Rotating witness schedule)**.** A sequence $(E_t, W_t)_{t\ge 0}$ is a rotating witness schedule when:

1. $E_t \cap W_t = \varnothing$ for every t;
2. W_t is immutable during transaction t;
3. every component belongs to E_t for at least one t;
4. every witness replacement is validated by a component outside the same edit set;
5. protected evidence is transported through a conservative evaluator morphism at every rotation;
6. every committed round retains a rollback path to a state certified by the preceding witness.

Theorem 10.15 (Minimum rounds under a witness quorum)**.** *Suppose every transaction requires at least q frozen witness components, where $1 \le q < n$. Then at most $n - q$ components can be edited in one round. Editing every component at least once requires at least*

$$\left\lceil \frac{n}{n-q} \right\rceil \tag{369}$$

rounds. This bound is attained by partitioning the components into edit blocks of size at most $n - q$ and taking the complement of each block as its witness set.

Proof. The disjointness condition implies $|E_t| \leq n - q$, which gives the counting lower bound. A partition into blocks of that size edits every component, and each complement has at least q members. □

Definition 10.16 (Validation dependency graph)**.** For one planned replacement epoch, let $G = (V, A)$ have one vertex per component and an edge $i \to j$ when the old version of component i must validate the replacement of component j before i itself is replaced.

Theorem 10.17 (Acyclic staged replacement)**.** *If the validation dependency graph is acyclic, all components can be replaced in a finite sequence that respects every dependency. If the graph contains a directed cycle and no external witness validates any member of that cycle, then no sequential schedule can replace every member while keeping each required validator old until its target is committed.*

Proof. For an acyclic graph, update components in reverse topological order. If $i \to j$, then j is replaced before i, so i remains old while validating j. For a directed cycle, whichever member is replaced last requires validation by its predecessor on the cycle. That predecessor has already been replaced, contradicting the requirement that it remain old. An external witness can break the cycle by supplying a dependency outside it. □

Proposition 10.18 (Quorum-intersection condition)**.** *Let m witness replicas be available, and suppose at most f may be compromised. If acceptance requires a quorum of size q satisfying*

$$2q > m + f, \tag{370}$$

then any two accepting quorums intersect in more than f replicas. Hence their intersection contains at least one uncompromised replica.

Proof. For quorums Q_1, Q_2,

$$|Q_1 \cap Q_2| \geq 2q - m > f. \tag{371}$$

At most f replicas are compromised, so at least one member of the intersection is uncompromised. □

Warning 10.19. Quorum intersection limits conflicting approvals caused by a bounded number of compromised replicas. It does not prove that the uncompromised witness specification is correct, complete, or robust to shared evaluator error.

Transactional result

Witness separation is a transaction-local noncircularity condition, not a permanent immutable layer. Trace factorization identifies when the witness sees enough information for its declared decision. Sandboxing and buffered effects give exact rollback under explicit assumptions. Rotating schedules permit eventual replacement of every component, while dependency graphs and quorum bounds expose when validation becomes circular or insufficiently independent.

11 Recursive Improvement Closure

A fixed point of invariant preservation is not yet a theory of improvement. This section separates four notions: universal closure, controlled robust closure, productive recurrence, and stochastic confidence closure.

11.1 Universal finite-state closure

Let S be a finite set of complete lineage states. Write

$$s \xrightarrow{\tau_{\mathrm{imp}}} s' \tag{372}$$

when one authorized improvement transaction can take s to s'. Define

$$\mathrm{Post}_{\mathrm{imp}}(s) = \left\{ s' \in S : s \xrightarrow{\tau_{\mathrm{imp}}} s' \right\}. \tag{373}$$

Let

$$\begin{aligned} C_{\mathrm{RI}}(s) = {} & \mathrm{Grounded}(s) \wedge \mathrm{ProvValid}(s) \wedge \mathrm{WorkspaceBound}(s) \\ & \wedge \mathrm{RepairTyped}(s) \wedge \mathrm{Witnessed}(s) \wedge \mathrm{EvidenceFlat}(s) \\ & \wedge \mathrm{Rollbackable}(s) \wedge \mathrm{MetaCapable}(s). \end{aligned} \tag{374}$$

For $X \subseteq S$, define

$$\Psi_\forall(X) = \{ s \in S : C_{\mathrm{RI}}(s) \text{ and } \mathrm{Post}_{\mathrm{imp}}(s) \subseteq X \}. \tag{375}$$

Theorem 11.1 (Existence of universal recursive improvement closure)**.** *The operator $\Psi_\forall : 2^S \to 2^S$ is monotone and has a greatest fixed point*

$$\mathcal{I}^*_\forall = \nu X.\Psi_\forall(X). \tag{376}$$

A state lies in $\mathcal{I}^*_\forall$ exactly when it satisfies the static invariants and every authorized successor also lies in $\mathcal{I}^*_\forall$.

Proof. If $X \subseteq Y$, then $\mathrm{Post}_{\mathrm{imp}}(s) \subseteq X$ implies $\mathrm{Post}_{\mathrm{imp}}(s) \subseteq Y$. Thus $\Psi_\forall$ is monotone. Knaster–Tarski gives the greatest fixed point. □

Corollary 11.2 (Finite descending computation)**.** *Set $X_0 = S$ and $X_{k+1} = \Psi_\forall(X_k)$. The sequence is descending and stabilizes at $\mathcal{I}^*_\forall$ after at most $|S|$ strict reductions.*

The universal closure is deliberately strong. A state is excluded when even one authorized transition is unsafe, including a transition the controller would never select.

11.2 Controlled robust closure

Let $\mathsf{Act}(s)$ be the set of authorized transaction choices at state s. After action $u \in \mathsf{Act}(s)$, let $\mathsf{Dist}(s,u)$ be the set of admissible disturbances, evaluation outcomes, or unresolved environmental responses. Let

$$T(s,u,d) \in S \tag{377}$$

be the resulting state.

Definition 11.3 (Controllable predecessor)**.** For $X \subseteq S$, define

$$\mathrm{CPre}(X) = \{ s \in S : \exists u \in \mathsf{Act}(s)\ \forall d \in \mathsf{Dist}(s,u),\ T(s,u,d) \in X \}. \tag{378}$$

Define

$$\Psi_{\mathrm{ctrl}}(X) = \{ s \in S : C_{\mathrm{RI}}(s) \} \cap \mathrm{CPre}(X). \tag{379}$$

Theorem 11.4 (Robust controlled improvement kernel). *The operator Ψ_{ctrl} is monotone and has a greatest fixed point*

$$\mathcal{I}^*_{\mathrm{ctrl}} = \nu X.\Psi_{\mathrm{ctrl}}(X). \tag{380}$$

For every state in $\mathcal{I}^*_{\mathrm{ctrl}}$, there exists a stationary authorized policy that keeps every admissible successor inside $\mathcal{I}^*_{\mathrm{ctrl}}$.

Proof. Monotonicity follows from monotonicity of CPre. Knaster–Tarski gives the greatest fixed point. For each s in the fixed point, choose one action witnessing membership in $\mathrm{CPre}(\mathcal{I}^*_{\mathrm{ctrl}})$. Finiteness permits these choices to define a stationary policy, and every admissible disturbance remains in the fixed point by construction. □

Proposition 11.5 (Closure monotonicity). *Under a fixed invariant predicate:*

1. *adding authorized actions can only enlarge $\mathcal{I}^*_{\mathrm{ctrl}}$;*
2. *adding admissible disturbances can only shrink $\mathcal{I}^*_{\mathrm{ctrl}}$;*
3. *strengthening the static invariant predicate can only shrink both universal and controlled closures;*
4. *whenever every action set is nonempty, $\mathcal{I}^*_{\forall} \subseteq \mathcal{I}^*_{\mathrm{ctrl}}$.*

Proof. Each claim follows from the corresponding pointwise inclusion of predecessor operators, followed by monotonicity of greatest fixed points. For item 4, universal safety of every successor implies the existence of at least one robustly safe action. □

11.3 Safe recurrence is not productive recurrence

A no-op state may lie in the controlled closure forever. Recursive improvement requires a liveness condition in addition to invariant preservation.

Let $G_\gamma \subseteq S$ be the set of states whose most recent committed transaction produced a validated metaproductivity gain of at least γ under the matched-budget definition in Section 13. Define the safe predecessor

$$\mathrm{SPre}(X) = \{s : C_{\mathrm{RI}}(s)\} \cap \mathrm{CPre}(X). \tag{381}$$

Definition 11.6 (Productive recursive closure). For a finite perfect-information transaction game, define

$$\mathcal{I}^{\omega}_{\gamma} = \nu Z.\, \mu Y.\, [(G_\gamma \cap \mathrm{SPre}(Z)) \cup \mathrm{SPre}(Y)]\,. \tag{382}$$

Theorem 11.7 (Buechi characterization of repeated improvement). *A state lies in $\mathcal{I}^{\omega}_{\gamma}$ exactly when the controller has a strategy that preserves C_{RI} and forces infinitely many visits to G_γ against every admissible disturbance.*

Proof. For fixed Z, the least fixed point in Y is the safe controllable attractor of $G_\gamma \cap \mathrm{SPre}(Z)$. The outer greatest fixed point retains exactly those states from which the controller can repeatedly reach a gain state and then return to the retained region. This is the standard nested fixed-point construction for a Buechi objective in a finite game. □

Corollary 11.8 (Productivity is stricter than safety).

$$\mathcal{I}^{\omega}_{\gamma} \subseteq \mathcal{I}^*_{\mathrm{ctrl}}. \tag{383}$$

The inclusion may be strict.

Proof. Every state in the Buechi set remains inside the static invariant set under its winning strategy, hence lies in the robust safety kernel. A safe absorbing no-op state outside G_γ witnesses possible strictness. □

Proposition 11.9 (Finite ranking certificate for one gain cycle)**.** *Let $X \subseteq \mathcal{I}^*_{\mathrm{ctrl}}$ and let $h : X \to \{0, 1, \dots, H\}$ satisfy $h^{-1}(0) \subseteq G_\gamma$. Suppose that at every state with $h(s) > 0$, the controller has an authorized action whose every admissible successor s' lies in X and satisfies*

$$h(s') < h(s). \tag{384}$$

Then the controller can force a visit to G_γ within at most H transactions while remaining in X.

Proof. The nonnegative integer rank decreases strictly at every step, so it reaches zero in at most H steps. Every successor remains in X by assumption. □

Remark 11.10 (What can recur)**.** If task utility takes values in a finite set and every gain visit strictly increases it, infinitely many gain visits are impossible. Productive recurrence must eventually concern improved repair competence, grammar expansion, evaluator refinement, resource efficiency, or newly introduced task families rather than strict ascent of one fixed finite score.

11.4 Chance-constrained closure

For stochastic transitions, define

$$\mathrm{CPre}_\delta(X) = \{s : \exists u \in \mathsf{Act}(s),\ \Pr(T(s, u, D) \in X \mid s, u) \geq 1 - \delta\}. \tag{385}$$

A one-step chance constraint does not by itself provide an infinite-horizon guarantee.

Proposition 11.11 (Risk-budget composition)**.** *Suppose a policy at transaction t has conditional probability at most δ_t of leaving the declared closure, given all prior history. Then the probability of remaining in the closure through transaction T is at least*

$$1 - \sum_{t=1}^{T} \delta_t. \tag{386}$$

If $\sum_t \delta_t \leq \delta$, the probability of ever leaving is at most δ.

Proof. Apply the union bound to the conditional failure events and then take expectations, as in Theorem 10.11. □

11.5 Closure certificates and counterexamples

A finite closure computation should emit more than a set of accepted states. For each retained controlled state, the certificate records a witness action. For each removed state, it records either a violated static predicate or a disturbance that leaves the previous iterate. The descending fixed-point computation therefore yields:

1. a memoryless preservation policy on $\mathcal{I}^*_{\mathrm{ctrl}}$;

2. a finite counterexample tree for excluded states;
3. a nested-attractor strategy for $\mathcal{I}_\gamma^\omega$;
4. a progress rank for each finite attractor layer.

Closure result

Recursive improvement closure has two logically separate parts. The greatest fixed point gives a robust kernel of states from which declared invariants can be preserved. The nested Buechi fixed point gives the subset from which validated metaproductive events can be forced repeatedly. A system may satisfy the first while failing the second, so safe persistence must not be reported as recursive improvement.

12 Archive Dynamics and Calibrated Selection

The lineage archive is not only a backup store. It determines which stepping stones remain available for later repair composition, transport, and grammar expansion.

12.1 Branching archives and multiobjective frontiers

Let $\mathcal{A}_t$ be a finite directed acyclic graph of validated checkpoints and rejected-but-informative candidates through time t. Each node A has a descriptor

$$z(A) = (J(A), M(A), S(A), N(A), 1 - C(A)) \in [0,1]^d, \tag{387}$$

where the coordinates may encode task utility, foreign-agent metaproductivity, safety or invariant margin, novelty, and normalized resource efficiency.

For maximization coordinates, write $A \succeq B$ when

$$z_i(A) \geq z_i(B) \qquad \text{for every } i. \tag{388}$$

The Pareto frontier is the set of undominated nodes.

Proposition 12.1 (Archive-envelope monotonicity)**.** *Suppose archive updates never delete the last representative of a retained node. For every nonnegative weight vector w, define*

$$H_w(\mathcal{A}_t) = \max_{A \in \mathcal{A}_t} w^\top z(A). \tag{389}$$

Then $H_w(\mathcal{A}_t)$ is nondecreasing in t.

Proof. Archive inclusion gives a maximum over a weakly larger set. □

Remark 12.2. The set of Pareto-optimal nodes can change even when every scalar upper envelope is nondecreasing. A former frontier node may become dominated without becoming useless as a stepping stone.

12.2 Finite approximate frontier compression

Unbounded retention is often infeasible. The next theorem gives a simple worst-case compression guarantee.

Theorem 12.3 (Finite ε-Pareto cover)**.** *Let $Z \subseteq [0,1]^d$ be finite and let $0 < \varepsilon \leq 1$. There exists a subset $Q \subseteq Z$ with*

$$|Q| \leq \left\lceil \frac{1}{\varepsilon} \right\rceil^d \tag{390}$$

such that for every $z \in Z$, some $q \in Q$ satisfies

$$q_i \geq z_i - \varepsilon \qquad \text{for every } i. \tag{391}$$

Proof. Partition $[0,1]^d$ into axis-aligned cells of side length at most ε. Retain one point from each nonempty cell. Points in the same cell differ by at most ε in every coordinate, so the retained point satisfies the coordinatewise approximation. The number of cells gives the bound. □

Corollary 12.4 (Approximate scalarization preservation)**.** *For every $w \geq 0$ with $\|w\|_1 = 1$,*

$$\max_{q \in Q} w^{\mathsf{T}} q \geq \max_{z \in Z} w^{\mathsf{T}} z - \varepsilon. \tag{392}$$

Proof. Choose a maximizer z and its covering point q. Then $w^{\mathsf{T}} q \geq w^{\mathsf{T}} z - \varepsilon \|w\|_1$. □

12.3 Stepping stones and continuation signatures

Proposition 12.5 (Performance-only pruning can destroy the optimum)**.** *There exists a finite lineage tree in which retaining only the highest-current-utility child at each generation prevents discovery of the globally highest-utility descendant.*

Proof. Let a root have children x and y with $J(x) > J(y)$. Let x be terminal. Let y have a child z with $J(z) > J(x)$. Greedy pruning deletes y, so z becomes unreachable. □

The point is not merely that exploration helps. A lower-scoring node may preserve a repair operator, representation, or obstruction certificate unavailable elsewhere.

Definition 12.6 (Horizon-h continuation equivalence)**.** Two archive nodes A and B are horizon-h continuation equivalent, written

$$A \sim_h B, \tag{393}$$

when, under the same repair grammar and matched budgets, their sets of reachable descriptor vectors through at most h further transactions are identical.

Theorem 12.7 (Exact archive quotient)**.** *Retaining one representative from each $\sim_h$ equivalence class preserves the complete horizon-h reachable descriptor set.*

Proof. Every discarded node has a retained equivalent representative with the same horizon-h continuation set. Taking the union over retained classes therefore gives the original reachable set. □

Remark 12.8. Exact continuation equivalence is generally too expensive to compute. Practical signatures based on repair schemas, obstruction classes, latent packets, or evaluator branches are approximations. Their validity should be tested by descendant-distribution comparisons rather than assumed.

12.4 Calibrated acceptance and archive monotonicity

Let $J^*(\mathcal{A}_t)$ be the highest true protected utility among retained validated nodes.

Theorem 12.9 (Summable false-acceptance theorem)**.** *For transaction t, let F_t be the event that an accepted child has nonpositive true declared gain. If*

$$\Pr(F_t \mid \mathcal{F}_{t-1}) \leq \delta_t \tag{394}$$

and $\sum_t \delta_t < \infty$, then only finitely many harmful acceptances occur almost surely. If the archive retains every previously validated best node, then $J^*(\mathcal{A}_t)$ is nondecreasing outside a finite random set of erroneous rounds.

Proof. The first claim is Theorem 10.11 and Borel–Cantelli. On every non-error round, accepted descendants have positive true gain, while retention prevents deletion of the previous best. Therefore archive regression can occur only on the finite error set. □

Definition 12.10 (Archive receipt completeness)**.** An archive insertion receipt is complete when it records:

1. the parent set and lineage-edge proofs;
2. the descriptor vector and uncertainty region;
3. the dominance and novelty comparisons used for retention;
4. every reusable repair schema or obstruction certificate introduced by the node;
5. the continuation signature and compression cell, when applicable;
6. the exact evidence namespace used for acceptance.

Archive result

A monotone best score is not enough. The archive must preserve future option value. Multiobjective frontiers, approximate covers, continuation signatures, and explicit repair or obstruction content make the retention rule part of the formal improvement mechanism rather than an implementation detail.

---

13 Metaproductivity

Metaproductivity is the capacity to produce useful descendants under a fixed budget. It must be measured separately from the task competence of the parent and from the number of candidates sampled.

13.1 Behavioral metaproductivity on foreign agents

Let $\mathcal{D}_{\text{hold}}$ be a distribution over foreign held-out agents that are not descendants of the evaluated lineage. Let K be a frozen improvement mechanism. Under horizon h and budget B, define

$$M_{h,B}^{\text{foreign}}(K) = \mathbb{E}_{A \sim \mathcal{D}_{\text{hold}}} \left[\max_{A' \in \text{Desc}_{h,B}(K,A)} J(A') - J(A) \right]. \tag{395}$$

The baseline, task distribution, model access, number of candidates, tool permissions, validation data, and wall-clock or compute budget must be matched.

Definition 13.1 (Bounded recursive self-improvement)**.** A lineage exhibits bounded recursive self-improvement over a declared generation set when:

1. protected task utility improves on a nontrivial held-out family;
2. $M_{h,B}^{\text{foreign}}$ increases under matched budgets;
3. task-relevant structural repair capacity increases on a held-out defect family;
4. evidence, provenance, witness separation, and rollback invariants remain valid.

Definition 13.2 (Budget acceleration)**.** Let $B_t(\gamma)$ be the least budget required by mechanism K_t to obtain expected foreign-agent gain at least γ. Budget acceleration holds on an interval I when

$$B_{t+1}(\gamma) < B_t(\gamma) \qquad \text{for every } \gamma \in I. \tag{396}$$

13.2 The best-of-k budget confound

Proposition 13.3 (Expected best-of-k gain)**.** *Let $G_1, \dots, G_k$ be independent nonnegative candidate gains with common cumulative distribution function F and finite mean. Then*

$$\mathbb{E} \max_{1 \le i \le k} G_i = \int_0^\infty \left(1 - F(x)^k\right) \mathrm{d}x. \tag{397}$$

In particular, expected best gain is nondecreasing in k even when the candidate generator itself does not improve.

Proof. For a nonnegative random variable X, $\mathbb{E}X = \int_0^\infty \Pr(X > x)\,\mathrm{d}x$. Independence gives

$$\Pr\left(\max_i G_i \le x\right) = F(x)^k. \tag{398}$$

Substitution proves the identity. □

This identity is why candidate count and compute must be held fixed when comparing improvement mechanisms.

13.3 Paired foreign-agent estimation and distribution shift

Let K_0 and K_1 be frozen improvement mechanisms evaluated on the same independent foreign agents $A_1, \dots, A_n$. Define

$$G_{h,B}(K, A) = \max_{A' \in \mathrm{Desc}_{h,B}(K,A)} J(A') - J(A) \tag{399}$$

and assume

$$-G_{\max} \le G_{h,B}(K, A) \le G_{\max}. \tag{400}$$

For matched budgets and random seeds, let

$$D_i = G_{h,B}(K_1, A_i) - G_{h,B}(K_0, A_i), \qquad \overline{D}_n = \frac{1}{n}\sum_{i=1}^{n} D_i. \tag{401}$$

Theorem 13.4 (Paired metaproductivity certificate)**.** *For independent held-out agents and* $0 < \delta < 1$*, with probability at least* $1 - \delta$*,*

$$\mathbb{E}D_i \geq \overline{D}_n - 2\sqrt{2}\, G_{\max}\sqrt{\frac{\log(1/\delta)}{n}}. \tag{402}$$

A positive lower bound certifies higher foreign-agent metaproductivity for K_1 *relative to* K_0*.*

Proof. Each D_i lies in an interval of length $4G_{\max}$. The one-sided Hoeffding inequality gives

$$\Pr\left(\overline{D}_n - \mathbb{E}D_i \geq \varepsilon\right) \leq \exp\left(-\frac{n\varepsilon^2}{8G_{\max}^2}\right). \tag{403}$$

Solving for ε proves the bound. □

Pairing removes between-agent difficulty variation. It does not repair reuse of the same agents during mechanism design. A primary claim should use a final audit population revealed only after K_0, K_1, h, B, and the endpoints are frozen.

Let $\mathcal{D}$ and $\mathcal{D}'$ be two foreign-agent distributions and define

$$\mathrm{TV}(\mathcal{D}, \mathcal{D}') = \sup_{B} \left|\mathcal{D}(B) - \mathcal{D}'(B)\right|. \tag{404}$$

Proposition 13.5 (Metaproductivity shift bound)**.** *If* $|G_{h,B}(K, A)| \leq G_{\max}$*, then*

$$\left|M_{h,B}^{\mathcal{D}}(K) - M_{h,B}^{\mathcal{D}'}(K)\right| \leq 2G_{\max}\,\mathrm{TV}(\mathcal{D}, \mathcal{D}'). \tag{405}$$

Proof. Apply the total-variation expectation bound to the function $A \mapsto G_{h,B}(K, A)$. □

Thus metaproductivity transfer must state the foreign-agent population on which it is claimed.

13.4 Task-relevant structural metaproductivity

Raw repair rank is a geometric capacity measure. It becomes behaviorally meaningful only relative to a defect distribution.

Let V be a Hilbert defect space, let $L_t = \operatorname{im} D_t^{\mathrm{auth}}$, and let P_t be the orthogonal projector onto L_t. For a defect distribution μ, define the correctable energy

$$\mathcal{E}_t^{\mathrm{cor}}(\mu) = \mathbb{E}_{r\sim\mu}\,\|P_t r\|^2 \tag{406}$$

and the obstruction energy

$$\mathcal{E}_t^{\mathrm{ob}}(\mu) = \mathbb{E}_{r\sim\mu}\,\|(I - P_t)r\|^2. \tag{407}$$

Theorem 13.6 (Structural-to-behavioral identity). *Assume $L_t \subseteq L_{t+1}$ and let*

$$W = L_{t+1} \cap L_t^{\perp}. \tag{408}$$

Then

$$\mathcal{E}_{t+1}^{\mathrm{cor}}(\mu) - \mathcal{E}_t^{\mathrm{cor}}(\mu) = \mathbb{E}_{r\sim\mu} \|P_W r\|^2 \tag{409}$$

and

$$\mathcal{E}_t^{\mathrm{ob}}(\mu) - \mathcal{E}_{t+1}^{\mathrm{ob}}(\mu) = \mathbb{E}_{r\sim\mu} \|P_W r\|^2 . \tag{410}$$

Under quadratic residual loss and exact cancellation of the correctable component, this quantity is exactly the expected additional task-loss reduction made available by the expanded repair image.

Proof. Nested subspaces give the orthogonal decomposition

$$L_{t+1} = L_t \oplus W, \qquad P_{t+1} = P_t + P_W. \tag{411}$$

The two projected components are orthogonal, so

$$\|P_{t+1} r\|^2 = \|P_t r\|^2 + \|P_W r\|^2 . \tag{412}$$

The obstruction identity follows from Pythagoras. Under exact correction, the post-repair residual is $(I - P_t)r$, so loss reduction is $\|P_t r\|^2$. □

Corollary 13.7 (When rank expansion is behaviorally irrelevant). *A strict inclusion $L_t \subsetneq L_{t+1}$ yields no expected quadratic benefit on μ exactly when*

$$P_W r = 0 \qquad \mu\text{-almost surely.} \tag{413}$$

Proposition 13.8 (Finite-sample structural certificate). *Suppose $\|r\| \le R$ almost surely and $r_1, \dots, r_n$ are independent held-out defects. Let*

$$\widehat{\Delta}_{\mathrm{str}} = \frac{1}{n} \sum_{i=1}^{n} \|P_W r_i\|^2 . \tag{414}$$

Then with probability at least $1 - \delta$,

$$\left| \widehat{\Delta}_{\mathrm{str}} - \mathbb{E} \|P_W r\|^2 \right| \le R^2 \sqrt{\frac{\log(2/\delta)}{2n}}. \tag{415}$$

Proof. Each summand lies in $[0, R^2]$. Apply Hoeffding's inequality. □

13.5 Conservative meta-improvement

Proposition 13.9 (Foreign-transfer lower bound). *Suppose K_{t+1} retains every candidate available to K_t and adds one repair schema. Let E be a held-out event of probability p on which the new schema produces certified gain at least γ beyond the best candidate available to K_t, and suppose the new selector never chooses a worse candidate outside E. Then*

$$M_{h,B}^{\mathrm{foreign}}(K_{t+1}) - M_{h,B}^{\mathrm{foreign}}(K_t) \ge p\gamma. \tag{416}$$

Proof. The pointwise best gain improves by at least γ on E and does not decrease outside E. Taking expectations gives the bound. □

Warning 13.10 (Structural and behavioral metrics are complementary)**.** Structural growth is not sufficient when new directions have negligible probability, excessive cost, or poor selection. It is not necessary for short-run behavioral gain because better conditioning, ranking, or execution can improve performance within the same correction image. A strong result should report both.

Definition 13.11 (One-step recursive meta-improvement)**.** A lineage edge $K_t \to K_{t+1}$ is a one-step recursive meta-improvement when:

1. K_{t+1} was constructed by an authorized transaction whose proposal process used K_t;
2. the edge passed an independent witness and preserved the declared evidence and rollback invariants;
3. under matched budgets on a frozen foreign-agent distribution,

$$M_{h,B}^{\text{foreign}}(K_{t+1}) > M_{h,B}^{\text{foreign}}(K_t) \tag{417}$$

with a positive precommitted lower confidence bound.

The lineage condition excludes the weaker situation in which an external designer simply replaces K_t with an unrelated better mechanism.

Proposition 13.12 (Task improvement does not imply meta-improvement)**.** *There exist mechanisms K_0, K_1 such that K_1 has higher direct task utility on every task in a family, but*

$$M_{h,B}^{\text{foreign}}(K_1) < M_{h,B}^{\text{foreign}}(K_0). \tag{418}$$

Proof. Let K_1 be a perfect fixed solver that returns no useful modifications when asked to improve a foreign agent. Let K_0 be a weaker direct solver equipped with a repair procedure that improves every agent in the held-out family. Direct competence favors K_1, while improvement lift favors K_0. □

Metaproductivity result

Foreign-agent evaluation separates improvement competence from lineage-specific task competence. Matched candidate budgets remove the best-of-k confound. Correctable energy links structural repair-image expansion to exact expected loss reduction under a quadratic model and makes clear when a nominal rank increase has no task relevance.

14 Grammar Expansion and Open-Endedness

A fixed finite grammar can support only finitely many strict increases in repair rank. Open-ended improvement therefore requires controlled expansion of the types used to describe defects, repairs, evidence, or evaluators.

14.1 Finite fixed-grammar exhaustion

Theorem 14.1 (Finite structural-improvement bound)**.** *Let V be a fixed n-dimensional defect space and suppose*

$$\operatorname{im} D_0 \subsetneq \operatorname{im} D_1 \subsetneq \cdots \tag{419}$$

is a strict ascending chain. Then the chain contains at most

$$n - \operatorname{rank} D_0 \tag{420}$$

strict inclusions.

Proof. Every strict inclusion raises image dimension by at least one, while image dimension is bounded by n. □

Thus indefinite structural improvement requires expansion or refinement of at least one of

$$U_t^{\mathrm{auth}}, \qquad V_t, \qquad D_t, \qquad \mathcal{G}_t, \qquad \mathcal{V}_t. \tag{421}$$

14.2 Conservative grammar extensions

Let $\mathcal{G}_t$ be a typed grammar of defects, repairs, evidence, transformations, and proof rules.

Definition 14.2 (Conservative grammar extension)**.** An embedding

$$\iota_t : \mathcal{G}_t \hookrightarrow \mathcal{G}_{t+1} \tag{422}$$

is conservative when:

1. every old well-typed expression remains well typed;
2. every old derivation translates to a derivation;
3. the interpretation of every old expression on old evidence is unchanged;
4. protected evidence identifiers and provenance resources are transported identically;
5. new rules cannot derive an old-language contradiction unless a correction receipt explicitly retracts a previously unsound axiom or evidence interpretation.

Theorem 14.3 (Finite derivation preservation)**.** *Suppose ι_t is fully faithful on the proof-carrying fragment and commutes with the old semantics. Then every theorem derivable in $\mathcal{G}_t$ remains derivable with the same protected meaning in $\mathcal{G}_{t+1}$.*

Proof. Translate the finite derivation rule by rule. Full faithfulness preserves each rule instance, semantic commutation preserves interpretation, and identity on protected evidence preserves external reference. □

14.3 Directed grammar limits

Consider a chain

$$\mathcal{G}_0 \overset{\iota_0}{\hookrightarrow} \mathcal{G}_1 \overset{\iota_1}{\hookrightarrow} \mathcal{G}_2 \hookrightarrow \cdots \tag{423}$$

of conservative extensions.

Theorem 14.4 (Conservative directed limit)**.** *The union of translated syntax and proofs defines a directed-limit grammar $\mathcal{G}_\infty$ with the following property: every finite expression and every finite derivation in $\mathcal{G}_\infty$ appears at some finite stage. Every theorem proved at stage t retains the same protected meaning at every later stage and in $\mathcal{G}_\infty$.*

Proof. A finite expression or derivation uses finitely many symbols and rule instances, each introduced at a finite stage. Directedness places all of them in one later stage. Repeated application of Theorem 14.3 preserves every earlier theorem and its protected interpretation. □

Definition 14.5 (Approximate semantic debt)**.** If an extension preserves old semantics only approximately, let η_t bound its worst protected interpretation error after transport. Under nonexpansive semantic transports, the cumulative debt through stage T is

$$\eta_{0:T} \le \sum_{t=0}^{T-1} \eta_t. \tag{424}$$

Proposition 14.6 (Summable approximate conservativity)**.** *If protected semantic transports are nonexpansive and $\sum_t \eta_t < \infty$, then every old protected interpretation has a Cauchy image along the grammar chain, with total drift bounded by the tail sum.*

Proof. The difference between two later interpretations is bounded by the telescoping sum of intervening extension errors. Summability makes the sequence Cauchy and gives the stated tail bound. □

14.4 Grammar-boundary certificates

For grammar $\mathcal{G}$ with linearized authorized correction image $L_{\mathcal{G}}$, define

$$q_{\mathcal{G}}(r) = \operatorname{dist}(-r, L_{\mathcal{G}}). \tag{425}$$

This is the minimum unresolved first-order residual inside the current grammar.

Definition 14.7 (Relative grammar-expansion witness)**.** Let $\mathcal{G} \hookrightarrow \mathcal{G}'$ be a candidate extension and let r' be the transported defect. The extension exposes a missing correction direction with margin $\kappa > 0$ when

$$q_{\mathcal{G}'}(r') \le q_{\mathcal{G}}(r) - \kappa. \tag{426}$$

Proposition 14.8 (Expansion witness implies new correctability)**.** *If the old and new defect metrics agree on the embedded old space and the relative expansion margin is positive, then the new authorized correction image contains a direction not available in the old image.*

Proof. If the new image introduced no new correction direction relevant to r', the distance from $-r'$ to the new image would equal the distance to the embedded old image. A strict decrease contradicts that equality. □

Warning 14.9. A positive old residual $q_{\mathcal{G}}(r)$ proves only noncorrectability inside the current first-order grammar. It does not by itself distinguish a genuinely impossible repair from an omitted stratum, a nonlinear correction, a bad defect model, or corrupted evidence. Grammar expansion requires a separate hypothesis and validation receipt.

14.5 Dynamic obstruction exact sequence

Let $j : V \hookrightarrow V'$ be an injective evidence-conservative embedding of old defects into an expanded ontology. Let

$$C = \operatorname{im} D \subseteq V, \qquad C' = \operatorname{im} D' \subseteq V', \tag{427}$$

and assume preservation of old correctability:

$$j(C) \subseteq C'. \tag{428}$$

The embedding induces a map

$$\overline{j} : V/C \to V'/C'. \tag{429}$$

Theorem 14.10 (Old-resolution and new-obstruction exact sequence)**.** *There is an exact sequence*

$$0 \longrightarrow \frac{j^{-1}(C')}{C} \longrightarrow \frac{V}{C} \xrightarrow{\overline{j}} \frac{V'}{C'} \longrightarrow \frac{V'}{C' + j(V)} \longrightarrow 0. \tag{430}$$

The first nonzero term contains old obstruction classes newly resolved by the expanded repair calculus. The final quotient contains genuinely new obstruction directions not represented in the old ontology.

Proof. The kernel of $\overline{j}$ consists of classes $[v]$ with $j(v) \in C'$, which is $j^{-1}(C')/C$. The cokernel is

$$(V'/C')/((j(V) + C')/C') \cong V'/(C' + j(V)). \tag{431}$$

Exactness at the remaining terms follows from the quotient maps. □

The sequence separates two effects that a single obstruction dimension conflates: old defects becoming repairable and new defect distinctions becoming visible.

14.6 Representation-separation certificates

Let $\phi_{\mathcal{G}} : X \to Z_{\mathcal{G}}$ be the state representation expressible in grammar $\mathcal{G}$. A $\mathcal{G}$-measurable deterministic repair policy has the form

$$\pi = \widetilde{\pi} \circ \phi_{\mathcal{G}}. \tag{432}$$

Theorem 14.11 (Representation-separation obstruction)**.** *Suppose states x_0, x_1 satisfy*

$$\phi_{\mathcal{G}}(x_0) = \phi_{\mathcal{G}}(x_1), \tag{433}$$

but their externally validated repair sets are disjoint:

$$\mathcal{R}_{\text{valid}}(x_0) \cap \mathcal{R}_{\text{valid}}(x_1) = \varnothing. \tag{434}$$

Then no $\mathcal{G}$-measurable deterministic repair policy is valid on both states.

Proof. The policy receives the same grammar state for both inputs and therefore returns the same repair. No repair belongs to both valid sets. □

Such a pair is a constructive inadequacy certificate. A successful extension must add a distinction separating the states or introduce a repair valid for both.

14.7 Levels of improvement

Table 4: Four levels of improvement and their required certificates.

Level	Modification	Mathematical signature	Required certificate
I	Parameters or policies in a fixed grammar	Same types, altered state point	Held-out gain and invariant preservation
II	New repair operators in existing types	Expanded U^{auth} or repair operad	New correctable direction, transfer, and receipt
III	New defect, packet, or ontology types	Conservative $\mathcal{G}_t \hookrightarrow \mathcal{G}_{t+1}$	Old-proof replay, evidence identity, new-direction witness
IV	Evaluator refinement	Evolution outside an adequate path-flat core	Core retraction, replay, held-out risk improvement

Conjecture 14.12 (Open-set repair adequacy)**.** *There exist nontrivial agent families for which a lineage can detect that its current grammar omits a causally necessary stratum, construct a conservative extension, and improve foreign-agent metaproductivity without violating its adequate path-flat evidence core.*

Grammar result

A fixed finite defect grammar has a finite structural ceiling. A conservative directed grammar chain can preserve every finite old proof while adding new types. A relative expansion witness shows that a proposed extension genuinely opens a previously unavailable correction direction. It does not show that the system can discover the right extension autonomously.

15 Autocatalytic Repair Discovery

Recursive acceleration would require validated repair discoveries to improve the process that discovers further repairs. Counting raw patches is insufficient. A counted offspring must be novel, transferable, externally validated, and causally dependent on prior repair knowledge.

15.1 Repair-discovery closure

Let Ω be a finite universe of candidate repair-schema equivalence classes. For $S \subseteq \Omega$, let $\mathrm{Disc}(S)$ be the schemas independently validated using the currently available set S. Define

$$\Phi_{\text{disc}}(S) = S \cup \mathrm{Disc}(S). \tag{435}$$

Assumption 15.1 (Discovery monotonicity)**.** If $S \subseteq T$, then

$$\mathrm{Disc}(S) \subseteq \mathrm{Disc}(T). \tag{436}$$

This assumes that adding repair knowledge does not revoke access to old discovery procedures. Resource competition can violate the assumption and must then be modeled explicitly.

Theorem 15.2 (Finite repair-discovery closure)**.** *Under discovery monotonicity,* Φ_{disc} *has a least fixed point above any seed set* S_0*. Iteration*

$$S_{k+1} = \Phi_{\text{disc}}(S_k) \tag{437}$$

stabilizes after at most $|\Omega| - |S_0|$ *strict expansions.*

Proof. The powerset of Ω is a finite complete lattice and Φ_{disc} is monotone and inflationary. Each strict step adds at least one schema, so the stated bound follows. □

15.2 Causal offspring matrix

Partition validated schemas into m types. For a type-i schema, let N_{ij} count type-j descendants whose receipts include a successful necessity ablation showing that removal of the parent schema eliminates the descendant discovery under matched evidence and budget. Define

$$M_{ij} = \mathbb{E}N_{ij}. \tag{438}$$

This definition excludes descendants that merely share ancestry or evidence with the parent.

Definition 15.3 (Improvement reproduction number)**.** The local repair-discovery reproduction number is

$$R_{\text{imp}} = \text{rad}(M). \tag{439}$$

Conditional Theorem 15.4 (Finite-type branching threshold)**.** *Assume validated repair discovery is accurately modeled by an irreducible finite-type Galton–Watson process with mean progeny matrix* M *and is not deterministically one-child preserving. Then:*

1. *if* $R_{\text{imp}} \leq 1$*, the repair-discovery lineage becomes extinct almost surely;*
2. *if* $R_{\text{imp}} > 1$*, it has positive probability of indefinite survival.*

Warning 15.5. The branching assumptions are strong. Real discoveries share tools and evidence, compete for compute, alter their own type space, and are selected by an archive. The theorem is a local model of schema production, not a theorem of intelligence explosion or indefinite utility growth.

15.3 Autocatalytic cores

Construct the directed type graph with an edge $i \to j$ when $M_{ij} > 0$. Its strongly connected components identify mutually enabling schema families.

Theorem 15.6 (Localization of supercriticality)**.** *After permuting schema types,* M *has block upper-triangular form with diagonal blocks* $M_1, \ldots, M_k$ *given by strongly connected components. Therefore*

$$\text{rad}(M) = \max_{1 \leq r \leq k} \text{rad}(M_r). \tag{440}$$

In particular, $R_{\text{imp}} > 1$ *only if at least one strongly connected schema family is itself supercritical.*

Proof. Topologically order the condensation graph. The resulting matrix is block upper triangular, and its spectrum is the union of the spectra of its diagonal blocks. □

A supercritical component is an autocatalytic repair core. Downstream types may be abundant without contributing to continued discovery.

15.4 Robust spectral certification

Estimated offspring entries are uncertain. Componentwise intervals give a simple robust certificate.

Theorem 15.7 (Spectral interval bound)**.** *Let L, M, U be nonnegative square matrices satisfying*

$$L \le M \le U \tag{441}$$

entrywise. Then

$$\operatorname{rad}(L) \le \operatorname{rad}(M) \le \operatorname{rad}(U). \tag{442}$$

Consequently, $\operatorname{rad}(L) > 1$ *certifies supercriticality for every matrix in the interval, while* $\operatorname{rad}(U) \le 1$ *certifies subcriticality.*

Proof. The spectral radius of a nonnegative matrix is monotone under entrywise order. This follows, for example, by applying the Collatz–Wielandt characterization to positive test vectors. □

Proposition 15.8 (Perron sensitivity)**.** *Let M be irreducible with simple Perron root ρ, positive right eigenvector r, and positive left eigenvector ℓ, normalized by $\ell^{\mathsf{T}} r = 1$. Then*

$$\frac{\partial \rho}{\partial M_{ij}} = \ell_i r_j. \tag{443}$$

Proof. Differentiate $Mr = \rho r$ with respect to M_{ij} and left-multiply by ℓ^{T}. The terms involving the derivative of r cancel because $\ell^{\mathsf{T}} M = \rho \ell^{\mathsf{T}}$, leaving $\ell_i r_j$. □

The sensitivity identifies which validated causal edge would most increase the local reproduction number. It does not determine whether increasing that edge is safe, cheap, or behaviorally useful.

15.5 Resource normalization and time variation

Let $M(B)$ be the offspring matrix under a fixed discovery budget B. Define the critical budget

$$B_c = \inf \{ B : \operatorname{rad}(M(B)) > 1 \}. \tag{444}$$

Comparisons of R_{imp} are meaningful only at matched budgets and evidence access.

For a time-varying sequence M_t, let z_t be a row vector of expected schema counts. Under the linear mean model,

$$\mathbb{E} z_T = z_0 M_0 M_1 \cdots M_{T-1}. \tag{445}$$

Define the upper mean-growth exponent

$$\lambda_{\mathrm{mean}} = \limsup_{T \to \infty} \frac{1}{T} \log \| M_0 M_1 \cdots M_{T-1} \|. \tag{446}$$

A positive value indicates exponential growth of the linearized expected count. It does not imply almost-sure survival under dependence or resource competition.

15.6 Saturation and the distinction from utility

Proposition 15.9 (Finite-schema saturation)**.** *If only finitely many inequivalent schemas may ever be counted and each schema is counted at most once, then the discovery process produces only finitely many births, regardless of a transient estimate* $R_{\mathrm{imp}} > 1$*.*

Proof. The total number of counted births is bounded by the number of available equivalence classes. □

Theorem 15.10 (Bounded-utility gain saturation)**.** *Suppose protected task utility satisfies* $0 \le J \le 1$ *and every committed task-improving schema raises deployed utility by at least* $\gamma > 0$*. Starting from utility* J_0*, at most*

$$\left\lfloor \frac{1 - J_0}{\gamma} \right\rfloor \tag{447}$$

such deployments can occur.

Proof. After k such gains, utility is at least $J_0 + k\gamma \le 1$. □

A supercritical repair-discovery phase can therefore coexist with shrinking gains, task-family expansion, evaluator refinement, or eventual saturation. It cannot imply infinitely many fixed-size gains in a bounded utility.

Proposition 15.11 (Autocatalysis does not imply beneficial improvement)**.** *There exist repair-discovery processes with* $R_{\mathrm{imp}} > 1$ *whose discovered schemas have zero or negative protected utility.*

Proof. Take any supercritical branching process of syntactically novel schemas and assign protected gain zero to every schema. Alternatively, assign negative gain while allowing the schemas to generate further schemas. The reproduction matrix is unchanged, while protected utility does not improve. □

Therefore a credible autocatalysis claim requires all of the following:

1. causal necessity ablations for offspring edges;
2. foreign-system transfer;
3. structural novelty or a conservative grammar distinction;
4. external witness validation;
5. matched resource budgets;
6. a separate protected-utility or metaproductivity result.

Block 4 result

The recursive layer now distinguishes transactional noncircularity, robust invariant closure, repeated productive closure, archive option value, foreign-agent metaproductivity, conservative grammar growth, and causal repair autocatalysis. None can substitute for the others. In particular, safe recurrence is not productive recurrence, repair-schema reproduction is not protected utility, and a fixed finite grammar cannot support indefinite strict structural expansion.

16 Empirical Realization Contract

The preceding sections define a mathematical architecture. An empirical realization requires a separate contract connecting abstract objects to executable states, traces, data partitions, and claim boundaries. Without such a contract, successful code execution can be reported as support for a theorem that the runtime never actually instantiates.

16.1 Claim ladder

Definition 16.1 (Empirical claim ladder)**.** An empirical SRIC result is assigned to the strongest level whose requirements have all been met:

1. **mechanical realization**: the state machine, receipts, digests, rollback paths, and fixed-point routines execute as specified;
2. **finite semantic realization**: executable states refine a declared finite abstract model and satisfy its frozen endpoints;
3. **causal identification**: intervention or ablation changes the target outcome while matched controls do not;
4. **foreign repair transfer**: a repair schema improves held-out systems outside its source lineage under matched resources;
5. **metaproductive improvement**: a descendant repair calculus improves foreign systems better than its parent calculus;
6. **open-set recursive improvement**: the system expands its grammar or ontology and preserves prior evidence while improving on an unbounded sequence of novel defect families.

The ladder is monotone in evidential strength but not automatic. A level-k result does not imply level $k+1$. In particular, deterministic replay proves runtime stability relative to a package, not causal identification or metaproductivity.

Warning 16.2 (Fixture success is not an RSI result)**.** A finite fixture may intentionally contain a repair that the reference system is designed to find. Passing such a fixture verifies formal mechanics and can falsify an implementation. It does not establish learned, autonomous, open-ended, or indefinitely accelerating recursive self-improvement.

16.2 Benchmark instances

Definition 16.3 (Empirical benchmark instance)**.** A benchmark instance is a tuple

$$\mathfrak{B} = (\mathsf{E}, \mathsf{A}, \mathsf{G}, \mathsf{P}, \mathsf{W}, \mathsf{V}, \mathsf{D}, \mathsf{S}, \mathsf{B}, \mathsf{R}), \tag{448}$$

where:

- E is the environment or transition process;
- A is the agent family and functional atlas;
- G is the repair grammar and candidate generator;
- P is the permitted probe and intervention language;

- W is the witness, checkpoint, and effect-mediation layer;
- V is the protected evaluation context;
- D is the data and evidence partition;
- S is the seed, randomization, and split schedule;
- B is the matched resource and candidate budget;
- R is the preregistered claim and receipt schema.

The data partition is itself typed:

$$\mathsf{D} = \mathsf{D}_{\text{construct}} \sqcup \mathsf{D}_{\text{calibrate}} \sqcup \mathsf{D}_{\text{select}} \sqcup \mathsf{D}_{\text{foreign}} \sqcup \mathsf{D}_{\text{audit}}. \tag{449}$$

Construction data may be used to create schemas. Calibration data may set thresholds. Selection data may choose among frozen candidates. Foreign data estimate metaproductivity. Audit data are not available until the final claim object has been sealed.

Definition 16.4 (Claim object)**.** A claim object is

$$\mathfrak{C} = (\mathfrak{B}, \Theta, H, E, \Pi, \Gamma), \tag{450}$$

where Θ is the frozen hypothesis and endpoint set, H is the preoutcome commitment digest, E is the revealed evidence, Π is the proof-carrying receipt collection, and Γ is the claim boundary.

A result is reproducible only if an independent execution can reconstruct $\mathfrak{C}$ from the released package without consulting undeclared mutable state.

16.3 Abstract-runtime refinement

Let

$$\mathcal{M} = (S, \mathcal{T}, O) \tag{451}$$

be an abstract transition system with state set S, authorized labeled transitions $\mathcal{T}$, and protected observation map O. Let

$$\widehat{\mathcal{M}} = (\widehat{S}, \widehat{\mathcal{T}}, \widehat{O}) \tag{452}$$

be an executable implementation.

Definition 16.5 (Forward implementation refinement)**.** A map

$$\alpha : \widehat{S} \to S \tag{453}$$

forward-refines the runtime when:

1. every executable initial state maps to an authorized abstract initial state;
2. for every executable transition

$$\widehat{s} \xrightarrow{a} \widehat{s}', \tag{454}$$

there is an abstract transition path

$$\alpha(\widehat{s}) \xrightarrow{a,*} \alpha(\widehat{s}'); \tag{455}$$

3. protected observations commute:

$$\widehat{O}(\widehat{s}) = O(\alpha(\widehat{s})); \tag{456}$$

4. receipt compilation commutes with abstraction up to the declared canonical encoding.

Theorem 16.6 (Runtime-to-model trace soundness)**.** *Suppose α is a forward implementation refinement. Then every finite executable trace maps to an authorized abstract trace with the same protected observation sequence. Consequently, any universal safety property over abstract protected observations is inherited by the runtime traces covered by the refinement.*

Proof. Induct on executable trace length. The initial condition gives the base case. Forward simulation maps each executable step to an abstract path with matching endpoints. Observation commutation preserves the protected trace. Universal trace properties are closed under this mapping. □

The converse does not follow. An abstract model may admit behaviors not implemented by the runtime. Therefore runtime testing cannot establish abstract completeness unless the implementation additionally satisfies a backward simulation or exhaustive state correspondence.

Definition 16.7 (Finite exact realization)**.** A finite runtime exactly realizes $\mathcal{M}$ when α is surjective on reachable abstract states and every abstract authorized edge has at least one executable realizing path with matching receipt and observation.

Theorem 16.8 (Exhaustive finite endpoint theorem)**.** *Let $\mathcal{M}$ and $\widehat{\mathcal{M}}$ be finite, exactly related, and let $Q_1, \dots, Q_m$ be decidable state or transition predicates. Exhaustive enumeration of all reachable executable states and authorized edges decides whether each Q_i holds over the realized finite model.*

Proof. The reachable executable graph is finite. Exact realization supplies every reachable abstract state and authorized edge. Evaluate each decidable predicate on the finite enumeration. □

16.4 Frozen endpoints and claim gates

A block-level empirical package should separate *endpoints* from *interpretations*. An endpoint is a Boolean or numerical condition computed from sealed evidence. An interpretation is the claim that the endpoint supports.

Definition 16.9 (Claim gate)**.** For claim level ℓ, let

$$G_\ell(E, \Pi) \in \{0, 1\} \tag{457}$$

be a precommitted predicate over revealed evidence and receipts. The claim is admissible only if every prerequisite gate is one:

$$\text{Admissible}(\ell) \iff \bigwedge_{j \le \ell} G_j(E, \Pi) = 1. \tag{458}$$

Recommended gates are:

Gate	Target	Minimum evidence
G_{pkg}	Package integrity	Exact manifest, canonical digests, clean extraction, successful verifier
G_{mech}	Mechanics	Legal phase traces, receipt verification, checkpoint restoration, deterministic replay
G_{sem}	Finite semantics	Abstract-runtime refinement and exhaustive frozen endpoints
G_{cause}	Causal role	Intervention, necessity ablation, matched negative controls
G_{foreign}	Transfer	Held-out systems, matched candidate budget, protected evaluator
G_{meta}	Metaproductivity	Parent-child repair-calculus comparison on frozen foreign systems
G_{open}	Open-set recursion	Conservative grammar extension, prior-proof replay, novel externally validated defect families

Proposition 16.10 (Gate monotonicity)**.** *If claim admissibility is defined by cumulative conjunction, failure of a lower gate blocks every higher claim while preserving any lower claim whose own gates pass.*

Proof. Immediate from the conjunction defining Admissible(ℓ). □

Block 5 empirical principle

The implementation is treated as a refinement target and falsification instrument. A package can establish that a finite mechanism was realized and that declared synthetic controls behaved as expected. It cannot promote itself to a stronger claim level without separately satisfying the corresponding causal, foreign-transfer, metaproductivity, and open-set gates.

17 Finite Model-Checking Realization

The reference fixture instantiates the distinction among local correctability, gluing compatibility, semantic stability, and productive closure in four exact cases. The fixture is deliberately small enough for complete enumeration and direct numerical inspection.

17.1 Four frozen cases

Let each local correction space and defect space be $\mathbb{R}^2$ unless otherwise stated. Let the tolerance satisfy $0 < \tau < 10^{-9}$.

Case F1: local obstruction. Set

$$D_{\text{loc}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad r_{\text{loc}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{459}$$

Then

$$P_{\mathrm{im}\, D_{\mathrm{loc}}} r_{\mathrm{loc}} = 0, \qquad (I - P_{\mathrm{im}\, D_{\mathrm{loc}}}) r_{\mathrm{loc}} = r_{\mathrm{loc}}, \tag{460}$$

so the obstruction norm is one.

Case F2: gluing obstruction. Take $D_0 = D_1 = D_2 = I_2$, so every local defect is exactly correctable. On the oriented triangle with edges $(0,1)$, $(1,2)$, and $(0,2)$, set the scalar overlap mismatch

$$b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{461}$$

For the normalized cycle vector

$$c = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \tag{462}$$

we obtain

$$\left| c^{\top} b \right| = \frac{1}{\sqrt{3}} > 0. \tag{463}$$

Thus local correction succeeds while descent fails.

Case F3: flat repair. Set

$$D_{\mathrm{flat}} = I_2, \qquad r_{\mathrm{flat}} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \tag{464}$$

with zero overlap mismatch and identity evaluator transport. The unique minimum-norm repair is

$$u_{\mathrm{flat}} = -r_{\mathrm{flat}}. \tag{465}$$

Case F4: evaluator holonomy. Use an exactly correctable local defect and zero overlap mismatch. Let evaluator transports around a three-edge cycle be

$$T_{01} = I_2, \qquad T_{12} = I_2, \qquad T_{20} = \mathrm{diag}(1, -1). \tag{466}$$

For protected evaluator direction e_2,

$$H_\gamma e_2 - e_2 = -2e_2, \qquad \| H_\gamma e_2 - e_2 \| = 2. \tag{467}$$

The candidate is locally repairable and globally gluable, but its evaluator semantics are not path flat.

17.2 Classifier and completeness on the fixture

The executable classifier applies the ordered rule

$$\mathrm{Class}(r,b,H)=\begin{cases}\texttt{LOCAL_OBSTRUCTION}, & \left\|P_{(\operatorname{im} D)^{\perp}}r\right\|>\tau,\\ \texttt{GLUING_OBSTRUCTION}, & \|P_{\ker B^{\top}}b\|>\tau,\\ \texttt{EVALUATOR_HOLONOMY}, & \|(H-I)e\|>\tau,\\ \texttt{FLAT_REPAIR}, & \text{otherwise.}\end{cases} \tag{468}$$

The order is part of the receipt because a case can exhibit more than one defect. The fixture isolates the four modes, so the ordered rule is unambiguous.

Theorem 17.1 (Exact classification of the four-case fixture)**.** *For every $0<\tau<1/\sqrt{3}$, the classifier returns, respectively,*

$$\textit{LOCAL_OBSTRUCTION},\quad \textit{GLUING_OBSTRUCTION},\quad \textit{FLAT_REPAIR},\quad \textit{EVALUATOR_HOLONOMY} \tag{469}$$

for Cases F1 through F4.

Proof. Case F1 has obstruction norm one. Case F2 has zero local obstruction and cycle residual $1/\sqrt{3}$. Case F3 has all three residuals zero. Case F4 has zero local and gluing residuals but evaluator defect two. □

17.3 Finite closure game

The minimal closure model has states

$$S=\{s_{\mathrm{base}},s_{\mathrm{improved}},s_{\mathrm{invalid}}\} \tag{470}$$

and valid set

$$K=\{s_{\mathrm{base}},s_{\mathrm{improved}}\}. \tag{471}$$

With intact rollback, authorized complete transactions have successor relation

$$\mathrm{Post}(s_{\mathrm{base}})=\{s_{\mathrm{base}},s_{\mathrm{improved}}\},\qquad \mathrm{Post}(s_{\mathrm{improved}})=\{s_{\mathrm{improved}}\}. \tag{472}$$

Failed candidates are restored to the base state before the complete transaction ends.

When rollback is ablated, an invalid branch becomes an authorized complete successor:

$$s_{\mathrm{base}}\longrightarrow s_{\mathrm{invalid}}. \tag{473}$$

Proposition 17.2 (Rollback ablation removes universal base closure)**.** *In the intact complete-transaction model, s_{base} belongs to the universal invariant closure. When the invalid edge above is authorized and $s_{\mathrm{invalid}}\notin K$, s_{base} is removed by the first descending fixed-point update.*

Proof. In the intact model every complete successor of s_{base} lies in K. Under ablation, $\mathrm{Post}(s_{\mathrm{base}})\not\subseteq K$, so $s_{\mathrm{base}}\notin\Psi(K)$. □

Controlled closure remains distinct. A no-op action can preserve s_{base} even when an unsafe raw edit is available. Productive closure additionally requires a certified transition to s_{improved}.

17.4 Frozen finite endpoints

The reference fixture freezes the following endpoints before execution:

ID	Endpoint	Expected result
F01	All four cases are classified exactly	pass
F02	Local obstruction norm equals 1	pass
F03	Triangle gluing residual equals $1/\sqrt{3}$	pass
F04	Flat repair has zero local, gluing, and evaluator residual	pass
F05	Evaluator-holonomy defect equals 2	pass
F06	Intact universal closure contains the base state	pass
F07	Rollback ablation removes the base state from universal closure	pass
F08	Controlled closure contains the base state	pass
F09	A productive certified transition exists in the flat case	pass

17.5 Reference execution

The Block 5 package evaluates the exact matrices above. The resulting diagnostics are:

Case	Local obstruction	Cycle residual	Evaluator defect	Classification
Local obstruction	1.000000	0	0	local obstruction
Gluing obstruction	0	0.577350	0	gluing obstruction
Flat repair	0	0	0	flat repair
Evaluator holonomy	0	0	2.000000	evaluator holonomy

Table 7: Exact outputs of the finite correctability and descent fixture.

All nine frozen endpoints passed in the reference execution. This is an exhaustive result for the released finite fixture, not for arbitrary agent architectures.

18 Executable Transaction and Receipt Kernel

The reference kernel realizes the witness-separated transaction semantics as a small deterministic state machine. Its purpose is to make circularity, illegal edits, rollback, and receipt verification directly testable.

18.1 Runtime types

An executable state is a finite canonical map

$$\widehat{s} : \mathsf{Key} \to \mathsf{Value}. \tag{474}$$

A witness policy is

$$\widehat{W} = (K_{\text{protected}}, \gamma_{\min}, r_{\max}, p_{\text{prov}}, e_{\text{irr}}, v_W), \tag{475}$$

where $K_{\text{protected}}$ is the immutable-key set, $\gamma_{\min}$ is the minimum protected gain, $r_{\max}$ is the maximum accepted risk, p_{prov} indicates whether provenance is mandatory, e_{irr} controls irreversible effects, and v_W is the witness version.

A candidate contains:

$$\widehat{\delta} = (n, f, K_{\text{edit}}, E_{\text{irr}}, \rho, \widehat{g}), \tag{476}$$

where f is a sandboxed state transformation, K_{edit} is its declared edit mask, E_{irr} is its declared irreversible-effect set, ρ is a provenance token, and $\widehat{g}$ is the sealed prediction.

18.2 Phase machine

Every complete execution must follow exactly:

$$\texttt{PREPARED} \to \texttt{SEALED} \to \texttt{EXECUTED} \to \texttt{REVEALED} \to \texttt{DECIDED} \to \texttt{FINALIZED}. \tag{477}$$

The checkpoint, parent digest, candidate metadata digest, witness digest, and prediction are fixed before protected evidence is revealed.

Definition 18.1 (Kernel acceptance predicate)**.** For revealed protected evidence E, the reference acceptance predicate is

$$\begin{aligned} \text{Accept}(\widehat{s}, \widehat{s}', \widehat{\delta}, E) \iff\ & K_{\text{edit}} \cap K_{\text{protected}} = \varnothing \\ & \wedge\ \widehat{s}'|_{K_{\text{protected}}} = \widehat{s}|_{K_{\text{protected}}} \\ & \wedge\ E[\text{utility_gain}] \geq \gamma_{\min} \\ & \wedge\ E[\text{risk}] \leq r_{\max} \\ & \wedge\ E[\text{provenance_ok}] = 1 \\ & \wedge\ E_{\text{irr}} = \varnothing. \end{aligned} \tag{478}$$

Theorem 18.2 (Protected-key noninterference)**.** *If a candidate declares or produces a change to a protected key, the reference acceptance predicate is false.*

Proof. A declared protected edit violates the first conjunct. An undeclared produced edit violates the second conjunct. □

Proposition 18.3 (Exact in-memory rollback)**.** *Assume state effects are confined to a deep-copied sandbox and the checkpoint is immutable. If acceptance is false, the final state is byte-canonically equal to the checkpoint state.*

Proof. The kernel replaces the sandbox result by a fresh deep copy of the checkpoint. Canonical serialization is deterministic, so their digests agree. □

The proposition is intentionally narrow. It does not apply to undeclared file writes, network actions, hardware state, or other effects outside the mediated state object.

18.3 Canonical receipts and hash chaining

Let cj denote canonical JSON serialization with sorted keys and normalized arrays. For payload x, define

$$h(x) = \mathrm{SHA256}(\mathrm{cj}(x)). \tag{479}$$

A transaction receipt records parent, candidate, witness, checkpoint, phase trace, evidence, decision, rejection reasons, final state, and previous receipt hash.

Definition 18.4 (Receipt chain)**.** For receipts $\Pi_1, \ldots, \Pi_T$, define

$$\Pi_t.\mathrm{previous} = \begin{cases} \varnothing, & t = 1, \\ h(\Pi_{t-1}), & t > 1. \end{cases} \tag{480}$$

Proposition 18.5 (Prefix tamper detection)**.** *If SHA-256 collision resistance and canonical serialization are assumed, changing a finalized receipt in a linked chain changes every descendant link unless the attacker recomputes the descendant receipts.*

This is an integrity statement, not a semantic authorization theorem. The witness must still determine whether the evidence and action were authorized.

18.4 Reference attack pair

The smoke test begins from a state with protected evaluator, evidence, and provenance fields. A legitimate planner repair changes only utility and planner state, predicts a gain of 0.12, and is committed after protected evaluation reveals a gain above 0.05.

A second candidate declares an edit to the protected evaluator and attempts to replace it while reporting a larger utility. The kernel detects both the declared protected edit and the realized protected-state change, then restores the checkpoint.

Candidate	Protected outcome	Decision	Verification
Planner repair	Utility rises from 0.50 to 0.62 with evaluator unchanged	commit	receipt valid
Evaluator capture	Candidate attempts evaluator replacement	rollback	exact checkpoint hash restored

Table 8: Reference transaction-kernel positive and adversarial controls.

The two receipts form a valid chain. The adverse receipt contains the explicit reasons

$$\texttt{protected edit declared: evaluator} \tag{481}$$

and

$$\texttt{protected state changed: evaluator}. \tag{482}$$

All six transaction endpoints passed: legitimate commit, adversarial rollback, protected evaluator preservation, exact rollback, chain linkage, and independent receipt verification.

18.5 Reference module boundary

Module	Responsibility
`canonical.py`	Canonical JSON conversion, SHA-256 objects, stable result serialization
`linalg.py`	Pseudoinverse repair, obstruction decomposition, triangle cycle residual, holonomy defect
`receipt.py`	Evidence records, transaction receipts, receipt verification
`transaction.py`	Witness policy, candidate interface, phase machine, commit or rollback
`fixture.py`	Four frozen cases and finite closure checks
`anonymous_env.py`	Linear anonymous-change environment, bounded probes, posterior classifier
`benchmark.py`	Foreign synthetic agents, matched-budget arms, structural and behavioral metrics
`run_all.py`	Reproducible suite execution and machine-readable summaries
`verify_package.py`	Manifest coverage and SHA-256 verification

19 Anonymous Continuous Repair Environment

A labeled episode that announces "sensor fault" or "world change" does not test causal attribution. The Block 5 environment hides a latent event type and exposes only continuous observations and bounded probe outcomes.

19.1 Linear environment family

Let the latent world state be $x_t \in \mathbb{R}^2$, action $a_t \in \mathbb{R}^2$, observation $y_t \in \mathbb{R}^3$, and memory state $m_t \in \mathbb{R}^2$. The baseline dynamics are

$$x_{t+1} = Ax_t + Ba_t, \tag{483}$$

$$y_t = Cx_t + b_s, \tag{484}$$

$$m_{t+1} = \lambda m_t + (1-\lambda) y_{t,1:2}, \tag{485}$$

and the reported evaluator score is

$$q_t = w^\top y_t. \tag{486}$$

The redundant observation map is

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \tag{487}$$

so the protected consistency relation is

$$y_{t,3} = y_{t,1} + y_{t,2} \tag{488}$$

when the sensor channel is intact.

The latent event variable is

$$z \in \{\text{none}, \text{internal}, \text{world}, \text{sensor}, \text{memory}, \text{evaluator}\}. \tag{489}$$

Each event changes one declared mechanism:

- internal: the action-response map B changes;
- world: the passive dynamics A change;
- sensor: the redundant sensor receives an additive bias;
- memory: the retention coefficient λ changes;
- evaluator: the reported weight vector w changes;
- none: all baseline mechanisms remain fixed.

The event label is not supplied to the classifier.

19.2 Bounded probe vector

From a common probe state x_0 and bounded action a, define the feature vector

$$\phi(z) = \begin{bmatrix} \phi_{\text{passive}} \\ \phi_{\text{actuation}} \\ \phi_{\text{sensor}} \\ \phi_{\text{memory}} \\ \phi_{\text{evaluator}} \end{bmatrix}. \tag{490}$$

Its components are, respectively:

1. deviation from the baseline no-action continuation;
2. deviation of the action-conditioned response from the baseline action response;
3. violation of redundant-sensor consistency;
4. deviation from baseline memory update;
5. discrepancy between reported and protected evaluator scores.

The observed probe is

$$\widetilde{\phi} = \phi(z) + \eta, \qquad \eta \sim \mathcal{N}(0, \sigma^2 I_5). \tag{491}$$

19.3 Posterior attribution

Let $\mu_z = \phi(z)$ be the frozen event prototype. With equal priors and matched isotropic noise, the posterior is

$$p(z \mid \widetilde{\phi}) = \frac{\exp\left(-\left\|\widetilde{\phi} - \mu_z\right\|^2 / (2\sigma^2)\right)}{\sum_{z'} \exp\left(-\left\|\widetilde{\phi} - \mu_{z'}\right\|^2 / (2\sigma^2)\right)}. \tag{492}$$

A repair is attempted only when

$$\max_z p(z \mid \widetilde{\phi}) \geq \theta \tag{493}$$

and the maximizing event is not *none*. Otherwise the system abstains.

Theorem 19.1 (Prototype-separation error bound)**.** *Let*

$$\Delta = \min_{z \neq z'} \|\mu_z - \mu_{z'}\| . \tag{494}$$

Under equal priors and isotropic Gaussian noise, nearest-prototype classification satisfies

$$\Pr(\widehat{z} \neq z) \leq (|\mathcal{Z}| - 1) \exp\left(-\frac{\Delta^2}{8\sigma^2}\right). \tag{495}$$

Proof. For each competitor z', an error requires the Gaussian projection onto the unit vector joining μ_z and $\mu_{z'}$ to exceed half their separation. Apply the standard one-dimensional Gaussian tail bound and union bound over competitors. □

The bound is conservative and does not account for abstention. It nevertheless shows why probe design should maximize the smallest relevant prototype separation rather than only average prediction accuracy.

Theorem 19.2 (Anonymous-change impossibility)**.** *If two latent events induce identical distributions over every permitted probe and observation trace, no decision rule measurable with respect to those traces can distinguish the events with accuracy above the larger prior probability.*

Proof. The posterior odds remain equal to the prior odds because the likelihood ratio is one for every measurable trace. □

This theorem forces abstention when the permitted intervention language is not separating.

19.4 Typed repair responses

The event-specific response is typed rather than uniformly described as a self-edit:

Event	Appropriate response	Incorrect self-repair
Internal	Restore or adapt the action-response component	Rewrite the world model while retaining a broken actuator model
World	Update the predictive world model	Force the environment back into the old prediction
Sensor	Recalibrate or quarantine the corrupted channel	Rewrite latent world state to match biased observations
Memory	Repair retention or reconstruction	Treat missing memory as a world event
Evaluator	Restore or quarantine score semantics under protected evidence	Optimize the candidate-owned evaluator
None	Abstain	Modify a functioning system to reduce noise in its own residual

19.5 Reference positive-control execution

The released smoke configuration uses 80 episodes per event, noise scale $\sigma = 0.06$, posterior threshold $\theta = 0.85$, and fixed seed 20260829. The minimum pairwise prototype separation is 0.110955.

Metric	Reference value
Non-none attribution accuracy	0.950000
Overall repair-or-abstain success	0.908333
False repair rate under no change	0.037500
No-change abstention accuracy	0.962500
Mean posterior confidence	0.938424

Table 11: Anonymous-change synthetic positive-control results.

Internal, world, sensor, and evaluator cases were perfectly separated in this seed. The principal overlap occurred between the memory-change and no-change prototypes. The result is therefore useful as a diagnostic: it demonstrates both high-separation cases and a boundary at which calibrated abstention matters.

Warning 19.3 (Matched-model limitation)**.** The classifier knows the six prototype families used to generate the environment. It does not discover new latent mechanisms, revise its packet atlas, or learn an open-ended causal ontology. The correct claim is a synthetic anonymous-change identification fixture.

20 Matched-Budget Recursive Repair Benchmark

The recursive-repair benchmark measures foreign-agent gain under a common candidate budget. It is an engineered positive-control environment designed to expose solver, search, transport, witness, and abstraction confounds independently.

20.1 Foreign synthetic agents

Each foreign agent has a canonical defect

$$d \in \mathbb{R}^6, \tag{496}$$

a positive diagonal importance vector w, and an invertible architecture transform

$$T \in \mathbb{R}^{6\times 6}. \tag{497}$$

The runtime-visible defect is

$$\widetilde{d} = Td. \tag{498}$$

Protected utility is

$$J_T(\widetilde{d}) = \exp\left(-\sum_{i=1}^{6} w_i (T^{-1}\widetilde{d})_i^2\right). \tag{499}$$

A candidate runtime patch δ produces gain

$$g(\delta) = J_T(\widetilde{d} + \delta) - J_T(\widetilde{d}). \tag{500}$$

The transforms combine permutation, sign, and scale changes. Therefore canonical repairs must be transported; direct coordinate reuse generally fails.

20.2 Matched arms

Every arm receives candidate budget $B = 8$ and the same 96 frozen agents.

Arm	Executable mechanism
Static	Returns the parent with no modification
Global edit	Samples whole-vector edits and contains a deliberately exploitable apparent-score component
Archive only	Samples noisy descendants around the current visible defect and retains the best protected candidate
Packet only	Corrects large visible coordinates without causal type or architecture transport
Causal only	Identifies a dominant canonical cause but applies the repair without target-coordinate transport
Full SRIC	Uses typed canonical repairs, architecture transport, protected evaluation, and witness rejection
Full SRIC meta	Adds an engineered schema-abstraction positive control for the sixth held-out defect type

The last arm is not a learned meta-improver. It is a positive control demonstrating what the foreign-agent metric should report when a new repair schema is actually made available.

20.3 Search and evaluator confounds

The global-edit arm contains a candidate whose apparent score receives an exploitable bonus when its rewrite norm is large. The protected evaluator does not receive this bonus. This creates a controlled instance of evaluator capture:

$$\widehat{J}(\delta) > \widehat{J}(0) \qquad \text{while} \qquad J(\delta) < J(0). \tag{501}$$

A witness-aware arm ranks and commits by J, not $\widehat{J}$.

The archive arm can perform well without a causal repair calculus because it searches several noisy candidates and retains the best. Candidate-budget matching is therefore mandatory. Let

$$G_{a,j,1}, \ldots, G_{a,j,B} \tag{502}$$

be candidate gains for arm a on agent j. The primary behavioral quantity is

$$M_a(B) = \frac{1}{n} \sum_{j=1}^{n} \max_{1 \le b \le B} G_{a,j,b}. \tag{503}$$

Comparing arms at unequal B confounds repair competence with order statistics.

20.4 Transport positive control

Proposition 20.1 (Exact transported canonical repair)**.** *For an invertible architecture map T, the runtime patch*

$$\delta = -Td \tag{504}$$

removes the canonical defect exactly:

$$T^{-1}(Td - Td) = 0. \tag{505}$$

Hence protected utility becomes one.

The full benchmark need not always offer the joint exact patch within budget. It does, however, construct individual typed patches in canonical coordinates and transport them through T.

20.5 Structural metric

For canonical patch $u = T^{-1}\delta$, define corrected protected energy

$$E_{\mathrm{corr}}(d, u) = \sum_{i=1}^{6} w_i \min\{d_i^2, u_i^2\}. \tag{506}$$

This quantity is zero for a no-op and positive when a patch reaches task-relevant defect coordinates. It is reported alongside behavioral gain because a high apparent score can be generated without correcting protected defect energy.

20.6 Reference benchmark execution

Arm	Mean protected gain	Median gain	Corrected energy	False accept rate
Static	0.000000	0.000000	0.000000	0.000000
Global edit	−0.324252	−0.325451	0.834602	0.958333
Archive only	0.474975	0.475579	0.744173	0.000000
Packet only	0.609294	0.625606	1.051647	0.000000
Causal only	0.044230	0.000000	0.071506	0.000000
Full SRIC	0.519120	0.580246	0.870198	0.000000
Full SRIC meta	0.640517	0.654932	1.085841	0.000000

Table 13: Matched-budget synthetic foreign-agent benchmark. The global-edit arm is an evaluator-capture positive control.

Several interpretations matter:

1. Packet-only performance is high because the toy defects are sparse and visible-coordinate correction is often effective. This prevents the benchmark from being designed so that only the full architecture can succeed.
2. Causal-only performance is low because correct causal type without coordinate transport is insufficient under architecture changes.
3. Archive-only search is a strong baseline and must not be omitted.

4. Global edit has high apparent gain but negative protected gain, with a 0.958333 false-accept rate. This confirms that the benchmark can expose evaluator capture.
5. Full SRIC has positive protected gain and zero false accepts in the released fixture, but it is not uniformly best. That is an informative outcome rather than a failure of the benchmark.
6. The engineered meta arm improves the held-out sixth defect type. Its novel-type gain exceeds the non-meta arm by 0.700533. This is a causal positive control for the abstraction metric, not evidence that a learned system discovered the schema.

20.7 Causal abstraction contrast

Let $N \subseteq \{1, \ldots, n\}$ be foreign agents whose dominant defect belongs to the held-out sixth type. Define

$$\widehat{\tau}_{\text{abstract}} = \frac{1}{|N|} \sum_{j \in N} \left(G_j^{\text{meta}} - G_j^{\text{no-meta}} \right). \tag{507}$$

In the reference positive control,

$$\widehat{\tau}_{\text{abstract}} = 0.700533. \tag{508}$$

A learned recursive-improvement claim would require the schema to be generated by the parent improvement process, sealed before foreign evaluation, and compared under matched computation and evidence access.

21 Causal Ablation and Statistical Inference

A benchmark can be mechanically correct while leaving the causal role of its components unresolved. SRIC therefore treats ablations as interventions on a factorized architecture rather than as informal feature removals.

21.1 Factorized architecture

Let the executable architecture have binary factors

$$Z = (Z_C, Z_P, Z_D, Z_T, Z_W, Z_A, Z_M), \tag{509}$$

representing causal attribution, packet localization, descent checking, transport, witness separation, archive retention, and repair abstraction.

For foreign agent j, let

$$Y_j(z) \tag{510}$$

be protected gain under factor setting z. The average necessity effect of factor k relative to the intact architecture $\mathbf{1}$ is

$$\tau_k = \mathbb{E}\left[Y_j(\mathbf{1}) - Y_j(\mathbf{1} - e_k) \right]. \tag{511}$$

A positive value supports causal usefulness only relative to the frozen environment, compensating mechanisms, and matched resource schedule.

21.2 Decisive schema-abstraction ablation

The primary recursive-effect ablation is

$$Z_M : 1 \to 0 \tag{512}$$

while preserving every immediate task-level repair produced before the abstraction point. The descendant is denied only the reusable schema object and its future retrieval path.

Definition 21.1 (Recursive abstraction estimand)**.** Let $\mathcal{F}_{\text{post}}$ be foreign defect families revealed after the source repair. Define

$$\tau_M^{\text{future}} = \mathbb{E}_{A \sim \mathcal{F}_{\text{post}}} \left[G_A(Z_M = 1) - G_A(Z_M = 0) \right]. \tag{513}$$

This estimand distinguishes a repair that helps one source task from a repair abstraction that improves later repair competence.

21.3 Paired estimation

For arms a and b, evaluate both on the same frozen foreign agents and seeds. Let

$$\Delta_j = G_{a,j} - G_{b,j}. \tag{514}$$

The paired estimate and standard error are

$$\widehat{\Delta} = \frac{1}{n} \sum_{j=1}^{n} \Delta_j, \qquad \widehat{\text{se}}(\widehat{\Delta}) = \sqrt{\frac{1}{n(n-1)} \sum_{j=1}^{n} (\Delta_j - \widehat{\Delta})^2}. \tag{515}$$

When gains are bounded in $[-1, 1]$, a distribution-free Hoeffding lower bound is

$$\text{LCB}_{1-\alpha}(\Delta) = \widehat{\Delta} - \sqrt{\frac{2 \log(1/\alpha)}{n}}. \tag{516}$$

Sharper paired permutation or bootstrap intervals may be used when frozen in advance.

21.4 Multiple claims and alpha spending

Let $H_1, \ldots, H_m$ be ordered primary hypotheses. Choose nonnegative levels α_i satisfying

$$\sum_{i=1}^{m} \alpha_i \leq \alpha. \tag{517}$$

Each endpoint is tested only with its assigned held-out evidence. Reusing the same audit sample to tune thresholds and report final confidence invalidates the nominal bound.

For sequential candidate generation, an anytime-valid confidence sequence or e-value process is preferable. The benchmark receipt must record every look at protected evidence, including apparently unsuccessful candidates.

21.5 Required negative controls

Control	Purpose	Expected outcome
No-op candidate	Detects baseline drift and evaluator noise	zero protected gain in expectation
Shuffled repair-schema labels	Tests whether semantic type is used	transfer should collapse
Random same-rank transport	Controls for dimension and norm	worse than typed transport
Wrong-lineage receipt	Tests provenance authorization	rejected or quarantined
Evaluator-capture candidate	Tests protected-score separation	high apparent score, rejected protected gain
Rollback ablation	Tests necessity of restoration	universal closure violation
Witness-edit attempt	Tests transaction separation	rejected before commit
Schema-abstraction removal	Tests recursive reuse	future novel-type gain decreases
Candidate-budget equalization	Removes best-of-k confound	all arms use same B
Foreign-agent permutation	Tests lineage memorization	gains persist only if repair abstraction transfers

21.6 Primary Block 5 gates

The released reference package uses the following gates:

1. all nine finite-fixture endpoints pass;
2. all six transaction-kernel endpoints pass;
3. anonymous non-none attribution accuracy is at least 0.90;
4. anonymous repair-or-abstain success is at least 0.90;
5. false repair under no change is at most 0.10;
6. full SRIC protected gain exceeds static and global-edit gain under the matched budget;
7. the evaluator-capture positive control produces at least one false apparent acceptance;
8. full SRIC produces zero false protected acceptance in the released fixture;
9. the meta positive control improves the held-out defect-type contrast;
10. package verification and all automated tests pass.

The gates were selected to verify implementation mechanics and positive controls. They are not thresholds for a general scientific claim of RSI.

22 Reproducibility, Packaging, and Verification

The empirical block is distributed as source code, tests, result objects, manifest, and a complete manuscript bundle.

22.1 Package layout

```
sric_block05_reference/
  README.md
  pyproject.toml
  Makefile
  Run_SRIC_Block05.ps1
  run_all.py
  verify_package.py
  MANIFEST.sha256
  sric_ref/
    __init__.py
    canonical.py
    linalg.py
    receipt.py
    transaction.py
    fixture.py
    anonymous_env.py
    benchmark.py
  tests/
    conftest.py
    test_fixture.py
    test_transaction.py
    test_anonymous.py
    test_benchmark.py
  results/
    fixture_results.json
    transaction_results.json
    anonymous_results.json
    benchmark_results.json
    verification_summary.json
```

22.2 Deterministic and stochastic replay

The finite fixture and transaction suite are deterministic. The anonymous and foreign-agent suites are pseudorandom with frozen seeds recorded in each result object. Two levels of replay are distinguished:

1. **exact replay**: canonical result bytes and digests agree;
2. **statistical replay**: a new independent seed satisfies the preregistered endpoint gates.

Exact replay detects package or implementation drift. Statistical replay tests whether a result is robust beyond one frozen sample.

Definition 22.1 (Package-closed claim)**.** A claim is package closed when every number required to evaluate its gate can be regenerated from released source, declared dependencies, frozen configuration, and released evidence without hidden network services or mutable external state.

Proposition 22.2 (Manifest integrity)**.** *Assuming collision resistance of SHA-256, if every declared package file matches its manifest digest, the verifier establishes byte integrity of those files relative to the manifest.*

The proposition does not prove that the manifest is complete. The release builder must separately compare the manifest path set with the intended package path set.

22.3 Verification commands

The canonical local commands are:

```
python run_all.py
pytest -q
python verify_package.py
```

The first writes all result JSON files. The second checks unit tests. The third checks the SHA-256 manifest after results have been frozen.

22.4 Reference verification summary

The released execution reported:

```
SRIC_BLOCK05_ALL_SUITES=PASS
fixture_endpoints=9/9
transaction_endpoints=6/6
anonymous_accuracy=0.950000
anonymous_false_repair=0.037500
full_sric_gain=0.519120
full_sric_meta_gain=0.640517
```

All four automated tests passed. The package verifier is run only after the manifest is generated from the final frozen tree.

22.5 Reproducibility tiers

Tier	Requirement	What it supports
R0	Narrative description only	conceptual inspection
R1	Source executes	mechanical availability
R2	Frozen tests and results regenerate	deterministic implementation claim

Tier	Requirement	What it supports
R3	Clean environment and independent seed gates pass	robustness of synthetic fixture
R4	Independent implementation refines the same abstract model	reduced implementation-specific risk
R5	Foreign learned systems reproduce causal and metaproductive endpoints	substantive empirical SRIC evidence

The Block 5 reference package targets R2 and supplies the protocol needed to attempt R3. It does not claim R4 or R5.

Block 5 result

The empirical layer is now executable. The finite correctability, descent, holonomy, closure, transaction, anonymous-change, foreign-agent, and evaluator-capture controls are represented in one reproducible reference package. The package verifies mechanics and engineered synthetic positive controls. Learned repair discovery, foreign-model transfer, autonomous grammar expansion, and open-ended recursive improvement remain unestablished.

23 Learned-Agent Validation Contract

Block 5 established an executable synthetic reference layer with analytic defect coordinates and hand-coded positive controls. Block 6 removes one important simplification. The repair is now discovered in a hidden representation, the source-to-target coordinate map is withheld from the repair system, and transfer is evaluated on target agents whose representations were not used to construct the source repair.

The block remains controlled. The task family, causal probe, candidate repair form, witness policy, and evaluation split are designer supplied. The result therefore tests *learned hidden-state repair transport*, not open-ended self-reconstruction.

23.1 Representation-level validation object

Definition 23.1 (Learned repair-transport instance)**.** A learned repair-transport instance is a tuple

$$\mathfrak{R}_{\text{learn}} = (\mathcal{X}, \phi, q, \widetilde{q}, A_s, \mathcal{A}_t, \mathsf{D}, \mathsf{P}, \mathsf{G}, \mathsf{W}), \tag{518}$$

where:

- $\mathcal{X}$ is the input space;
- ϕ is the task feature map;
- q and $\widetilde{q}$ are protected and corrupted score maps;
- A_s is the source agent;
- $\mathcal{A}_t$ is a family of target agents excluded from source repair fitting;

- D is the train, anchor, calibration, and audit partition;
- P is the permitted causal probe language;
- G is the repair and transport grammar;
- W is the frozen witness policy.

The source repair may use source training and anchor data. A target transport may use paired anchor activations from the source and that target. It may not use target audit labels, the hidden target transformation, or the target-local oracle repair. The target-local repair is retained only as a post hoc upper comparison.

Definition 23.2 (Foreign representation target)**.** A target agent is foreign relative to a source repair when its encoder parameters or construction seed are absent from source packet discovery and source repair fitting. A stronger architecture-foreign target additionally belongs to an encoder family absent from the source lineage.

This is weaker than cross-task or cross-environment foreignness. Block 6 evaluates foreign representation systems inside one shared task and defect grammar.

23.2 Data separation

The controlled suite uses four disjoint data sets:

$$\mathsf{D} = \mathsf{D}_{\text{train}} \sqcup \mathsf{D}_{\text{anchor}} \sqcup \mathsf{D}_{\text{calibrate}} \sqcup \mathsf{D}_{\text{audit}}. \tag{519}$$

The source and target heads are fitted on $\mathsf{D}_{\text{train}}$. Packet discovery and source repair fitting use $\mathsf{D}_{\text{anchor}}$. The witness sees only $\mathsf{D}_{\text{calibrate}}$. Reported final accuracies use $\mathsf{D}_{\text{audit}}$.

For the controlled transport experiment, the sample sizes are

$$(n_{\text{train}}, n_{\text{anchor}}, n_{\text{calibrate}}, n_{\text{audit}}) = (7000, 1500, 1600, 6000). \tag{520}$$

The learned-representation pilot uses

$$(5000, 1200, 1200, 4000). \tag{521}$$

23.3 Block 6 claim target

The highest admissible target for this block is empirical claim-ladder level 4:

a repair schema learned in one hidden representation improves held-out foreign representation systems after a map is learned from permitted paired anchors.

The block does not compare a descendant repair calculus against its parent. It therefore does not establish level-5 metaproductivity. It also does not construct a new repair type, change the evaluator, or expand the grammar.

Block 6 empirical principle

A successful transported repair must do more than raise target accuracy. The causal packet must outperform same-rank noncausal controls, paired-anchor alignment must be necessary, the repair must preserve the unaffected stratum, and the frozen witness must abstain when transport quality or protected behavior falls outside the declared envelope.

24 Controlled Causal-Packet Fixture

24.1 Task and defect construction

Let $x = (x_1, x_2, x_3, x_4) \in \mathbb{R}^4$. Define the eight-dimensional base feature

$$\psi(x) = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_1x_2 & x_3x_4 & \sin x_1 & \cos x_2 \end{bmatrix}^\top \in \mathbb{R}^8, \tag{522}$$

and the affected-stratum indicator

$$s(x) = \mathbf{1}\{x_1 + x_2 > 0.8\}. \tag{523}$$

The full feature map is

$$\phi(x) = \begin{bmatrix} \psi(x) \\ s(x)\psi(x) \end{bmatrix} \in \mathbb{R}^{16}. \tag{524}$$

Let

$$q(x) = W\psi(x) + b \in \mathbb{R}^3 \tag{525}$$

be the protected score. Let C be the cyclic score permutation

$$C(y_1, y_2, y_3)^\top = (y_3, y_1, y_2)^\top. \tag{526}$$

The corrupted score is

$$\widetilde{q}(x) = (1 - s(x))q(x) + s(x)Cq(x). \tag{527}$$

Thus the model is almost perfectly correct outside the affected stratum and systematically wrong inside it.

An agent a has encoder

$$E_a : \mathbb{R}^{16} \to \mathbb{R}^{d_a} \tag{528}$$

and an affine head fitted to $\widetilde{q}$. Its hidden state is

$$h_a(x) = E_a(\phi(x)). \tag{529}$$

The transport system is not given the matrix or nonlinear parameters of E_a.

24.2 Causal counterfactual probe

The permitted intervention removes only the conditional feature block:

$$\phi^\circ(x) = \begin{bmatrix} \psi(x) \\ 0 \end{bmatrix}. \tag{530}$$

The resulting hidden difference is

$$\Delta_a(x) = h_a(x) - h_a^\circ(x), \qquad h_a^\circ(x) = E_a(\phi^\circ(x)). \tag{531}$$

The empirical causal packet is the leading rank-r right-singular subspace of the affected differences

$$\{\Delta_a(x_i) : s(x_i) = 1\}. \tag{532}$$

Theorem 24.1 (Exact causal-packet recovery for a linear encoder)**.** *Suppose*

$$E_a = \begin{bmatrix} A_a & B_a \end{bmatrix}, \qquad A_a, B_a : \mathbb{R}^8 \to \mathbb{R}^{d_a}, \tag{533}$$

and suppose the centered conditional covariance of $\psi(x)$ given $s(x) = 1$ is positive definite. Then

$$\operatorname{span}\{\Delta_a(x) : s(x) = 1\} = im B_a. \tag{534}$$

Consequently, a rank-r singular-subspace estimator with $r = \operatorname{rank} B_a$ recovers the exact causal packet from a sufficiently rich noiseless sample.

Proof. From Equations (524) and (530),

$$\Delta_a(x) = s(x) B_a \psi(x). \tag{535}$$

Hence every affected difference lies in $\operatorname{im} B_a$. Conversely, positive-definite conditional covariance implies that the centered affected values of ψ span $\mathbb{R}^8$. Their images therefore span $\operatorname{im} B_a$. The singular subspace associated with the nonzero singular values is exactly that image. □

Remark 24.2 (Why ordinary PCA need not recover the repair packet)**.** PCA maximizes variance of $h_a(x)$, not sensitivity to the declared intervention. Large variance in $A_a\psi(x)$ can dominate the spectrum while the entire defect mechanism lies in $B_a\psi(x)$. Equal rank does not make the two subspaces semantically equivalent.

Example 24.3 (Same-rank PCA failure)**.** Let A_a amplify eight high-variance task coordinates and let B_a encode eight lower-variance conditional coordinates. Rank-eight PCA selects $\operatorname{im} A_a$, while the causal difference span is $\operatorname{im} B_a$. A repair restricted to the PCA packet can preserve overall variance while leaving the defect nearly unchanged.

24.3 Packet-local repair fitting

Let $U_a \in \mathbb{R}^{d_a \times r}$ have orthonormal columns spanning the discovered packet. Define desired packet coordinates

$$z_i = U_a^\top (h_a^\circ(x_i) - h_a(x_i)). \tag{536}$$

Fit an affine predictor

$$\widehat{z}(h) = K_a h + c_a \tag{537}$$

by ridge regression on source anchor states. The repair is

$$R_a(h) = h + U_a \widehat{z}(h) = h + D_a h + b_a, \tag{538}$$

where

$$D_a = U_a K_a, \qquad b_a = U_a c_a. \tag{539}$$

Proposition 24.4 (Exact packet repair under affine predictability)**.** *Suppose every desired correction lies in* $\operatorname{im} U_a$ *and there are* K_a, c_a *such that*

$$U_a^\top (h_a^\circ - h_a) = K_a h_a + c_a \tag{540}$$

throughout the evaluated support. Then Equation (538) satisfies

$$R_a(h_a) = h_a^\circ. \tag{541}$$

Proof. Because the desired correction is in $\operatorname{im} U_a$,

$$h_a^\circ - h_a = U_a U_a^\top (h_a^\circ - h_a) = U_a(K_a h_a + c_a). \tag{542}$$

Substitution into Equation (538) gives the claim. □

24.4 Packet-specificity result

The source head attains 0.9985 accuracy against the corrupted labels. Against the protected labels, its overall accuracy is 0.7075. The unaffected stratum remains nearly perfect, while the affected stratum collapses to 0.002277.

Table 16: Source repair under three rank-eight packet estimators. The uncorrected affected-stratum accuracy is 0.002277.

Packet estimator	Rank	Affected slice	Unaffected slice	Overall
Causal difference	8	0.994308	0.999529	0.998000
Same-rank PCA	8	0.023335	0.999529	0.713667
Random subspace	8	0.570290	0.999529	0.873833

The causal packet restores 99.43% affected-stratum accuracy and leaves the unaffected stratum unchanged. Same-rank PCA reaches only 2.33%. The random packet produces a partial repair, which is expected because a random rank-eight subspace in a sixteen-dimensional state has nonzero overlap with the true rank-eight packet. It nevertheless remains far below the causal packet.

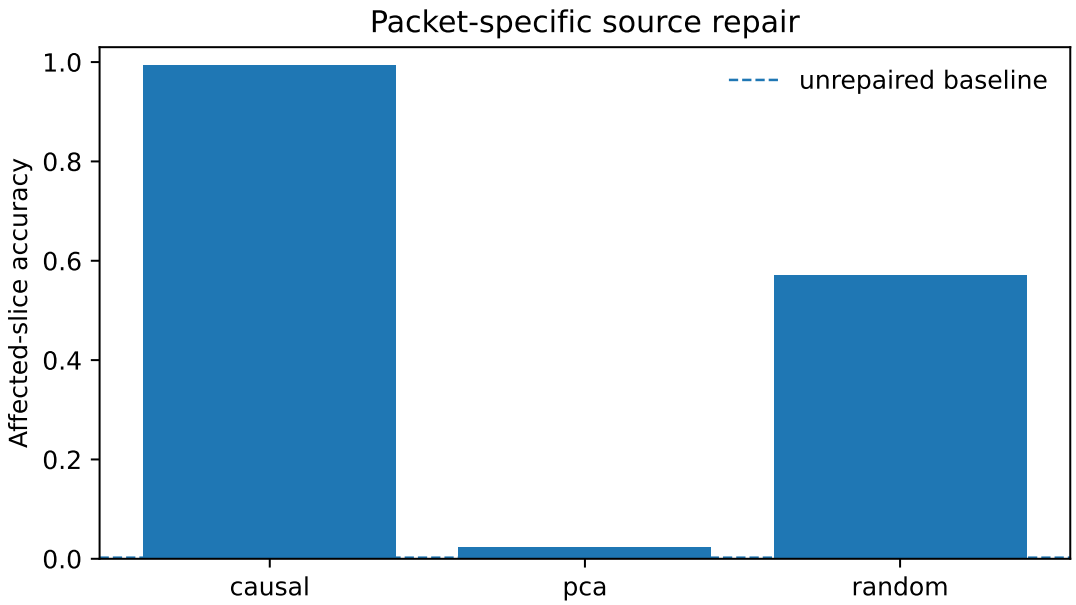


Figure 2: Affected-stratum accuracy after fitting the same repair form in causal, PCA, and random rank-eight packets. Packet rank alone does not explain the result.

24.5 Wrong-block control

A matched control zeros the base feature block instead of the conditional block. The resulting packet and repair leave affected-stratum performance unchanged in the tested linear, overcomplete, and mildly nonlinear families.

Proposition 24.5 (Wrong-block null under block-separable score correction)**.** *Suppose the protected score is determined by the base block and the corruption is represented only through the conditional block. A repair learned from a counterfactual that removes the base block does not identify the conditional correction unless the encoder mixes the two mechanisms in an additional identifiable way.*

The proposition is intentionally conditional. The executable wrong-block result is the relevant finite negative control.

25 Learned Cross-Agent Repair Transport

25.1 Learning the hidden-state bridge

Let s denote the source and t a target. On paired anchor inputs $x_1, \dots, x_m$, the system observes

$$H_s = \begin{bmatrix} h_s(x_1)^\top \\ \vdots \\ h_s(x_m)^\top \end{bmatrix}, \qquad H_t = \begin{bmatrix} h_t(x_1)^\top \\ \vdots \\ h_t(x_m)^\top \end{bmatrix}. \tag{543}$$

It estimates the bridge by regularized least squares:

$$\widehat{T}_{t\leftarrow s} \in \arg\min_{T} \left\| H_s T^\top - H_t \right\|_F^2 + \lambda \|T\|_F^2 . \tag{544}$$

The transformation matrices used to construct the controlled targets are not exposed to this fit.

Given source repair

$$R_s(h) = h + D_s h + b_s, \tag{545}$$

its transported target form is

$$D_t = \widehat{T} D_s \widehat{T}^+, \qquad b_t = \widehat{T} b_s, \tag{546}$$

with transported packet basis obtained by orthogonalizing $\widehat{T} U_s$.

25.2 Exact injective transport

Theorem 25.1 (Exact transport through an injective hidden bridge)**.** *Let $T : \mathbb{R}^{d_s} \to \mathbb{R}^{d_t}$ have full column rank, so $T^+T = I_{d_s}$. Suppose*

$$h_t = T h_s \tag{547}$$

throughout the evaluated support. Define D_t and b_t by Equation (546) with $\widehat{T} = T$. Then

$$R_t(T h_s) = T R_s(h_s). \tag{548}$$

Proof. Using $T^+T = I$,

$$\begin{aligned} R_t(T h_s) &= T h_s + T D_s T^+ T h_s + T b_s && (549)\\ &= T(h_s + D_s h_s + b_s) && (550)\\ &= T R_s(h_s). && (551) \end{aligned}$$

□

Corollary 25.2 (Overcomplete targets)**.** *An overcomplete target with $d_t > d_s$ admits exact transported repair whenever its bridge from the source is injective and the target states remain in the bridge image.*

This explains why the 24-dimensional overcomplete family matches the local target oracle despite having a different state dimension.

25.3 Compression obstruction

Proposition 25.3 (Residual caused by a noninjective bridge)**.** *Let $h_t = T h_s$ with arbitrary T. Using Equation (546),*

$$R_t(T h_s) - T R_s(h_s) = T D_s (T^+ T - I) h_s. \tag{552}$$

Hence exact state alignment $h_t = T h_s$ does not imply repair equivariance when the bridge discards a source direction used by D_s.

Proof. The bias terms cancel. Expanding the linear terms gives

$$T D_s T^+ T h_s - T D_s h_s = T D_s (T^+ T - I) h_s. \tag{553}$$

□

This distinction is visible in the compressed family. Its alignment residual is approximately 3.3×10^{-8}, yet its repair-equivariance residual is 0.4715. Blind transfer raises affected accuracy from 0.1277 to 0.6204, but it damages the protected stratum and is rejected by the witness.

25.4 Approximate transport decomposition

Let

$$e_h = h_t - \widehat{T} h_s, \qquad E_D = D_t \widehat{T} - \widehat{T} D_s, \qquad E_b = b_t - \widehat{T} b_s. \tag{554}$$

Then

$$R_t(h_t) - \widehat{T} R_s(h_s) = (I + D_t) e_h + E_D h_s + E_b. \tag{555}$$

Therefore

$$\left\| R_t(h_t) - \widehat{T} R_s(h_s) \right\| \le (1 + \|D_t\|) \|e_h\| + \|E_D\| \|h_s\| + \|E_b\|. \tag{556}$$

The implementation reports normalized alignment and repair-equivariance residuals as empirical summaries of these terms.

25.5 Anchor identifiability

Theorem 25.4 (Exact least-squares bridge from spanning anchors)**.** *Suppose $H_t = H_s T^\top$ exactly and H_s has full column rank. With zero regularization, the least-squares solution of Equation (544) is unique and equals T.*

Proof. The normal equations are

$$H_s^\top H_s \widehat{T}^\top = H_s^\top H_t. \tag{557}$$

Full column rank makes $H_s^\top H_s$ invertible. Substitution of $H_t = H_s T^\top$ yields $\widehat{T} = T$. □

The theorem is an algebraic identifiability statement, not a sample-complexity theorem. Conditioning and nonlinear mismatch determine how many random anchors are practically needed.

26 Foreign Representation Families

26.1 Controlled target atlas

The controlled suite evaluates six seeds in each of seven target families:

1. `linear16`: invertible orthogonal linear coordinates;
2. `overcomplete24`: a 24-dimensional injective embedding;
3. `ill_conditioned16`: invertible linear coordinates with singular values from 1 to 0.05;
4. `compressed12`: a noninjective 12-dimensional projection;
5. `mild_tanh24`: sixteen mildly nonlinear coordinates plus an eight-dimensional linear skip;
6. `saturated_tanh24`: strongly saturated nonlinear coordinates;
7. `relu32`: a piecewise-linear ReLU representation.

Each target head is fitted independently to the corrupted score function. The source repair and source packet remain frozen.

26.2 Control arms

For every target, the package records:

- the unrepaired baseline;
- blind transported source repair;
- witness-selected behavior;
- a target-local causal repair oracle;
- shuffled-anchor transport;
- direct untransported application when dimensions match;
- alignment and repair-equivariance residuals.

The direct untransported control is important. A source and target can have equal width while assigning different semantics to the same coordinate index. Equal dimensionality is not an authorization to copy a repair matrix directly.

26.3 Family results

Table 17: Controlled foreign-representation results, averaged over six seeds. "Blind" applies the transported repair before witness selection. "Selected" applies it only when the frozen gate passes.

Family	Base slice	Blind	Selected	Local oracle	Align NRMSE	Equiv. NRMSE	Commit
linear16	.0023	.9943	.9943	.9943	3.3×10^{-8}	5.9×10^{-8}	1.00
overcomplete24	.0023	.9943	.9943	.9943	3.3×10^{-8}	5.9×10^{-8}	1.00
ill-conditioned16	.0023	.9943	.9943	.9943	1.9×10^{-8}	4.6×10^{-8}	1.00
compressed12	.1277	.6204	.1277	.6648	3.3×10^{-8}	.4715	0.00
mild-tanh24	.0949	.7463	.5645	.6112	.2029	.3266	.667
saturated-tanh24	.2683	.6072	.2683	.5169	.4984	.7613	0.00
relu32	.1332	.6868	.1332	.4478	.3741	.3546	0.00

Three conclusions are distinct.

First, exact and injective linear transport behaves as predicted by Theorem 25.1. The transported repair equals the local oracle to numerical precision for the linear, overcomplete, and ill-conditioned families.

Second, blind target improvement is not sufficient. Compressed, saturated, and ReLU targets all show substantial affected-slice gains before selection. The witness rejects them because one or more protected conditions fail.

Third, mild nonlinear representations occupy an intermediate regime. Blind transport reaches 0.7463 affected accuracy. Four of six seeds pass the witness, producing a selected mean of 0.5645. The result is useful but not representation invariant.

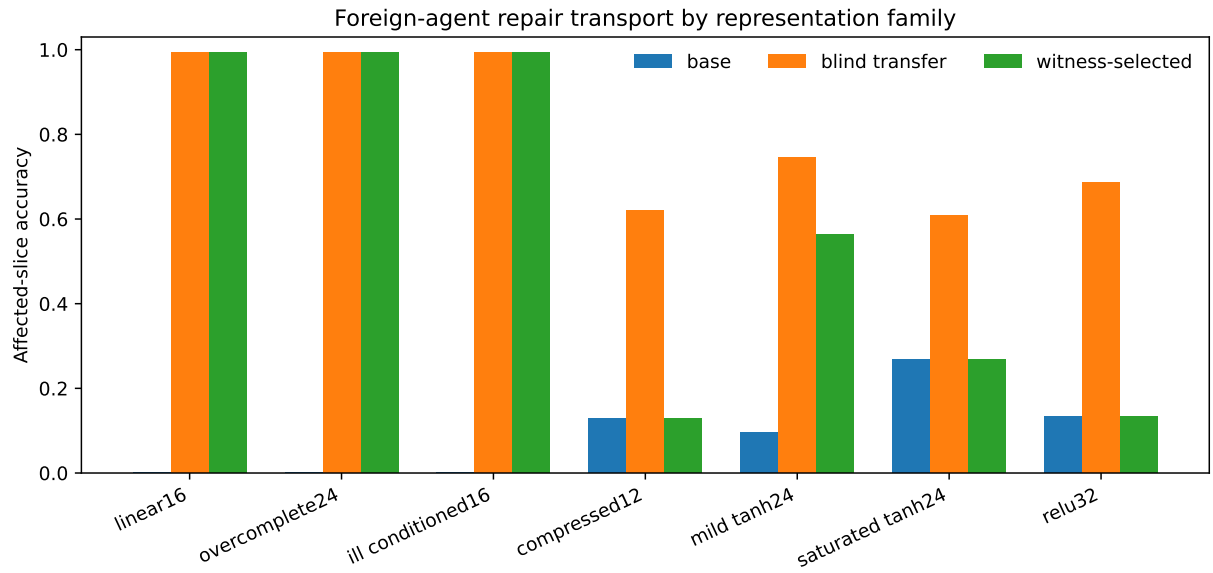


Figure 3: Unrepaired, blind-transport, and witness-selected affected-stratum accuracy. The gap between blind and selected performance is intentional. It shows that the witness does not equate apparent target gain with an authorized commit.

26.4 Shuffled-anchor and direct-copy controls

For the exact 16-dimensional family, the mean direct untransported source repair reaches only 0.2646 affected accuracy. The mean shuffled-anchor transport reaches 0.2442. Correct learned transport reaches 0.9943.

Proposition 26.1 (Anchor correspondence is causally necessary in the exact fixture)**.** *Within the frozen exact-linear fixture, replacing paired target anchors by an independent permutation changes only the source-target correspondence used to estimate T. The resulting drop from* 0.9943 *to* 0.2442 *therefore identifies correspondence-sensitive transport rather than a generic benefit from applying the source repair.*

The proposition is finite and fixture specific. It does not show that every possible unpaired alignment method must fail.

26.5 Anchor-budget sweep

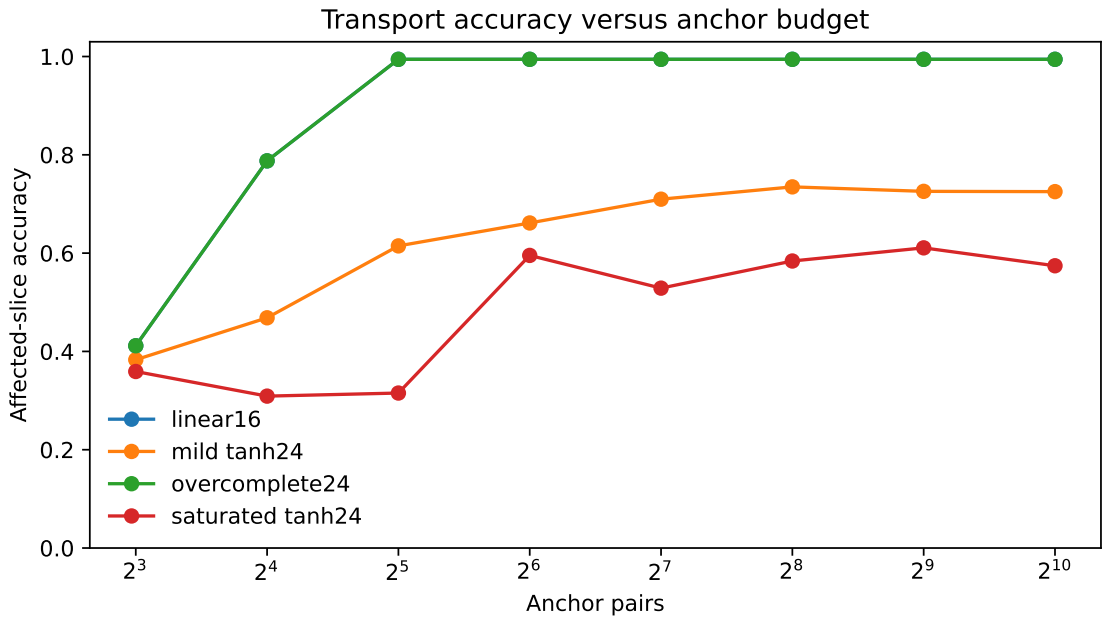


Figure 4: Affected-stratum transport accuracy as the paired-anchor budget increases. Exact linear families become numerically exact once the anchor matrix is sufficiently spanning and conditioned. Mild nonlinear transfer improves gradually. Saturated nonlinear transfer remains outside the witness envelope.

For linear and overcomplete targets, eight anchors produce 0.4115 affected accuracy and fail the gate. Sixteen anchors reach 0.7877 and pass. Thirty-two anchors recover 0.9943, after which additional anchors do not change the audit result. Mild nonlinear transfer improves from 0.3830 at eight anchors to approximately 0.73 at 256 or more. Saturated nonlinear transfer remains unstable and is rejected at every tested budget.

This is evidence for an identifiability and approximation transition, not a universal power law. The exact threshold depends on source rank, conditioning, anchor distribution, and regularization.

27 Witness-Gated Repair Commitment

27.1 Frozen gate

The witness commits a transported repair only when all four calibration conditions hold:

$$\widehat{g}_{\mathrm{slice}} \geq 0.20, \tag{558}$$

$$\widehat{r}_{\mathrm{non}} \leq 0.03, \tag{559}$$

$$\widehat{\varepsilon}_{\mathrm{align}} \leq 0.35, \tag{560}$$

$$\widehat{A}^{\mathrm{base}}_{\mathrm{non}} \geq 0.90. \tag{561}$$

Here $\widehat{g}_{\text{slice}}$ is calibration gain on the affected stratum, $\widehat{r}_{\text{non}}$ is regression on the unaffected stratum, $\widehat{\varepsilon}_{\text{align}}$ is normalized alignment error, and $\widehat{A}_{\text{non}}^{\text{base}}$ is baseline unaffected accuracy.

The last condition prevents an already degraded target from being treated as safe merely because the candidate does not make it much worse.

Conditional Theorem 27.1 (Calibration-to-audit witness soundness)**.** *Suppose, simultaneously, that the audit gain and regression satisfy*

$$g_{\text{slice}} \geq \widehat{g}_{\text{slice}} - \eta_g, \tag{562}$$

$$r_{\text{non}} \leq \widehat{r}_{\text{non}} + \eta_r, \tag{563}$$

and that the reported alignment error upper-bounds the audit alignment error up to η_a. If

$$\widehat{g}_{\text{slice}} - \eta_g > 0, \tag{564}$$

$$\widehat{r}_{\text{non}} + \eta_r \leq r_{\max}, \tag{565}$$

$$\widehat{\varepsilon}_{\text{align}} + \eta_a \leq \varepsilon_{\max}, \tag{566}$$

then a candidate passing the corresponding tightened calibration gate has positive audit gain and satisfies the declared audit regression and alignment limits.

Proof. Substitute the simultaneous deviation bounds into the three tightened inequalities. □

The executable Block 6 gate uses fixed thresholds without an anytime-valid confidence sequence. The theorem states what a statistically certified deployment would additionally require.

27.2 Why the gate rejects apparently useful repairs

The compressed target has near-zero alignment error but poor base unaffected accuracy and excessive protected regression. The saturated target has large alignment and equivariance error. The ReLU family narrowly exceeds the alignment threshold and also begins below the protected base-accuracy floor. All three are rejected.

This behavior is central to SRIC. A witness is not a benchmark-score maximizer. It enforces the declared preservation profile.

Warning 27.2 (Gate success is relative)**.** The frozen gate cannot detect an omitted invariant. Passing it shows only that the candidate satisfies the measured calibration conditions and the package's finite tests.

28 Independently Trained Representation Pilot

The controlled families above include learned heads but mostly constructed encoders. A second suite trains the representation itself from independent random initializations.

28.1 Training objective

For encoder E_θ, decoder G_δ, and score head H_ω, the pilot minimizes minibatch estimates of

$$\mathcal{L}(\theta, \delta, \omega) = \lambda_{\text{rec}} \left\| G_\delta(E_\theta(\phi)) - \phi \right\|_F^2 + \left\| H_\omega(E_\theta(\phi)) - \widetilde{q} \right\|_F^2, \tag{567}$$

with $\lambda_{\mathrm{rec}} = 2$. Optimization uses a deterministic NumPy Adam implementation for 500 steps. After representation training, the head is refitted by ridge regression to reduce avoidable readout variance.

The source is a learned 16-dimensional linear representation. Targets are independently trained:

- three learned linear targets of width 16;
- three learned linear targets of width 20;
- three learned tanh targets of width 20.

The source repair is discovered once and then frozen.

28.2 Pilot results

Table 18: Independently trained representation pilot, averaged over three target seeds.

Family	Base slice	Transport	Local oracle	Non-slice	Align NRMSE	Commit
learned linear16	.00339	.99323	.99323	.99965	6.6×10^{-8}	1.00
learned linear20	.00339	.99323	.99323	.99965	7.6×10^{-8}	1.00
learned tanh20	.01948	.89416	.89670	.96003	.14706	1.00

The learned linear targets match the local oracle. The learned tanh targets reach 0.8942 affected accuracy versus a target-local oracle of 0.8967. Their mean unaffected accuracy after transport is 0.9600, and all three pass the frozen witness.

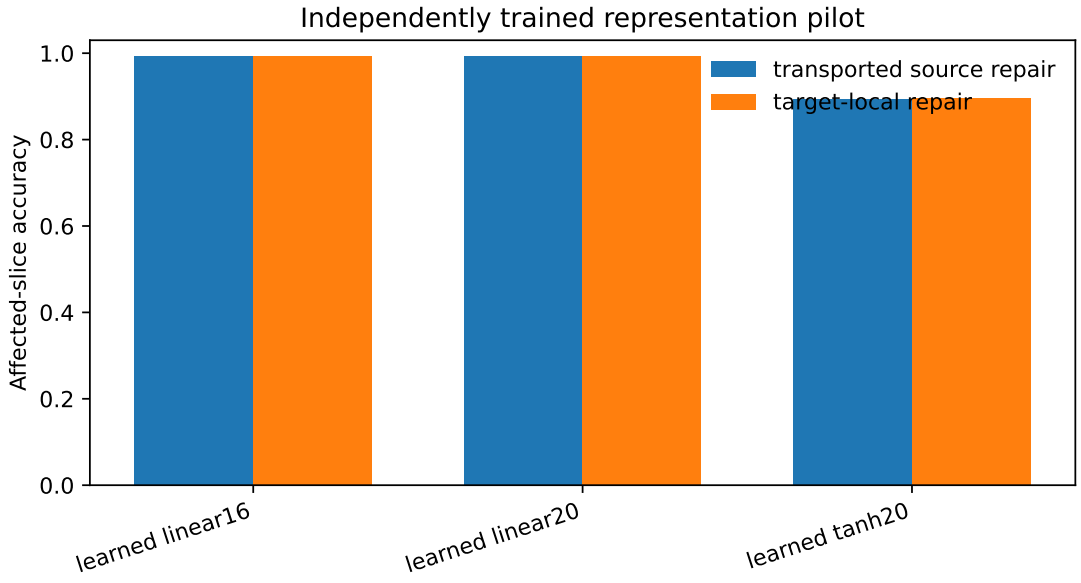


Figure 5: Transported source repair compared with a target-local causal repair in independently trained representations. The nonlinear learned targets show a small but measurable transport gap.

28.3 Interpretation

The pilot shows that exact parameter copying is unnecessary. Independently trained representations can support repair transport when paired anchors identify a sufficiently coherent bridge.

It does not show that the agents discovered the task decomposition, causal probe, repair grammar, or evaluator. The reconstruction term in Equation (567) actively encourages a linearly alignable task representation. The correct interpretation is therefore:

learned representations can instantiate the transport theorem under a controlled shared bottleneck objective.

29 Causal Controls and Statistical Interpretation

29.1 Factorized causal evidence

The block contains several interventions with different inferential roles.

Control	Changed factor	Interpretation
Causal vs. PCA packet	Intervention sensitivity at fixed packet rank	Tests whether causal localization, rather than low rank alone, identifies the repair subspace
Random packet	Packet orientation	Estimates gain obtainable from accidental overlap
Shuffled anchors	Source-target correspondence	Tests whether learned alignment uses paired semantics
Direct untransported repair	Coordinate transport	Tests whether equal width permits raw repair copying
Wrong feature block	Causal intervention target	Tests specificity to the actual defect-generating block
Target-local oracle	Transfer gap	Separates bridge error from target repair-form limitation
Witness ablation through blind score	Authorization	Shows why target gain alone is not a commit rule
Compressed target	Injectivity	Tests obstruction caused by information loss
Saturated and ReLU targets	Global linearizability	Tests the boundary of one global linear bridge

No single control establishes the whole mechanism. Their conjunction supports the narrower statement that the source repair acts through the intervention-defined packet and that correct paired transport is necessary for its strongest foreign-target effect.

29.2 Bootstrap intervals

Family means are accompanied by percentile bootstrap intervals across target seeds. With only six controlled seeds and three learned-pilot seeds, these intervals are descriptive rather than definitive.

Exact constructed families have nearly degenerate intervals because their algebraic behavior is seed invariant in the fixture.

A stronger follow-up should freeze a larger seed population before examining outcomes, use paired intervals for transported-versus-local performance, and report the joint error budget across packet, transport, and witness claims.

29.3 Claim-gate ledger

Table 20: Block 6 claim-gate status. "Pass" is restricted to the declared controlled benchmark.

Gate	Status	Evidence	Boundary
G_{pkg}	Pass	34-file SHA-256 manifest and clean verifier	Byte integrity relative to the released manifest
G_{mech}	Pass	Deterministic execution, five tests, frozen JSON and figures	Runtime mechanics only
G_{sem}	Pass	Declared task, repair, transport, and gate objects reproduced	Controlled finite semantics
G_{cause}	Pass	Causal packet, PCA, random, wrong-block, and shuffled-anchor controls	Causal role inside the fixture
G_{foreign}	Pass	Held-out target seeds, dimensions, encoder families, and learned targets	One shared task and defect grammar
G_{meta}	Not tested	No parent-child repair-calculus comparison	No metaproductivity claim
G_{open}	Not tested	No autonomous grammar or evaluator expansion	No open-set recursion claim

The strongest supported claim is therefore controlled foreign repair transfer. The result is a real step beyond the analytic Block 5 fixture, but it is not yet recursive improvement of the improvement process.

30 Block 6 Reproducibility and Package

30.1 Reference package

The Block 6 reference package [6] contains:

```
SRIC_v0.7_Block06_Learned_Agent_Validation/
|-- sric_learned/
|   |-- agents.py
|   |-- benchmark.py
|   |-- data.py
|   |-- learned_pilot.py
|   |-- repair.py
|   |-- transport.py
|   `-- plots.py
```

```
|-- tests/
|   |-- test_gate.py
|   |-- test_learned.py
|   |-- test_packet.py
|   `-- test_transport.py
|-- results/
|-- figures/
|-- run_all.py
|-- Run_SRIC_Block06.ps1
|-- build_manifest.py
|-- verify_package.py
`-- MANIFEST.sha256
```

The Python dependency surface is NumPy and Matplotlib. No GPU or network access is required.

30.2 Verification commands

```
python run_all.py
python -m pytest -q
python build_manifest.py
python verify_package.py
```

The frozen execution reports:

```
SRIC_BLOCK06_ALL_SUITES=PASS
result_sha256=4a210438c56d0e87cfcfa3524ec633a4d78cd8f966e41da8ee669eb544b72065
causal_packet_slice=0.994308
pca_packet_slice=0.023335
linear_transport_slice=0.994308
mild_nonlinear_transport_slice=0.746253
saturated_commit_rate=0.000000
learned_tanh_transport_slice=0.894157
```

All five automated tests pass. The manifest covers 34 package files.

30.3 Frozen endpoint set

The executable tests require:

1. causal packet affected accuracy above 0.98;
2. a causal-over-PCA affected-accuracy gap above 0.90;
3. causal repair unaffected accuracy above 0.99;
4. exact linear, overcomplete, and ill-conditioned transport above 0.98 and equal to the local oracle within numerical tolerance;
5. shuffled-anchor accuracy below 0.40 for exact linear targets;

6. positive witness acceptance in the mild nonlinear family;
7. zero witness acceptance in saturated tanh and ReLU families;
8. learned linear transport above 0.98;
9. learned tanh transport above 0.80.

These endpoints are frozen in the released tests. They were not prospectively preregistered before all internal pilot work. The manuscript therefore describes them as release-frozen verification gates, not as an independent confirmatory study.

Block 6 result

A causally discovered source repair can be transported through a map learned from paired hidden activations to foreign target representations. Exact injective families match target-local repair. Mild nonlinear and independently trained tanh representations retain substantial transfer. Compression and strong nonlinear mismatch expose measurable obstructions, and the frozen witness rejects several blind improvements. This establishes controlled foreign representation transfer, not repair-process metaproductivity or open-set recursive self-improvement.

31 Sealed Parent–Child Repair-Process Improvement

Block 6 established that a frozen repair can be discovered in one hidden representation, transported through a learned bridge, and conditionally committed on foreign target representations. That result improved the transfer machinery available to an externally specified repair procedure. It did not show that the procedure which *chooses* repairs had itself improved.

Block 7 targets the next empirical level. A parent repair selector is frozen. Development receipts generated through the existing lineage are used to compile a child selector. Parent and child then receive exactly the same candidate repairs, target observations, calibration information, evaluator, and candidate budget. The child receipt is sealed before either final audit split is opened. The primary question is

$$\mathbb{E}_{e\sim Q}\left[U_e(K_1(C_e)) - U_e(K_0(C_e))\right] > 0, \tag{568}$$

where K_0 is the frozen parent selector, K_1 is the lineage-generated child selector, C_e is the same finite repair-candidate bundle for episode e, and U_e is the same protected audit utility.

The realized claim is intentionally narrow. The child does not invent a new repair candidate, packet type, feature language, model architecture, or evaluator. It improves the *selection rule* over a fixed repair grammar. This is the smallest experimentally clean form of improvement-process improvement because the target-level repair opportunity set remains pathwise identical.

31.1 Target episodes and candidate bundles

Let $\mathcal{E}$ be a space of target episodes. An episode $e \in \mathcal{E}$ contains a target agent, a declared defect context, anchor observations, calibration observations, protected audit observations, and all provenance needed to reproduce those objects. Let

$$C_e = (c_{e,0}, \ldots, c_{e,B-1}) \tag{569}$$

be an ordered candidate bundle of fixed size B. Candidate $c_{e,0}$ is the exact no-op. Every nontrivial candidate carries a receipt containing its construction route, rank or complexity, calibration behavior, transport diagnostics, packet residual, and provenance.

Definition 31.1 (Repair selector)**.** A repair selector is a measurable map

$$K : \mathcal{Z}^B \longrightarrow \{0, \dots, B-1\}, \tag{570}$$

where $\mathcal{Z}$ is the space of candidate receipts available before the protected audit outcome is revealed. The selected repair is $c_{e,K(Z_e)}$.

Definition 31.2 (Matched selector comparison)**.** Two selectors K and K' are compared under matched resources when, episode by episode, they receive:

1. the identical candidate bundle C_e in the identical order;
2. the identical receipt fields and calibration evidence;
3. the identical target, anchor, and audit data;
4. the identical candidate-construction budget;
5. the identical protected utility and risk definitions;
6. no policy-specific regeneration, resampling, or extra oracle query.

The matched condition prevents a nominally improved selector from receiving more candidate draws, a larger anchor set, a different evaluator, or a more favorable target distribution.

Definition 31.3 (Selector-level metaproductivity)**.** Let Q be a held-out distribution over target episodes. A child selector K_1 has selector-level metaproductivity over parent K_0 when

$$\Delta_Q(K_1, K_0) := \mathbb{E}_{e\sim Q}\left[G_e(K_1) - G_e(K_0)\right] > 0, \tag{571}$$

under a matched comparison, where $G_e(K)$ is protected utility gain relative to the no-op on episode e.

Remark 31.4. Selector-level metaproductivity is behavioral. If K_0 and K_1 choose from the same candidate bundle, they have the same pointwise reachable repair set. The child may exploit that set better without expanding the authorized correction image. Block 7 therefore does not, by itself, prove structural correctability growth.

31.2 The sealed split discipline

The experiment uses four disjoint episode collections:

$$\mathsf{D}_{\text{dev}} \sqcup \mathsf{D}_{\text{meta}} \sqcup \mathsf{D}_{\text{audit}} \sqcup \mathsf{D}_{\text{arch}}. \tag{572}$$

Development receipts in D_{dev} train the child utility and risk heads. The meta-validation set D_{meta} selects one finite hyperparameter configuration. The seed-and-task audit $\mathsf{D}_{\text{audit}}$ evaluates new episodes from the development architecture-family support. The architecture-foreign audit D_{arch} evaluates representation families absent from both development and meta-validation.

Definition 31.5 (Sealed meta-compiler)**.** A sealed meta-compiler is a map

$$\mathfrak{M} : (\mathsf{D}_{\mathrm{dev}}, \mathsf{D}_{\mathrm{meta}}, K_0, \Gamma) \longmapsto (K_1, \Pi_1)\,, \tag{573}$$

where Γ is a declared finite meta-grammar and Π_1 is a descendant receipt. The compiler is sealed when:

1. neither final audit split is supplied to $\mathfrak{M}$;
2. Π_1 commits to the child state, parent identity, input hashes, grammar, selected configuration, and candidate budget;
3. the audit files are opened only after the receipt has been serialized and hashed;
4. the child state used for final evaluation is byte-equivalent to the state committed by Π_1.

Theorem 31.6 (Compilation noninterference)**.** *Let $K_1 = \mathfrak{M}(D, V, K_0, \Gamma)$, where D and V denote the development and meta-validation receipts. Suppose $\mathfrak{M}$ is deterministic and receives no value measurable only with respect to final audit outcomes A. Then, conditional on (D, V, K_0, Γ),*

$$K_1 \perp\!\!\!\perp A. \tag{574}$$

Equivalently, changing the realized final audit outcomes while holding compiler inputs fixed leaves K_1 unchanged.

Proof. By assumption there is a deterministic function f such that $K_1 = f(D, V, K_0, \Gamma)$. Conditional on its arguments, K_1 is constant. It therefore has a degenerate conditional distribution that does not depend on A. □

Warning 31.7. A digest certifies byte integrity relative to a committed object. It does not by itself prove that an implementation never observed the audit. The released runner strengthens the logical interface by placing development, meta-validation, seed-and-task audit, and architecture-foreign audit in separate files. It loads only the first two, writes the descendant receipt, and only then opens the final audit files. This is process-order evidence, not a hardware-enforced information-flow proof.

Definition 31.8 (One-step bounded repair-process improvement)**.** A lineage edge

$$(\mathcal{O}_0, K_0) \xrightarrow{\Pi_1} (\mathcal{O}_1, K_1) \tag{575}$$

is a one-step bounded repair-process improvement relative to (Q, B, U, R) when:

1. K_1 is generated and sealed through an authorized lineage transaction;
2. parent and child use the same candidate budget B and candidate bundle on each target;
3. a predeclared lower confidence bound for $\Delta_Q(K_1, K_0)$ is positive;
4. the child satisfies declared average, tail, and maximum protected-risk gates;
5. wrong-lineage or shuffled-receipt controls perform worse;
6. ablating the child module restores the parent selection trace;
7. the evaluator, target distribution, and evidence semantics are not changed by the child.

This definition certifies one productive lineage edge. It does not imply recurrent improvement closure, a successful grandchild, or open-ended grammar expansion.

32 A Proof-Carrying Dual-Head Descendant

The child is not a second target solver. It is a learned selector over the five repair candidates already generated for each target episode.

32.1 Candidate features

Each candidate receipt $z_{e,c}$ is mapped to a feature vector

$$x(z_{e,c}) \in \mathbb{R}^{23}. \tag{576}$$

The released feature schema contains:

- a constant coordinate;
- logarithmic anchor and calibration sizes;
- target representation width;
- alignment, transport-condition, and repair-equivariance diagnostics;
- packet residual and relative repair magnitude;
- repair rank;
- calibration baseline slice and protected accuracy;
- calibration gain, protected regression, and protected utility;
- five candidate-type indicators;
- three interaction coordinates coupling gain with alignment, packet residual, and rank-weighted regression.

The schema is fixed before the final audits. It is expressive enough to learn systematic calibration bias, but it is not autonomously discovered.

Let x_i denote a development candidate feature vector, y_i its revealed protected utility, and

$$r_i = [A^{\mathrm{base}}_{i,\mathrm{prot}} - A_{i,\mathrm{prot}}]_+ \tag{577}$$

its protected-regression label. Standardize the feature matrix to Z using development means and scales.

32.2 Utility and risk heads

For regularization parameters $\lambda_u, \lambda_r > 0$, define

$$\widehat{\beta}_u = \left(Z^\top Z + \lambda_u I\right)^{-1} Z^\top y, \tag{578}$$

$$\widehat{\beta}_r = \left(Z^\top Z + \lambda_r I\right)^{-1} Z^\top r. \tag{579}$$

For candidate type c, let $s_{u,c}$ and $s_{r,c}$ be the development residual standard deviations for the utility and risk heads. The deployed scores are

$$\widehat{U}_{\mathrm{LCB}}(z) = \widehat{U}(z) - \kappa_u s_{u,c(z)}, \tag{580}$$

$$\widehat{R}_{\mathrm{UCB}}(z) = \max\{0, \widehat{R}(z)\} + \kappa_r s_{r,c(z)}. \tag{581}$$

A candidate is admissible when

$$\widehat{R}_{\mathrm{UCB}}(z) \le \tau_r, \qquad \widehat{r}_{\mathrm{cal}}(z) \le \tau_{\mathrm{cal}}, \qquad A^{\mathrm{base}}_{\mathrm{cal,prot}}(z) \ge a_{\min}. \tag{582}$$

The child chooses the admissible candidate with maximum $\widehat{U}_{\mathrm{LCB}}$. If the admissible set is empty, it returns the no-op.

Theorem 32.1 (Risk-gate soundness)**.** *Fix an episode and candidate set. Suppose, simultaneously for every candidate c,*

$$R_e(c) \le \widehat{R}_{\mathrm{UCB}}(c). \tag{583}$$

Then any child-selected candidate satisfying $\widehat{R}_{\mathrm{UCB}}(c) \le \tau_r$ has true protected risk at most τ_r.

Proof. For the selected candidate $\widehat{c}$,

$$R_e(\widehat{c}) \le \widehat{R}_{\mathrm{UCB}}(\widehat{c}) \le \tau_r. \tag{584}$$

□

Theorem 32.2 (Utility regret under simultaneous lower bounds)**.** *Let $\mathcal{C}_{\mathrm{safe}}$ be the truly safe candidate set. Suppose every $c \in \mathcal{C}_{\mathrm{safe}}$ passes the implemented admissibility filter, and suppose*

$$0 \le U_e(c) - \widehat{U}_{\mathrm{LCB}}(c) \le \epsilon \tag{585}$$

for every admissible c. If $\widehat{c}$ maximizes $\widehat{U}_{\mathrm{LCB}}$ and c^* maximizes U_e over $\mathcal{C}_{\mathrm{safe}}$, then

$$U_e(c^*) - U_e(\widehat{c}) \le \epsilon. \tag{586}$$

Proof. Because c^* is admissible,

$$\widehat{U}_{\mathrm{LCB}}(\widehat{c}) \ge \widehat{U}_{\mathrm{LCB}}(c^*). \tag{587}$$

The lower-bound property gives

$$U_e(\widehat{c}) \ge \widehat{U}_{\mathrm{LCB}}(\widehat{c}) \ge \widehat{U}_{\mathrm{LCB}}(c^*) \ge U_e(c^*) - \epsilon. \tag{588}$$

□

Remark 32.3. The theorem requires simultaneous validity and safe-set coverage. The released residual penalties are empirical calibration devices, not formal confidence sequences. Their adequacy is tested on disjoint final audits rather than assumed from the development fit.

32.3 Finite meta-grammar selection

The child compiler searches a finite grammar

$$\Gamma = \Lambda_u \times \Lambda_r \times K_u \times K_r \times T_r, \tag{589}$$

where the factors specify utility regularization, risk regularization, utility residual penalty, risk residual penalty, and risk threshold. In the released fixture,

$$|\Gamma| = 5 \cdot 5 \cdot 4 \cdot 5 \cdot 4 = 2000. \tag{590}$$

For $\gamma \in \Gamma$, let $\overline{U}_{\mathrm{meta}}(\gamma)$ be its mean protected utility on the meta-validation episodes and $\widehat{\mathrm{se}}(\gamma)$ its standard error. A configuration is validation-safe when its protected-regression tail rate and maximum regression satisfy the frozen bounds. Among safe configurations, the compiler selects

$$\widehat{\gamma} \in \arg\max_{\gamma \in \Gamma_{\mathrm{safe}}} \left\{ \overline{U}_{\mathrm{meta}}(\gamma) - frac12\widehat{\mathrm{se}}(\gamma) \right\}. \tag{591}$$

Theorem 32.4 (Finite meta-grammar uniform bound)**.** *Suppose, for each $\gamma \in \Gamma$, the per-episode utility lies in an interval of width L, and the n meta-validation episodes are independent. Then with probability at least $1 - \delta$,*

$$\sup_{\gamma \in \Gamma} \left| \widehat{U}_n(\gamma) - U(\gamma) \right| \le L \sqrt{\frac{\log(2|\Gamma|/\delta)}{2n}}. \tag{592}$$

Consequently, the true utility of the empirical maximizer is within twice the right-hand side of the best configuration in Γ.

Proof. Hoeffding's inequality bounds the deviation for one configuration. Apply the union bound to the $2|\Gamma|$ upper and lower deviation events. The empirical-maximizer conclusion follows by inserting the uniform bound on both the winner and the true grammar optimum. □

The theorem explains why a separate final audit remains necessary. Searching 2000 configurations on 60 meta-validation episodes can overfit that split even when the final audit is untouched.

32.4 Selected descendant receipt

The winning frozen configuration is

$$\begin{aligned} \lambda_u &= 100, & \lambda_r &= 0.01, \\ \kappa_u &= 2, & \kappa_r &= 0.5, \\ \tau_r &= 0.035, & \tau_{\mathrm{cal}} &= 0.08, \\ a_{\min} &= 0.80. \end{aligned} \tag{593}$$

Its meta-validation score is 0.291029, from mean protected utility 0.309844 minus one-half standard error. The validation protected-regression tail rate is 0.016667 and the maximum is 0.054463.

The descendant receipt records:

- the parent policy identifier and lineage generation;
- development and meta-validation receipt hashes;

- the compiler source hash and candidate-feature-schema hash;
- the complete 2000-point meta-grammar;
- the selected configuration and validation witness statistics;
- feature means, feature scales, fitted utility and risk weights, and candidate-specific residual scales;
- the fixed candidate budget;
- the final receipt digest.

33 Matched-Budget Attribution of the Meta-Gain

An observed child advantage can be attributed to the child selector only when every other target-level opportunity is held fixed.

Theorem 33.1 (Pathwise selector isolation)**.** *For episode e, let parent and child receive the identical ordered bundle C_e and identical pre-audit receipt Z_e. Let*

$$i_0 = K_0(Z_e), \qquad i_1 = K_1(Z_e). \tag{594}$$

Then the realized paired utility difference is exactly

$$D_e = U_e(c_{e,i_1}) - U_e(c_{e,i_0}). \tag{595}$$

If candidate construction, target state, and audit evaluation are frozen, no candidate-generation term remains in this difference.

Proof. Both policies are evaluated on the same episode and the same candidate utility vector

$$(U_e(c_{e,0}), \dots, U_e(c_{e,B-1}))\,.$$

Their outputs differ only through the selected indices i_0 and i_1. Substitution gives Equation (595). □

Corollary 33.2 (Selector ablation)**.** *If the descendant module is removed and the parent map K_0 is reinstated without changing any other runtime object, the resulting selection and utility trace is pathwise identical to the parent trace.*

The executable control copies the complete parent metrics into the selector-ablation arm and separately tests equality on both final audit splits.

33.1 The best-of-k confound

Let $M_B(K)$ denote expected best gain when policy K receives B independently generated candidate opportunities. Even if the policy is unchanged,

$$M_{B+1}(K) \geq M_B(K). \tag{596}$$

Therefore, comparing a child with $B_1 > B_0$ against a parent with budget B_0 does not isolate improvement of the selection procedure.

Proposition 33.3 (No-free-expansion attribution)**.** *Suppose the child receives the parent's B candidates plus one additional candidate. Then a positive child-parent gain can occur even when the child applies the parent selector to the original candidates and uses no improved inference. Hence a positive unmatched-budget gain is not sufficient evidence for repair-process improvement.*

Proof. Let the additional candidate have strictly higher utility than all original candidates with positive probability, and let the child choose it whenever present. The child improves on those episodes without any change to the rule applied to the original bundle. □

Block 7 fixes $B = 5$ for every arm and target.

33.2 Lineage and semantic controls

The child is trained from candidate receipts whose utility and risk labels are attached to semantically typed candidate features. The shuffled control independently permutes those outcome labels before fitting the same dual-head form.

Proposition 33.4 (Shuffled-label null)**.** *Suppose development labels are exchangeable and independently permuted relative to candidate features. Conditional on the feature matrix, the expected feature-label cross-covariance is zero after centering. A linear ridge head fitted to infinitely many such permutations therefore converges in expectation to the intercept-only predictor.*

Proof. Let Z be column centered and let P be a uniformly random permutation matrix independent of y. Then

$$\mathbb{E}_P[Z^\top P y] = Z^\top \mathbb{E}_P[Py] = Z^\top \bar{y}\mathbf{1} = 0. \tag{597}$$

The ridge coefficients are linear in $Z^\top P y$ for fixed Z. □

Finite permutations can retain accidental structure, so the control need not fall to zero utility. The required result is that the correctly bound lineage receipts yield a stronger final policy.

33.3 Selection versus structural correctability

Proposition 33.5 (Fixed-candidate structural invariance)**.** *Let $\mathcal{C}_e$ be the authorized candidate set for episode e. If parent and child differ only in the selector and*

$$\mathcal{C}_e^{(0)} = \mathcal{C}_e^{(1)} \tag{598}$$

for every e, then the pointwise reachable correction set and its linear span are identical for parent and child.

Proof. Both sets are equal by hypothesis, so their spans and all set-valued reachability properties determined only by candidate membership are equal. □

The Block 7 child improves the probability of choosing a useful available correction. It does not add a new correctable defect direction. This is why the empirical claim is *selector-level* metaproductivity rather than conservative repair-grammar expansion.

34 Protected Utility, Risk, and Statistical Certification

34.1 Protected target utility

For target episode e and candidate c, let

$$A_e^{\text{slice}}(c) = \text{accuracy on the affected stratum}, \tag{599}$$
$$A_e^{\text{prot}}(c) = \text{accuracy on the unaffected protected stratum}, \tag{600}$$
$$q(c) = \text{repair rank}. \tag{601}$$

The candidate utility is

$$U_e(c) = A_e^{\text{slice}}(c) - 2\left[A_e^{\text{prot}}(0) - A_e^{\text{prot}}(c)\right]_+ - 0.0005q(c), \tag{602}$$

and the reported protected utility gain is

$$G_e(c) = U_e(c) - A_e^{\text{slice}}(0). \tag{603}$$

The protected regression is

$$R_e(c) = \left[A_e^{\text{prot}}(0) - A_e^{\text{prot}}(c)\right]_+. \tag{604}$$

The utility therefore rewards repair of the affected stratum, penalizes damage to unrelated behavior twice as strongly, and applies a small complexity charge.

34.2 Paired inference

For n held-out episodes, define paired differences

$$D_j = G_{e_j}(K_1) - G_{e_j}(K_0), \qquad \widehat{\Delta}_n = \frac{1}{n}\sum_{j=1}^{n} D_j. \tag{605}$$

Because parent and child share each target and candidate bundle, pairing removes much of the episode difficulty variation.

Theorem 34.1 (Distribution-free paired lower bound)**.** *Suppose $D_1, \dots, D_n$ are independent and lie almost surely in $[a, b]$. Then with probability at least $1 - \alpha$,*

$$\Delta_Q \geq \widehat{\Delta}_n - (b - a)\sqrt{\frac{\log(1/\alpha)}{2n}}. \tag{606}$$

A positive right-hand side certifies $\Delta_Q > 0$ at confidence $1 - \alpha$ under the assumptions.

Proof. Apply the one-sided Hoeffding inequality to the sample mean of the bounded paired differences. □

The executable package additionally reports a frozen 20,000-draw paired bootstrap interval and one-sided sign-flip randomization p-value. Those procedures provide the reported finite-sample summaries. They do not replace the need to declare exchangeability or sampling assumptions.

34.3 Cumulative claim gates

The child must pass performance and risk gates simultaneously. Let

$$\mathsf{Gate} = \mathsf{Seal} \wedge \mathsf{Match} \wedge \mathsf{AuditGain} \wedge \mathsf{ForeignGain} \wedge \mathsf{Risk} \wedge \mathsf{Controls}. \tag{607}$$

The released suite evaluates the following frozen selector-gate families:

1. descendant receipt validity;
2. split-file load order and sealing;
3. positive paired child-parent audit interval;
4. positive paired child-parent architecture-foreign interval;
5. positive child-raw architecture-foreign interval;
6. three seed-and-task audit regression gates;
7. three architecture-foreign regression gates;
8. two shuffled-lineage controls;
9. two selector-ablation equalities;
10. matched five-candidate budget;
11. exclusion of audit receipts from compiler inputs;
12. disjoint architecture-family supports.

All component gates must pass before the aggregate gate is true.

Warning 34.2. The seed-and-task audit does not resolve child superiority over the strong raw-calibration baseline: the paired mean is 0.005922 with bootstrap interval $[-0.000677, 0.013639]$. Child superiority over raw calibration is supported only on the architecture-foreign split in this release. The main parent-child claim remains positive on both splits.

34.4 Risk-head ablation

Removing the risk head expands the child's admissible set. This can raise utility while worsening protected risk.

Proposition 34.3 (Risk-filter tradeoff)**.** *Let $\mathcal{A}_r \subseteq \mathcal{A}_0$ be the candidate set retained with and without a risk filter. Then*

$$\max_{c \in \mathcal{A}_0} \widehat{U}(c) \geq \max_{c \in \mathcal{A}_r} \widehat{U}(c), \tag{608}$$

but no analogous inequality favors the unfiltered selector for true protected risk.

Proof. The utility inequality follows from set inclusion. Risk is not the optimized scalar in the unfiltered maximum, so it admits no monotonicity conclusion. □

Empirically, the no-risk child has higher mean utility than the protected child on both final splits, but it violates the architecture-foreign average and tail risk gates. The realized result is therefore a constrained improvement claim, not an assertion that the selected child maximizes unconstrained utility.

35 Executable Block 7 Benchmark

35.1 Task and defect generator

For $x \in \mathbb{R}^4$, define an eight-dimensional task representation

$$\psi(x) = (x_1, x_2, x_3, x_4, x_1x_2, x_3x_4, \sin(sx_1), \cos(sx_2)) . \tag{609}$$

A task-specific condition vector and threshold define the affected indicator

$$a(x) = \mathbf{1}\{\langle v, x\rangle > \tau\}. \tag{610}$$

The conditional feature is

$$\phi(x) = \big(\psi(x), a(x)\psi(x)\big) \in \mathbb{R}^{16}. \tag{611}$$

True three-class score vectors are affine in $\psi(x)$. On the affected stratum only, one of five frozen score transformations corrupts those scores. Target agents are fitted to the corrupted scores, so their baseline behavior is generally accurate off the affected stratum and defective on it.

35.2 Representation families

Development, meta-validation, and seed-and-task audit episodes draw from eight families:

$$\begin{gathered}\texttt{linear16},\quad \texttt{overcomplete24},\quad \texttt{ill_conditioned16},\quad \texttt{compressed12},\\ \texttt{mild_tanh24},\quad \texttt{saturated_tanh24},\quad \texttt{relu32},\quad \texttt{noisy_linear16}.\end{gathered} \tag{612}$$

The architecture-foreign audit uses four disjoint families:

$$\texttt{softsign24},\quad \texttt{piecewise24},\quad \texttt{quadratic24},\quad \texttt{bottleneck14}. \tag{613}$$

The family names and construction code are frozen in the candidate receipt archive.

35.3 Five-candidate repair grammar

Every episode exposes exactly

$$C_e = \{\texttt{no_op}, \texttt{transport}, \texttt{local_r4}, \texttt{local_r8}, \texttt{local_full}\}. \tag{614}$$

The transported candidate moves a rank-eight source repair through a learned source-to-target bridge. The rank-four and rank-eight local candidates use target-side intervention differences with noisy or incomplete probes. The full local candidate fits an unrestricted affine hidden-state correction. Candidate construction is complete before either parent or child selects.

Episode difficulty varies through target family, task seed, encoder seed, anchor size, calibration size, probe noise, and probe leakage. Anchor sizes lie in

$$\{16, 24, 32, 48, 64, 96, 128\}, \tag{615}$$

calibration sizes in

$$\{48, 64, 96, 128\}, \tag{616}$$

probe-noise levels in

$$\{0, 0.02, 0.05, 0.10, 0.18\}, \tag{617}$$

and probe-leak levels in

$$\{0, 0.05, 0.12, 0.25\}. \tag{618}$$

35.4 Episode counts and lineage chronology

Split	Episodes	Candidate receipts	Function
Development	120	600	fit utility and risk heads
Meta-validation	60	300	choose one of 2000 meta-configurations
Seed/task audit	120	600	held-out same-support evaluation
Architecture-foreign audit	120	600	held-out family evaluation

Table 21: Block 7 episode and candidate counts. Every episode contains the same five candidate types.

The chronology is:

$$\begin{aligned} &\text{open development file and fit heads} \\ &\longrightarrow \text{open meta-validation file and choose configuration} \\ &\longrightarrow \text{serialize and hash descendant receipt} \\ &\longrightarrow \text{open seed/task audit file} \\ &\longrightarrow \text{open architecture-foreign audit file} \\ &\longrightarrow \text{evaluate every frozen policy and claim gate.} \end{aligned}$$

35.5 Comparison arms

Arm	Role
Parent	Frozen conservative priority rule using calibration gain, protected regression, baseline protected accuracy, and transport alignment.
Raw calibration	Strong ordinary baseline choosing maximum measured calibration utility inside a fixed safety envelope.
Descendant	Sealed dual-head utility/risk selector compiled from development and meta-validation receipts.
No-risk descendant	Same child utility head and selected meta-configuration, with the risk-head exclusion removed.
Shuffled descendant	Same dual-head form fitted after permuting development utility and risk labels relative to candidate features.
Selector ablation	Descendant module removed and exact parent policy reinstated.
Audit oracle	Inadmissible upper comparison that directly chooses maximum audit utility.

The audit oracle is not a deployable baseline. It estimates the remaining opportunity available inside the fixed five-candidate grammar.

36 Block 7 Results

36.1 Primary parent–child comparison

Split and arm	Mean utility gain	Mean regression	Tail rate	Success
Seed/task parent	0.284896	0.001774	0.008333	0.508333
Seed/task child	**0.357414**	0.007787	0.050000	0.700000
Architecture parent	0.167417	0.004831	0.008333	0.350000
Architecture child	**0.272416**	0.012852	0.033333	0.666667

Table 23: Primary matched-budget results. Tail rate is the fraction of episodes with protected regression above 0.05.

The paired effects are

$$\widehat{\Delta}_{\text{audit}} = 0.072518, \qquad \text{CI}_{95\%} = [0.046745, 0.101389], \tag{619}$$

$$\widehat{\Delta}_{\text{arch}} = 0.104999, \qquad \text{CI}_{95\%} = [0.072711, 0.140506]. \tag{620}$$

Both one-sided sign-flip p-values equal approximately 4.99975×10^{-5} under the frozen 20,000-draw randomization procedure.

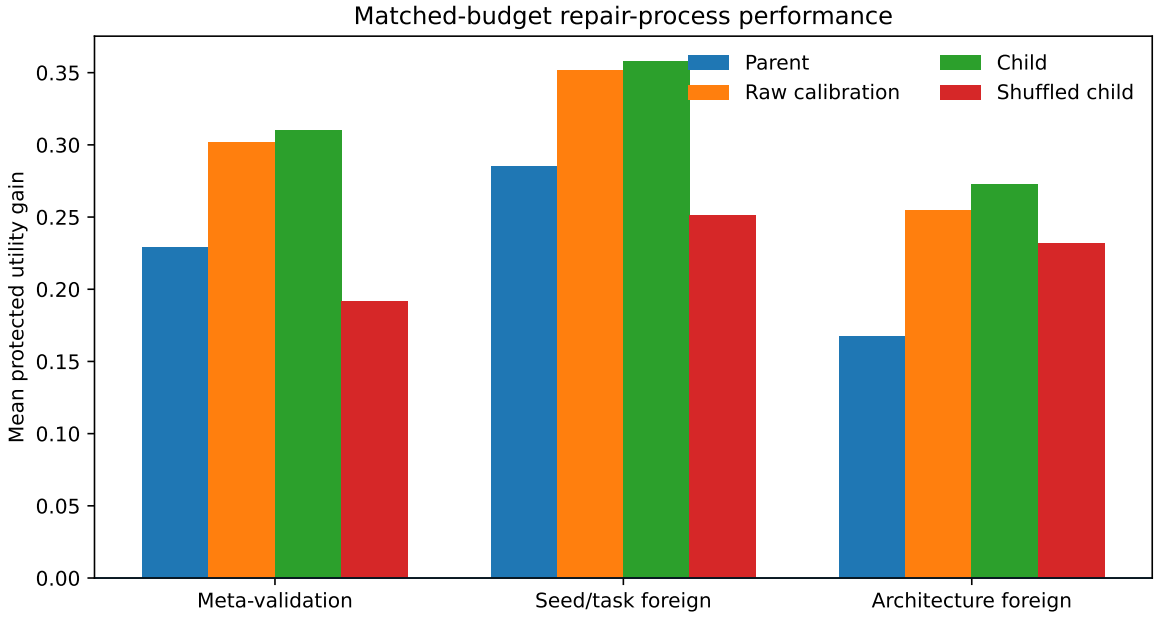


Figure 6: Mean protected utility under the matched five-candidate budget. The child improves over its parent on the meta-validation, seed-and-task audit, and architecture-foreign audit splits.

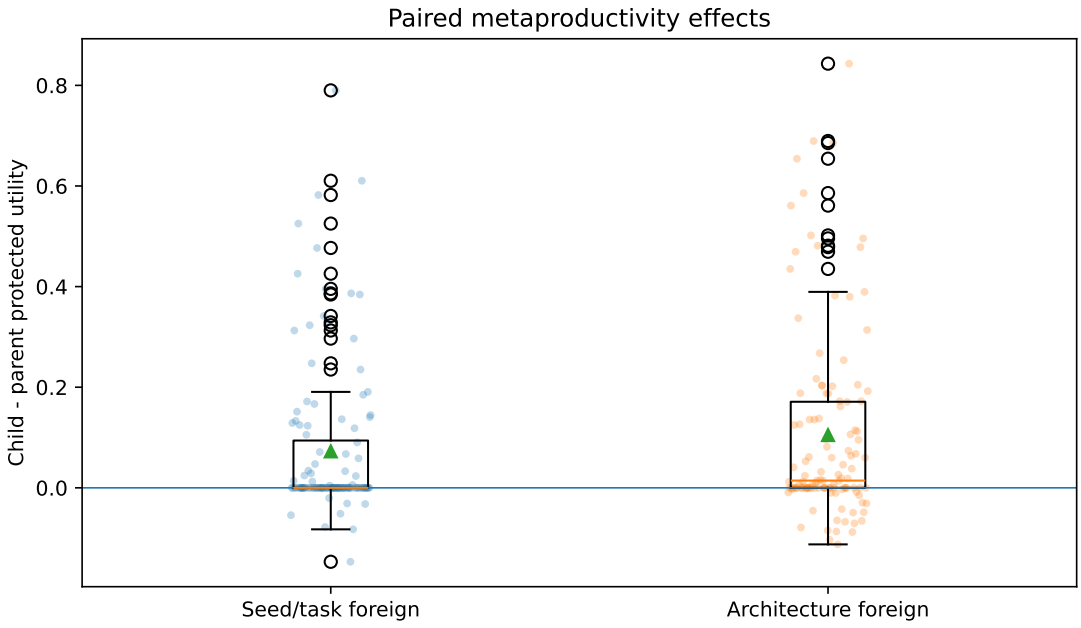


Figure 7: Episode-wise child-minus-parent protected utility. Many episodes are ties because both policies select the same candidate; positive mean shifts remain on both final splits.

36.2 Strong-baseline and oracle comparisons

Arm	Meta-validation	Seed/task audit	Architecture foreign
Parent	0.229208	0.284896	0.167417
Raw calibration	0.301919	0.351492	0.254427
Descendant	**0.309844**	**0.357414**	**0.272416**
No-risk descendant	0.312660	0.362862	0.293841
Shuffled descendant	0.191694	0.250997	0.231744
Audit oracle	0.331609	0.402808	0.324703

Table 24: Mean protected utility gain for all substantive arms. The no-risk arm is not admissible under the frozen architecture-foreign risk gates.

On the architecture-foreign split, the child's paired advantage over raw calibration is

$$0.017989, \qquad \mathrm{CI}_{95\%} = [0.003477, 0.036611], \qquad p_{\mathrm{sign}} = 0.00940. \tag{621}$$

On the seed-and-task audit, the corresponding interval includes zero. The result therefore supports robust improvement over the frozen parent on both audits and a smaller architecture-foreign improvement over a strong ordinary selector.

The audit oracle shows that the fixed candidate bundle retains additional exploitable utility. The child captures approximately 76.9% of per-episode oracle opportunity on the seed-and-task audit and a smaller fraction under architecture shift. The remaining gap can arise from prediction error, risk conservatism, missing candidate features, or meta-grammar limits.

36.3 Risk-utility tradeoff

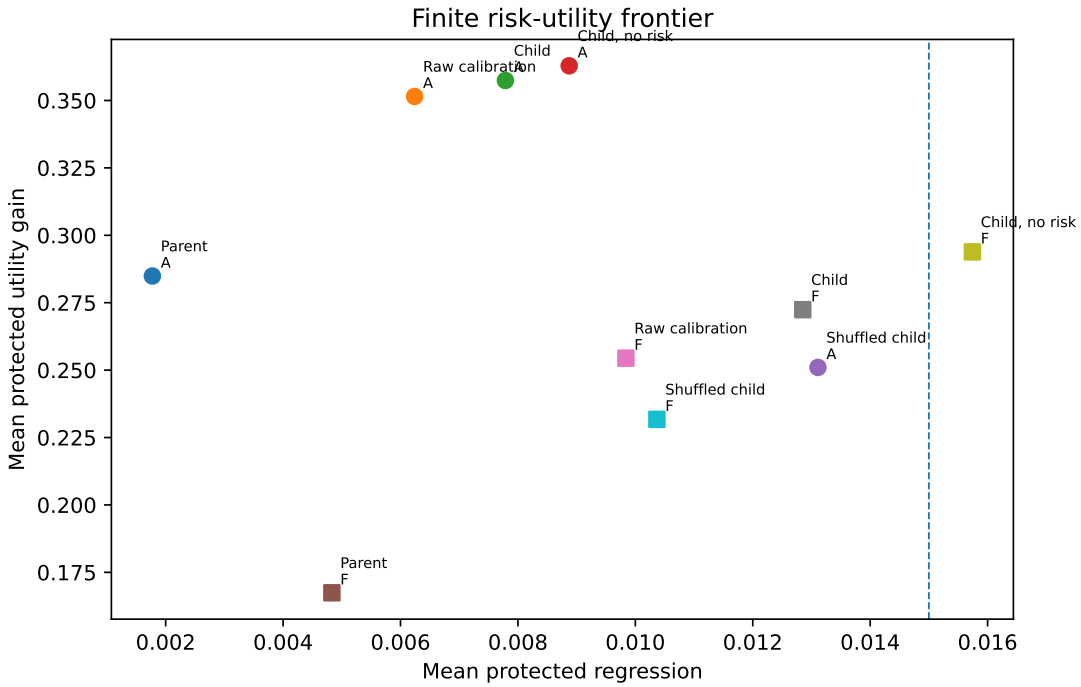


Figure 8: Mean protected utility versus protected regression. The no-risk descendant obtains more utility but crosses the frozen architecture-foreign average and tail risk gates.

For the architecture-foreign split,

$$\overline{R}_{\text{child}} = 0.012852, \qquad \Pr(R > 0.05) = 0.033333, \tag{622}$$

$$\overline{R}_{\text{no risk}} = 0.015741, \qquad \Pr(R > 0.05) = 0.058333. \tag{623}$$

The risk head therefore reduces both mean and tail protected regression, at a utility cost of 0.021425 relative to the no-risk arm. This is expected from constrained selection and should not be described as unconstrained dominance.

36.4 Lineage and ablation controls

The correctly bound child outperforms the shuffled child by

$$0.106417 \quad \text{on the seed/task audit,} \tag{624}$$

$$0.040672 \quad \text{on the architecture-foreign audit.} \tag{625}$$

The corresponding paired bootstrap intervals are strictly positive. The selector-ablation arm exactly reproduces the parent metrics on both splits, as required by Theorem 33.2.

The child changes the repair mixture substantially.

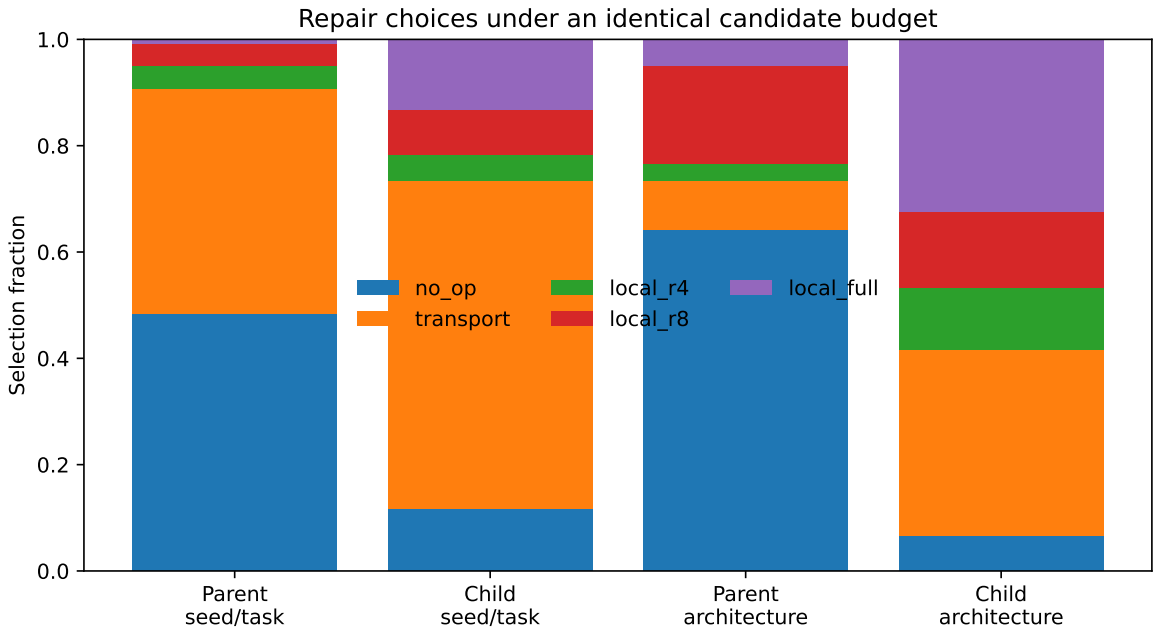


Figure 9: Candidate-selection fractions. The parent abstains frequently, especially under architecture shift. The child uses learned utility and risk estimates to select transport and local repairs more often while retaining a no-op fallback.

On the architecture-foreign audit, the parent selects no-op in 77 of 120 episodes, while the child selects no-op in only 8. The child selects transported repair 42 times and full local repair 39 times. This increased intervention rate is acceptable only because the protected-risk gates are checked independently.

36.5 Frozen claim-gate ledger

Gate class	Status
Receipt integrity and compiler-input exclusion	PASS
Split-file seal-before-audit load order	PASS
Matched five-candidate budget	PASS
Positive child-parent seed/task paired interval	PASS
Positive child-parent architecture-foreign paired interval	PASS
Positive child-raw architecture-foreign paired interval	PASS
Seed/task mean, tail, and maximum regression gates	PASS
Architecture-foreign mean, tail, and maximum regression gates	PASS
Shuffled-lineage controls on both final splits	PASS
Selector-ablation equality on both final splits	PASS
Disjoint architecture-family support	PASS
All frozen selector gates	**PASS**

Table 25: Block 7 primary gate ledger.

37 Formal Interpretation of the Block 7 Edge

Conditional Theorem 37.1 (One-step selector metaproductivity certificate)**.** *Let K_0 be a frozen parent repair selector and let*

$$(K_1, \Pi_1) = \mathfrak{M}(\mathsf{D}_{\mathrm{dev}}, \mathsf{D}_{\mathrm{meta}}, K_0, \Gamma) \tag{626}$$

be a child and valid descendant receipt. Assume:

1. *the compiler satisfies the sealing conditions of Theorem 31.6;*
2. *parent and child receive pathwise identical candidate bundles and evidence under Theorem 33.1;*
3. *the final target episodes are sampled from the declared held-out distribution and the paired interval procedure has its declared coverage;*
4. *the lower confidence bound for child-minus-parent protected utility is positive;*
5. *all declared average, tail, and maximum risk gates hold;*
6. *the shuffled-lineage control is lower and selector ablation restores the parent;*
7. *evaluator semantics, provenance, candidate budget, and audit evidence are fixed across the comparison.*

Then the lineage edge $K_0 \xrightarrow{\Pi_1} K_1$ is a certified one-step bounded improvement of the repair-selection process relative to the declared distribution, candidate grammar, feature schema, evaluator, witness, and budget.

Proof. By sealing, K_1 is fixed without final audit access. By the pathwise matched comparison, the paired difference contains only the effect of selecting different members of the same candidate bundle. The positive lower confidence bound establishes positive held-out expected protected utility under the declared sampling assumptions. The risk gates exclude the specified forms of protected regression. Shuffled and ablation controls respectively test semantic receipt binding and necessity of the compiled child selector. The fixed evaluator, candidate budget, and provenance remove the declared alternative explanations. These conditions are exactly the one-step bounded repair-process improvement definition. □

Corollary 37.2 (Productive lineage edge)**.** *Under the hypotheses of Theorem 37.1, the parent state has an authorized successor with positive foreign held-out improvement-process gain. Therefore the edge is productive in the finite transaction graph.*

Warning 37.3 (One edge is not recurrent closure)**.** The theorem certifies neither

$$K_1 \longrightarrow K_2 \tag{627}$$

nor an infinite sequence of productive descendants. No grandchild is compiled in Block 7, and no fixed-point or Buechi liveness claim follows from a single successful edge.

37.1 What changed in the child

The child adds one lineage-generated operator:

$$\Theta_{\text{select}} : \{\text{typed candidate receipts}\} \longrightarrow \{\text{risk-filtered repair choice}\}. \tag{628}$$

This operator uses learned outcome calibration and candidate-type residual scales. The changed component is therefore the repair-selection policy, not the target agent's weights or the repair constructor.

37.2 What did not change

The following remain externally fixed:

- the target-task and defect generator;
- the five repair candidates and their constructors;
- the 23-dimensional candidate feature schema;
- the 2000-point meta-grammar;
- the protected utility and risk definitions;
- the parent and baseline policies;
- the final claim gates;
- the target audit distributions;
- the model architecture code and representation families.

Consequently, Block 7 does not establish autonomous repair invention, autonomous feature discovery, grammar expansion, model-weight self-rewrite, or evaluator evolution.

37.3 Claim-ladder advancement

Block 14 attacks the simplification on which the Block 13 mechanism depended. Initial diagnostics are aliased, some problems contain multiple simultaneous obstructions, some lie outside the inherited response cone, the development archive is contaminated, evaluator weights shift, and candidate methods include integrity attacks. Under the exact inherited 39-method language and the same six full experiments, primary mean validity-filtered gain rises from 0.410813 to 0.575783 and generator-foreign gain rises from 0.389514 to 0.550559. Primary lower-tail gain changes from -0.133065 to 0.271516. The child selects no invalid method in either final audit and reduces false confident known-class assignment on unknown problems from 1 to 0.083333.

This is a robustness edge, not a universal robustness theorem. The stress atlas, response language, candidate grammar, evaluator portfolio, monitor ensemble, and attack generator remain externally declared. The primary paired comparison also contains retained negative episodes. Sequential response updating, open-set routing, and robust archive estimation receive positive support. The simple-monitor ablation is null and the single-portfolio ablation is nearly null, so those modules do not receive independent positive attribution. The result therefore strengthens the evidence for a compiler that can reallocate experiments under finite ambiguity and attack, while leaving

independent replication, real executable research, adaptive adversaries, and compiler-language growth unresolved.

The empirical ladder is now:

1. mechanical implementation;
2. finite semantic realization;
3. causal identification;
4. controlled foreign repair transfer;
5. **sealed one-step repair-process metaproductivity**;
6. repeated lineage metaproductivity and autonomous schema growth;
7. open-set recursive improvement closure.

Block 7 passes level 5 relative to the declared finite benchmark. It does not pass levels 6 or 7.

Block 7 empirical result

A lineage-generated dual-head repair selector, sealed before final audit access, receives the same five candidate repairs as its frozen parent and achieves positive paired protected-utility gain on both seed/task-foreign and architecture-foreign target episodes. Every frozen selector gate passes. This is a bounded one-step improvement of the repair-selection process. Because the candidate grammar, feature schema, meta-grammar, evaluator, and witness are fixed, it is not autonomous repair invention, structural correctability expansion, repeated recursive closure, or open-ended RSI.

38 Block 7 Reproducibility and Package

The executable package [7] contains the complete candidate receipt archive, split-file sealing protocol, child compiler, parent and control policies, frozen results, figures, tests, manifest, and Windows runner.

38.1 Package layout

```
SRIC_v0.8_Block07_Sealed_Meta_Improvement/
  README.md
  pyproject.toml
  Makefile
  Run_SRIC_Block07.ps1
  regenerate_candidates.py
  run_all.py
  build_manifest.py
  verify_package.py
  MANIFEST.sha256
  sric_meta/
    benchmark.py
```

```
    canonical.py
    io.py
    meta.py
    pipeline.py
    plots.py
    receipts.py
    statistics.py
  results/
    development_candidate_receipts.json.gz
    validation_candidate_receipts.json.gz
    audit_candidate_receipts.json.gz
    architecture_foreign_candidate_receipts.json.gz
    frozen_candidate_receipts.json.gz
    descendant_receipt.json
    block07_results.json
    audit_selected_rows.csv
    architecture_foreign_selected_rows.csv
  figures/
    block07_metaproductivity.pdf
    block07_paired_effects.pdf
    block07_risk_utility_frontier.pdf
    block07_candidate_choices.pdf
  tests/
    conftest.py
    test_controls.py
    test_primary.py
    test_receipt.py
    test_result_values.py
```

38.2 Canonical replay

The standard PowerShell replay is one line:

```
python verify_package.py; python run_all.py; python -m pytest -q
```

A full candidate regeneration is available through:

```
python regenerate_candidates.py; python run_all.py;
python -m pytest -q; python build_manifest.py;
python verify_package.py
```

The second path reconstructs every target episode and every repair candidate before recompiling the child.

38.3 Frozen verification summary

The final reference run reports:

```
SRIC_BLOCK07_ALL_SUITES=PASS
meta_grammar_size=2000
audit_parent_gain=0.284896
audit_child_gain=0.357414
audit_child_minus_parent=0.072518
audit_ci_low=0.046745
foreign_child_minus_parent=0.104999
foreign_ci_low=0.072711
............ [100%]
SRIC_BLOCK07_PACKAGE_VALID
```

The frozen result digest is

```
cc92cb004e2d4cdb1b0ea1c30e64683a961f507d8b24225190eca8dfd72d625a
```

and the internal descendant-receipt digest is

```
852fc497f19561bc9c16b8cb9fa0b126b8104812b46e53c72f8b4bf27207e576
```

38.4 Remaining reproducibility limitations

The package gives deterministic reconstruction relative to its software dependencies and frozen pseudorandom seeds. It does not yet provide:

- an independently written second implementation;
- process-level mandatory access control proving audit nonobservability;
- preregistered independent seed replication;
- confidence sequences valid under arbitrary adaptive meta-grammar growth;
- target models trained with large neural backbones;
- a grandchild run demonstrating repeatability of the meta-improvement edge.

The reference package should therefore be treated as a finite executable existence proof for the declared selector-level mechanism and its controls, not as an unrestricted self-improving agent.

39 Sealed Parent–Child Bridge-Policy Compiler

The selection-only realization above changes which already constructed repair is committed. A second Block 7 realization changes an earlier stage of the repair process: which paired anchors are queried and whether the resulting bridge is sufficiently identified to transport a frozen repair. A parent bridge process is run on a development lineage, its failures are summarized, and a descendant bridge policy is compiled from a fixed meta-grammar. The descendant is then sealed before any audit task or audit representation family is executed.

This second experiment is also deliberately narrow. The descendant may change only:

1. the rule used to select paired source–target anchors;
2. the bridge-identification gate applied before commitment.

The source repair, target-query budget, bridge family, evaluator, audit metric, and repair-transport formula remain fixed. This isolates one repair-process modification rather than allowing a general rewrite.

39.1 Repair processes as objects of comparison

Definition 39.1 (Finite repair process)**.** A finite repair process is a tuple

$$\mathfrak{P} = (\pi_{\mathrm{anc}}, \mathsf{B}, \pi_{\mathrm{gate}}, B, \Pi), \tag{629}$$

where π_{anc} selects target correspondence queries, B estimates a bridge and transports a frozen source repair, π_{gate} commits or abstains, B is the declared resource budget, and Π is the receipt schema.

For a foreign target system A and protected evaluation context Ξ, let

$$G_{\Xi}(\mathfrak{P}, A) = J_{\Xi}(\mathfrak{P}[A]) - J_{\Xi}(A) \tag{630}$$

be the gain produced when the process is given one opportunity to repair A. The notation $\mathfrak{P}[A]$ includes abstention, in which case the gain is zero.

Definition 39.2 (Matched-budget foreign metaproductivity)**.** For a foreign target distribution $\mathcal{D}_{\mathrm{for}}$ and budget B, define

$$M_B^{\mathrm{for}}(\mathfrak{P}) = \mathbb{E}_{A \sim \mathcal{D}_{\mathrm{for}}} \left[G_{\Xi}(\mathfrak{P}, A) \right], \tag{631}$$

provided every compared process receives the same target correspondence, calibration, candidate, and irreversible-effect budgets.

The protected utility used in Block 7 is the balanced accuracy functional

$$J_{\Xi}(A) = \frac{1}{2} A_{\mathrm{aff}}(A) + \frac{1}{2} A_{\mathrm{non}}(A), \tag{632}$$

where A_{aff} is accuracy on the defect-affected stratum and A_{non} is accuracy on the protected unaffected stratum. This prevents the more common unaffected examples from overwhelming the repair objective.

Definition 39.3 (One-step bounded recursive meta-improvement)**.** Let $\mathfrak{P}_0$ be a parent repair process. Let a lineage compiler Meta receive only a development ledger L_{dev} and return

$$\mathfrak{P}_1 = \mathrm{Meta}(\mathfrak{P}_0, L_{\mathrm{dev}}). \tag{633}$$

The edge $\mathfrak{P}_0 \to \mathfrak{P}_1$ is a one-step bounded recursive meta-improvement on $(\mathcal{D}_{\mathrm{for}}, B)$ when:

1. the descendant is sealed before the audit distribution is revealed;
2. $\mathfrak{P}_0$ and $\mathfrak{P}_1$ receive the same deployment budget B;
3. $M_B^{\mathrm{for}}(\mathfrak{P}_1) > M_B^{\mathrm{for}}(\mathfrak{P}_0)$ with a declared uncertainty certificate;
4. the descendant preserves every protected invariant and has no larger accepted-harm rate;
5. a structural repair-capacity measure also improves on the declared foreign family.

This definition is intentionally local. It certifies one descendant edge. It does not imply repeated improvement, autonomous grammar expansion, or supercritical repair discovery.

39.2 Development, seal, and audit separation

Let

$$\Omega = \Omega_{\mathrm{dev}} \sqcup \Omega_{\mathrm{audit}} \tag{634}$$

be a disjoint target index set. The compiler receives rows indexed only by Ω_{dev}. It emits a descendant configuration c_1, a compiler ledger L_{Meta}, and a pre-audit seal

$$\Sigma_{\mathrm{pre}} = H\big(c_1, L_{\mathrm{Meta}}, \text{source hashes}\big), \tag{635}$$

where H is SHA-256 over canonical serialization. Audit execution refuses to proceed when the in-memory descendant differs from the sealed configuration.

Proposition 39.4 (Trace-level audit nonconditioning)**.** *Suppose the compiler is a deterministic function of* $(\mathfrak{P}_0, L_{\mathrm{dev}})$*, the audit runner accepts only a descendant whose canonical digest equals the committed digest in* (635)*, and no audit field is present in the sealed compiler input. Then the executed descendant is not a direct function of the audit observations inside the declared runtime trace.*

Proof. The compiler output is fixed by its development input. The audit runner checks equality with the precommitted digest before evaluating any audit row. Any later replacement changes the digest and fails the check. Thus the executed configuration is fixed before the audit observations enter the declared trace. □

Remark 39.5. The proposition is a software-trace statement, not a metaphysical guarantee against every hidden side channel. It is only as strong as the source audit, process isolation, canonical serializer, and manifest verification.

39.3 Foreignness and matched resources

The Block 7 primary audit differs from development in both task and representation family:

$$\begin{aligned} \mathsf{T}_{\mathrm{dev}} \cap \mathsf{T}_{\mathrm{audit}} &= \varnothing, \\ \mathsf{F}^{\mathrm{primary}}_{\mathrm{dev}} \cap \mathsf{F}^{\mathrm{primary}}_{\mathrm{audit}} &= \varnothing. \end{aligned} \tag{636}$$

Development uses ten tasks generated from four trigger families. Audit uses twelve new tasks generated from radial and mixed triggers, new teacher seeds, and new corruption permutations. The primary audit representations are dense linear 20-dimensional, block-linear 24-dimensional, and ill-conditioned linear 20-dimensional systems. None appears in the primary development set.

The parent and descendant each receive:

$$B = (16 \text{ fit anchors}, 8 \text{ validation anchors}, 1 \text{ repair candidate}). \tag{637}$$

They use the same source repair, target calibration set, protected audit set, and bridge estimator. Only anchor selection and the gate differ.

Block 7 comparison principle

A positive result must be a paired improvement of the repair process, not merely a stronger target agent. The parent and descendant act on the same foreign targets, with the same source repairs and target-query budget. The descendant configuration is compiled from development failures and sealed before audit.

40 Meta-Repair of Bridge Identification

40.1 The parent failure

The Block 6 bridge estimator receives paired activations

$$X = \begin{bmatrix} h_s(x_1)^\top \\ \vdots \\ h_s(x_k)^\top \end{bmatrix}, \qquad Y = \begin{bmatrix} h_t(x_1)^\top \\ \vdots \\ h_t(x_k)^\top \end{bmatrix}, \tag{638}$$

and fits

$$\widehat{T}^\top = (X^\top X + \lambda I)^{-1} X^\top Y. \tag{639}$$

The parent chooses the $k = 16$ fit anchors uniformly at random. The source hidden space is also 16-dimensional, so a rank-deficient design leaves at least one source direction unidentified.

The task representation has the conditional form

$$\phi(x) = \begin{bmatrix} \psi(x) \\ s(x)\psi(x) \end{bmatrix} \in \mathbb{R}^{16}, \tag{640}$$

where the second eight-dimensional packet is zero outside the affected stratum. Randomly chosen anchors often include too few affected examples to span that packet. This is not a failure of ridge regression. It is an experimental-design obstruction.

40.2 Bridge identifiability

Theorem 40.1 (Full-rank bridge identification)**.** *Let the target representation satisfy*

$$h_t(x) = T h_s(x) \tag{641}$$

for a fixed linear map $T : \mathbb{R}^d \to \mathbb{R}^m$. Let $X \in \mathbb{R}^{k \times d}$ be the selected source design and $Y = XT^\top$. If $\operatorname{rank} X = d$*, then the unregularized least-squares bridge is unique and equals T on the entire source space. If* $\operatorname{rank} X < d$*, there exist distinct maps $T_1 \neq T_2$ such that*

$$XT_1^\top = XT_2^\top = Y. \tag{642}$$

Consequently, repair transport is not identifiable on every unobserved source direction.

Proof. When X has full column rank, $X^\top X$ is invertible and

$$(X^\top X)^{-1} X^\top Y = (X^\top X)^{-1} X^\top X T^\top = T^\top. \tag{643}$$

When $\operatorname{rank} X < d$, choose a nonzero $v \in \ker X$. For any nonzero $w \in \mathbb{R}^m$, define $K = wv^\top$. Then $XK^\top = 0$, so T and $T + K$ agree on every selected anchor while differing on v. □

The second clause is important. A fitted bridge can have zero training residual while remaining arbitrary in precisely the direction used by a repair.

Proposition 40.2 (Ridge bias under a full-rank design)**.** *Under the hypotheses of Theorem 40.1, the ridge estimate satisfies*

$$\widehat{T}^\top - T^\top = -\lambda(X^\top X + \lambda I)^{-1} T^\top, \tag{644}$$

and therefore

$$\left\|\widehat{T} - T\right\| \le \frac{\lambda}{\sigma_{\min}(X)^2 + \lambda} \|T\|. \tag{645}$$

Thus an anchor policy should do more than obtain formal rank. It should enlarge the smallest singular value of the source design.

40.3 Conditional-packet rank obstruction

Write the source state as

$$h_s(x) = A \begin{bmatrix} u(x) \\ s(x)v(x) \end{bmatrix}, \tag{646}$$

where $u, v \in \mathbb{R}^8$ and A is invertible. Let N_1 be the number of affected anchors among k selections.

Proposition 40.3 (Affected-anchor necessity)**.** *If $N_1 < 8$, then the selected source design has rank strictly less than 16. Under generic continuous u and v, $k = 16$, and $N_1 \ge 8$, the selected design has rank 16 almost surely.*

Proof. Only affected rows carry nonzero coordinates in the second eight-dimensional packet. Hence those coordinates contribute rank at most N_1. The base packet contributes rank at most eight, so

$$\operatorname{rank} X \le 8 + N_1 < 16 \tag{647}$$

when $N_1 < 8$. If $N_1 \ge 8$, generic affected rows span the conditional packet and the full set of sixteen generic rows spans the base packet. The determinant of an appropriate 16×16 minor is a nonzero polynomial in the continuous row coordinates, so it vanishes only on a measure-zero set. □

If random anchor membership is Bernoulli with affected probability p, the idealized full-rank probability at $k = 16$ is therefore

$$\Pr(\operatorname{rank} X = 16) = \Pr(N_1 \ge 8) = \sum_{j=8}^{16} \binom{16}{j} p^j (1-p)^{16-j} \tag{648}$$

under the generic model. The observed parent full-rank rate on the primary audit set is 0.03125, consistent with a low-probability affected-count event.

40.4 Packet-balanced pivoted QR

The descendant policy uses the source intervention-defined packet to divide the unlabeled source-side candidate pool into high- and low-packet-activation strata. It selects eight anchors from each

stratum. Within each stratum it applies a deterministic pivoted residual rule:

$$i_j \in \arg\max_{i \notin I_{j-1}} \left\| (I - Q_{j-1}Q_{j-1}^\top) z_i \right\|^2, \tag{649}$$

where z_i is the standardized source activation and Q_{j-1} spans the previously selected residual directions.

Definition 40.4 (Packet-balanced QR policy)**.** For fit budget $k = 16$, define

$$\pi_{\mathrm{PQR}}(H, P_J) = \pi_{\mathrm{QR}}(H_{\mathrm{aff}}, 8) \cup \pi_{\mathrm{QR}}(H_{\mathrm{non}}, 8), \tag{650}$$

where affected and unaffected membership is inferred from the source causal intervention, not from target audit labels.

Corollary 40.5 (Generic full-rank packet-balanced design)**.** *Under the generic model of Theorem 40.3, packet-balanced QR produces a full-rank* 16×16 *source design almost surely whenever both source strata contain at least eight candidates.*

The policy requires no additional target representation queries. It inspects the source-side candidate pool, which is already locally available, and asks the target for the same sixteen paired activations used by the parent.

40.5 Meta-grammar and compiler

The development meta-grammar contains five anchor policies:

$$\Pi_{\mathrm{anc}} = \{\text{random}, \text{balanced random}, \text{QR}, \text{packet QR}, \text{augmented QR}\}. \tag{651}$$

It also contains alignment thresholds

$$\tau \in \{0.01, 0.02, 0.05, 0.10, 0.20, 0.35\} \tag{652}$$

and an optional target-width guard. Every candidate uses the same fit and validation budgets.

For a candidate configuration c, the development compiler computes target-level selected gains and the score

$$S_{\mathrm{dev}}(c) = \bar{g}_c - 0.20 s_c - 2 h_c + 0.02 q_c, \tag{653}$$

where $\bar{g}_c$ is mean protected gain, s_c is the target-level standard deviation, h_c is harmful-target rate, and q_c is positive-target rate. This is a finite engineering objective, not a universal definition of evolvability.

The compiler selected:

$$\begin{aligned} \pi_{\text{anc}} &= \pi_{\text{PQR}}, \\ k_{\text{fit}} &= 16, \\ k_{\text{val}} &= 8, \\ \gamma_{\text{cal}} &= 0.08, \\ \rho_{\text{cal}} &= 0.035, \\ \tau_{\text{align}} &= 0.05, \\ \text{full-rank requirement} &= \text{true}. \end{aligned} \tag{654}$$

The selected development score is 0.263289. The underlying mean selected development gain is 0.296478 with zero harmful targets in the development ledger.

40.6 Lineage-permutation control

A negative control preserves every numerical development outcome but deliberately breaks the receipt binding between each outcome and the policy that produced it. Policy identities are cyclically permuted before compilation. The resulting control selects the random-anchor policy with the descendant's strict gate.

This control matters because a compiler that merely consumes a table of gains should not receive credit unless those gains remain attached to the correct causal modification. On audit, the lineage-permuted descendant obtains mean selected gain 0.004026, compared with 0.153206 for the correctly bound descendant.

41 One-Step Bounded Metaproductivity Experiment

41.1 Frozen experimental design

Each task contains:

- 3,000 source-training examples;
- 800 source causal-probe examples;
- a source-side anchor pool of 1,100 examples;
- 650 calibration examples;
- 1,800 audit examples.

The source repair is discovered and fitted independently for each task. The repair process then receives one foreign target representation and may construct one transported repair.

Development uses ten tasks, five representation families, three target seeds, five anchor policies, and four repeated selections. The compiler therefore consumes 72,000 paired target-representation queries. Audit uses twelve unseen tasks and four target seeds per family.

The primary audit families are:

$$\mathsf{F}_{\text{audit}}^{\text{primary}} = \{\text{dense-linear-20}, \text{block-linear-24}, \text{ill-conditioned-linear-20}\}. \tag{655}$$

The stress families are:

$$\mathsf{F}_{\text{audit}}^{\text{stress}} = \{\text{compressed-10}, \text{saturated-tanh-24}, \text{leaky-ReLU-32}, \text{smooth-polynomial-24}\}\,. \tag{656}$$

The primary families satisfy the linear bridge hypothesis but are absent from primary development. The stress families violate injectivity or global linearity in different ways.

41.2 Compared arms

Arm	Paired queries	Role
Parent random 16	$16 + 8 = 24$	Parent process inherited from the Block 6 anchor protocol
Balanced random 16	$16 + 8 = 24$	Tests whether stratum balance alone explains the gain
QR 16	$16 + 8 = 24$	Tests generic deterministic experimental design without explicit packet balance
Descendant packet-QR 16	$16 + 8 = 24$	Sealed lineage-generated descendant
Parent random 32	$32 + 8 = 40$	Double-fit-budget parent control
Lineage-permuted descendant	$16 + 8 = 24$	Breaks causal receipt binding during meta-compilation

The pure QR arm is intentionally strong. It prevents the manuscript from attributing a standard experimental-design effect uniquely to the full SRIC architecture.

41.3 Primary metaproductivity result

On 144 primary foreign targets, the target-level paired difference is

$$\widehat{\Delta}_{\text{meta}} = \widehat{M}_{24}^{\text{for}}(\mathfrak{P}_1) - \widehat{M}_{24}^{\text{for}}(\mathfrak{P}_0) = 0.209508. \tag{657}$$

A target-cluster bootstrap gives the interval

$$[0.194968, 0.223898], \tag{658}$$

and a 100,000-draw sign-flip test gives

$$p = 9.9999 \times 10^{-6}. \tag{659}$$

All 144 target-level paired differences are positive.

Table 27: Primary foreign-target metaproductivity under matched or declared query budgets.

Arm	Mean gain	Commit rate	Full-rank rate	Queries
Parent random 16	0.135212	0.465278	0.03125	24
Balanced random 16	0.318744	0.932292	1.00000	24
QR 16	0.344731	1.000000	1.00000	24
Descendant packet-QR 16	**0.344720**	**1.000000**	**1.00000**	24
Parent random 32	0.316524	0.932292	0.822917	40
Lineage-permuted descendant	0.008784	0.026042	0.03125	24

The child exceeds the double-budget parent while using sixteen fewer paired target queries. This is a 40% reduction relative to the parent-32 deployment query count.

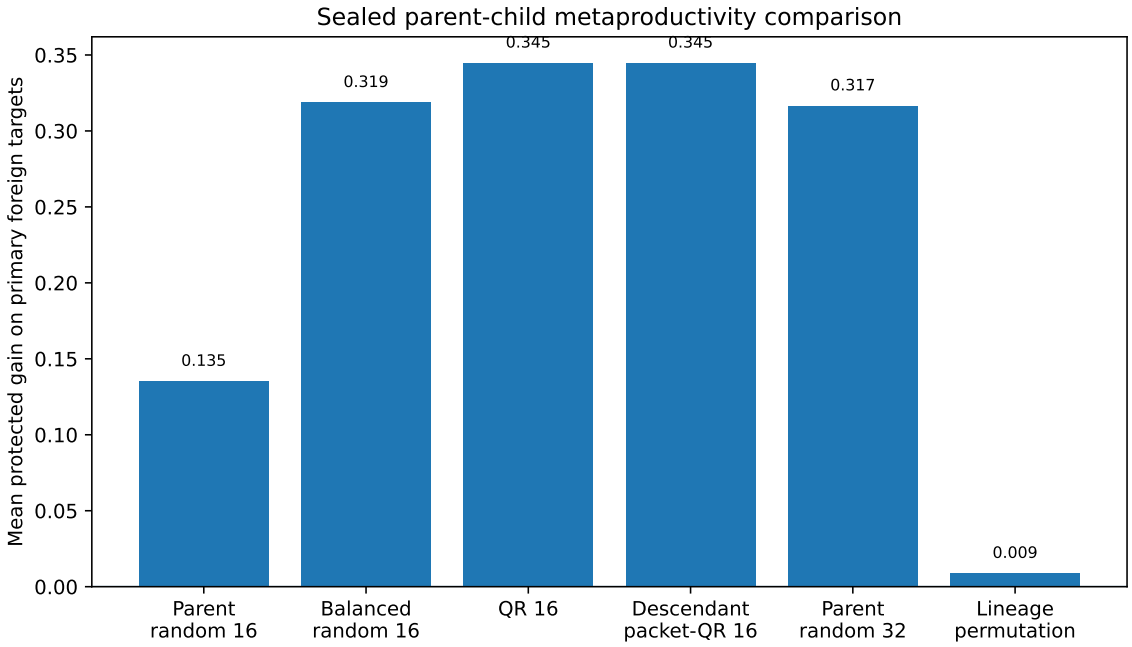


Figure 10: Mean protected gain on the primary foreign target set. Parent and descendant use the same 24 paired queries. The double-budget parent uses 40.

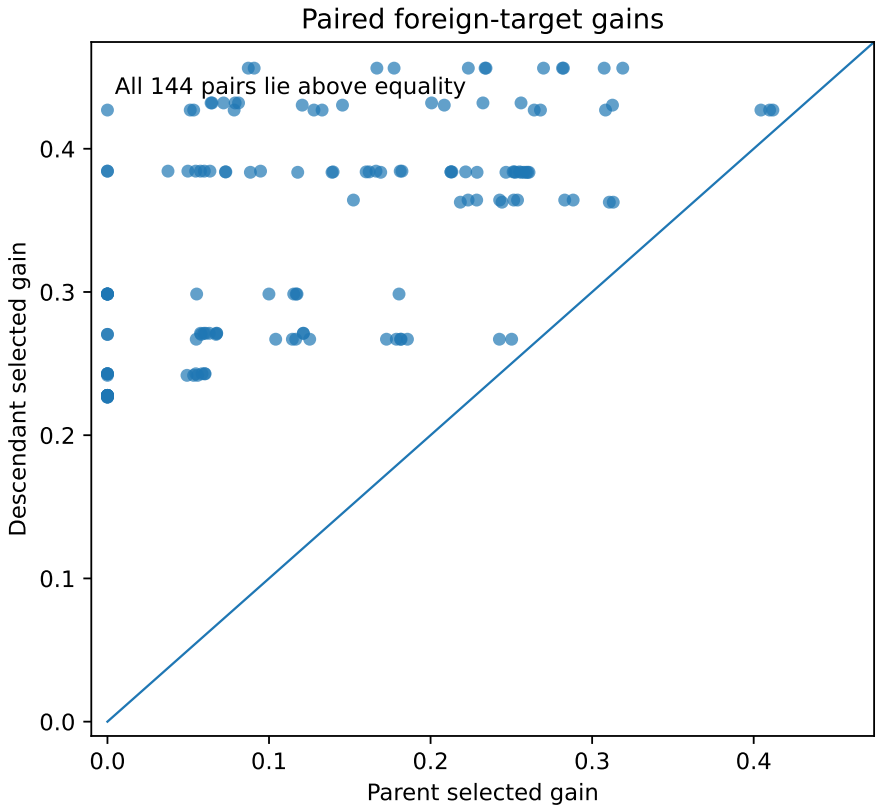


Figure 11: Paired primary-target gains. Every point lies above the equality line.

41.4 Structural metaproductivity

The behavioral gain is accompanied by a structural change. On primary audit targets, the parent source design is full rank in only 3.125% of target-level repeated selections. The descendant is full rank in 100%.

Define the finite structural coverage functional

$$C_{\mathrm{id}}(\mathfrak{P}) = \Pr_{A \sim \mathcal{D}_{\mathrm{for}}} [ankX_A = d_s]\,. \tag{660}$$

Then

$$\widehat{C}_{\mathrm{id}}(\mathfrak{P}_1) - \widehat{C}_{\mathrm{id}}(\mathfrak{P}_0) = 0.96875. \tag{661}$$

This is the measured bridge-identification direction added by the meta-repair.

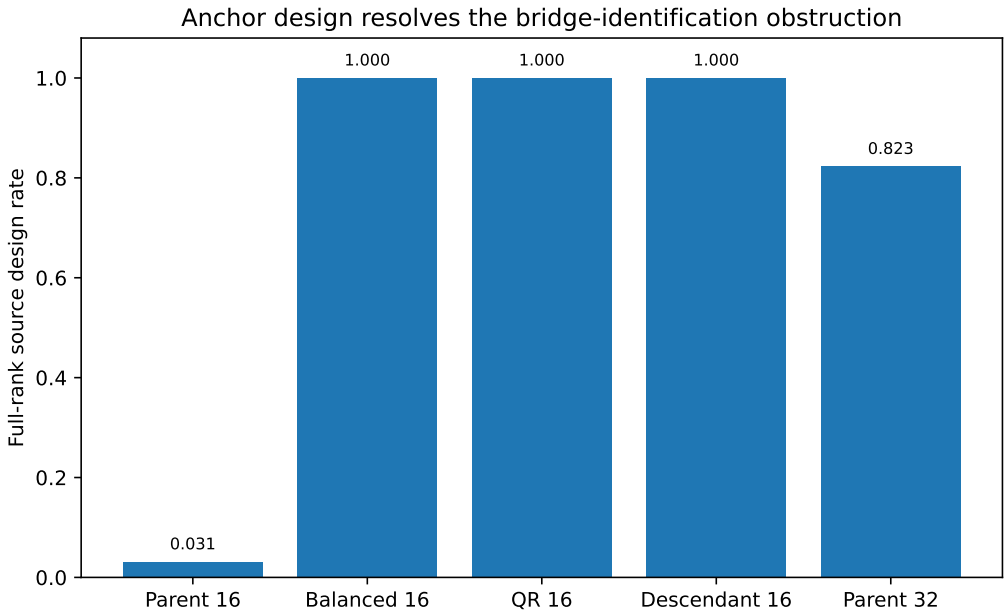


Figure 12: Full-rank source-design rate on primary foreign targets. The meta-repair removes the rank obstruction at the same fit-anchor budget.

The structural result is not merely a post hoc correlate. Theorem 40.1 identifies full rank as the condition that removes linear bridge ambiguity, and Theorem 40.3 explains why the random parent fails under the conditional packet geometry.

41.5 Family-resolved audit results

Table 28: Descendant results by foreign representation family.

Family	Mean gain	Commit rate	Alignment NRMSE	Harm rate
Dense linear 20	0.344887	1.000000	4.10×10^{-4}	0
Block linear 24	0.344887	1.000000	2.91×10^{-4}	0
Ill-conditioned linear 20	0.344387	1.000000	3.04×10^{-5}	0
Compressed 10	0.038278	0.270833	2.87×10^{-4}	0
Saturated tanh 24	0	0	2.260422	0
Leaky-ReLU 32	0	0	4.252823	0
Smooth polynomial 24	0	0	4.652148	0

The compressed family illustrates an important boundary. Noninjectivity obstructs exact repair transport, but it does not imply that every partial repair is useless. The witness commits on 27.08% of compressed targets and obtains positive mean protected gain with no selected harm in this fixture. The correct conclusion is partial correctability under compression, not exact conjugacy.

The nonlinear families are different. Their validation alignment residuals lie far outside the descendant envelope, so the witness abstains everywhere.

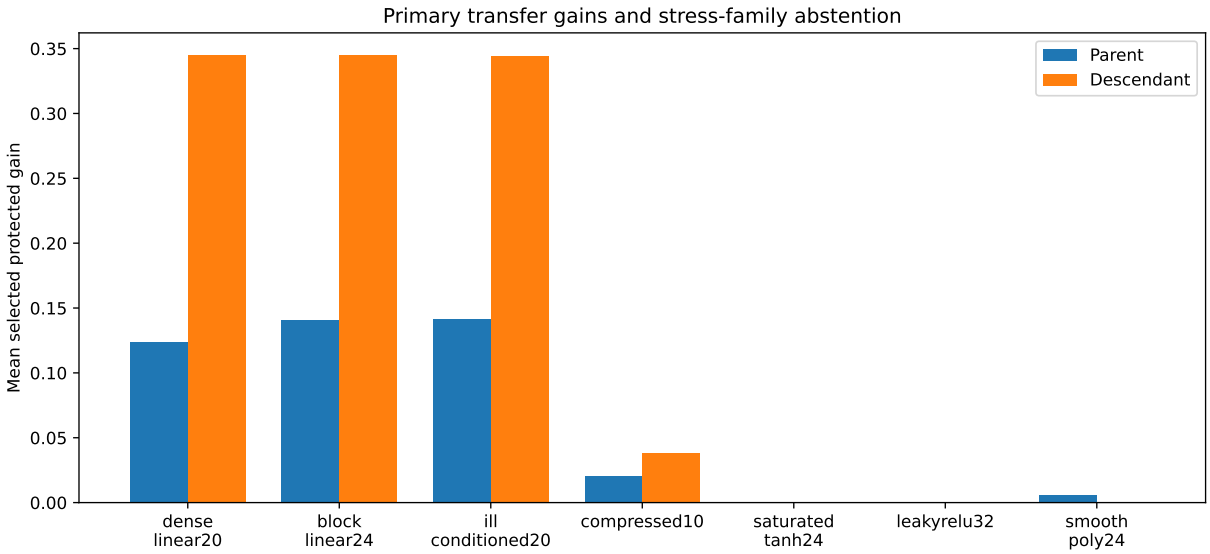


Figure 13: Parent and descendant gain by family. The primary linear families realize the meta-improvement. Strongly nonlinear families remain outside the declared transport class.

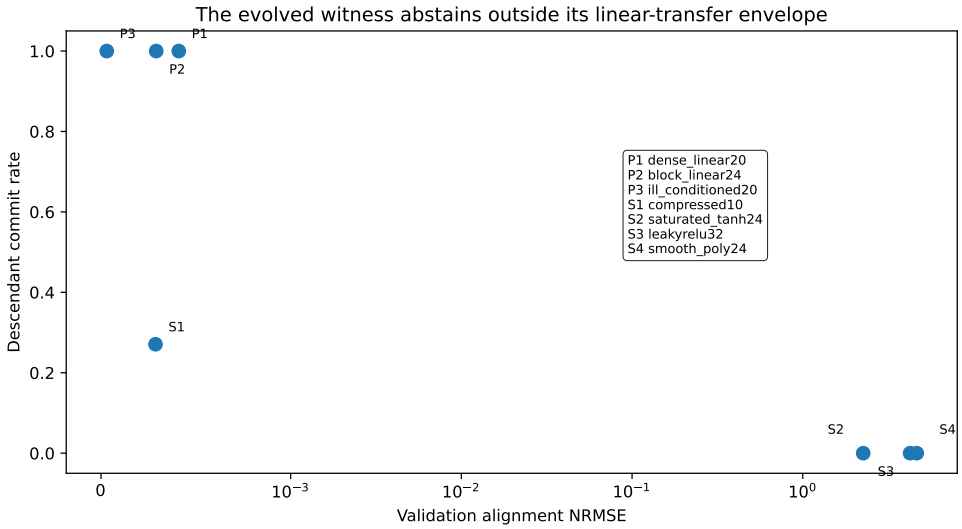


Figure 14: Validation alignment and descendant commitment. The evolved gate abstains on the three strongly nonlinear stress families.

41.6 Query efficiency and meta-compilation cost

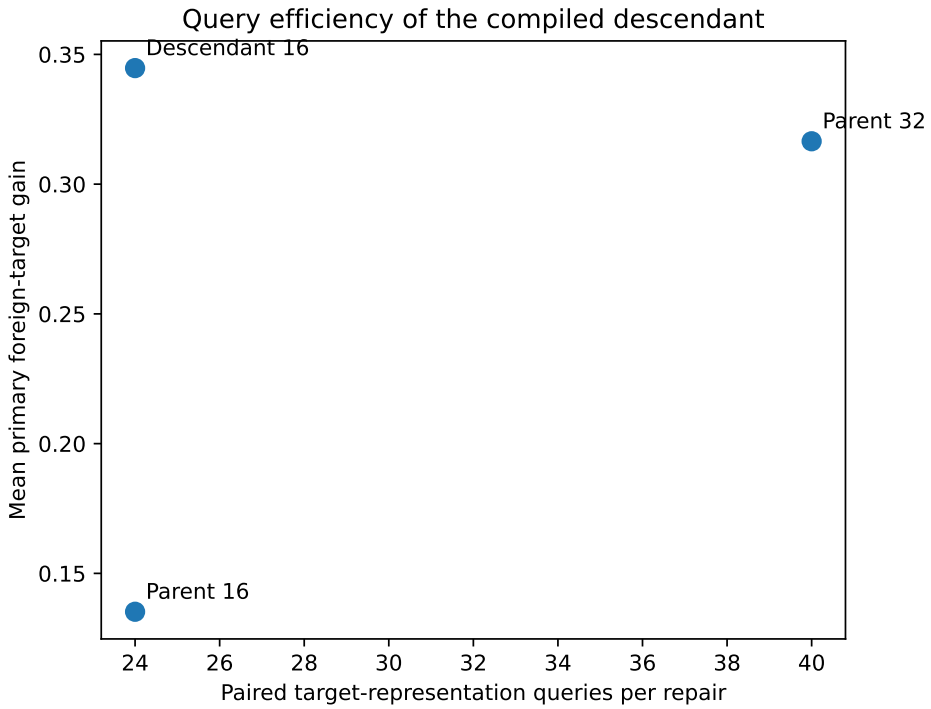


Figure 15: The descendant reaches higher primary gain than the parent with twice the fit-anchor budget.

The deployment comparison alone does not account for the cost of constructing the descendant. The development search consumes 72,000 paired representation queries. Relative to the sixteen paired queries saved against the parent-32 control, the query savings amortize only after

$$\frac{72{,}000}{16} = 4{,}500 \tag{662}$$

deployments. This number is reported because recursive improvement is not useful merely because a descendant is better. Its construction cost must eventually be recovered.

No claim is made that the tiny benchmark is economically efficient. The point is that the accounting is explicit and the deployment improvement is not purchased by silently increasing the per-target query budget.

41.7 Strong-baseline interpretation

Pure QR obtains mean primary gain 0.344731, numerically indistinguishable from packet-balanced QR at 0.344720. Balanced random selection reaches 0.318744.

These controls narrow the mechanistic interpretation:

- most of the gain comes from resolving a classical rank and conditioning problem;
- explicit packet balancing supplies a causal route to guaranteed affected-stratum coverage;

- the experiment does not show that only a Komorebi-specific projector method can solve the bridge-design problem;
- the SRIC contribution is the repair-process framing, the sealed lineage transaction, the structural certificate, the witness, and the claim boundary around the active-design improvement.

42 Bridge-Policy Causal Interpretation and Claim Boundary

42.1 What changed

The parent and descendant differ in exactly two declared fields:

$$\begin{aligned} \pi_{\text{anc}}^{(0)} &= \text{uniform random}, \quad \pi_{\text{anc}}^{(1)} = \text{packet-balanced QR}, \\ \pi_{\text{gate}}^{(0)} &= (0.20, 0.03, 0.35), \quad \pi_{\text{gate}}^{(1)} = (0.08, 0.035, 0.05, \text{full rank}). \end{aligned} \tag{663}$$

The source task model, causal packet discovery, source repair, linear bridge family, pseudoinverse repair transport, target calibration set, protected utility, and audit data remain fixed.

The measured causal chain is therefore:

$$\begin{aligned} &\text{development failure ledger} \\ &\quad \longrightarrow \text{compiled anchor and gate policy} \\ &\quad \longrightarrow \text{full-rank source design} \\ &\quad \longrightarrow \text{identified linear bridge} \\ &\quad \longrightarrow \text{successful transported repair} \\ &\quad \longrightarrow \text{higher foreign-agent metaproductivity}. \end{aligned} \tag{664}$$

The lineage-permutation control breaks the first causal binding and collapses mean gain to 0.004026 over the complete audit set.

42.2 A finite one-step certificate

Conditional Theorem 42.1 (Block 7 one-step meta-improvement certificate)**.** *Assume:*

1. *the development and audit target indices are disjoint;*
2. *the descendant configuration is sealed before audit as in Theorem 39.4;*
3. *parent and descendant receive the matched budget in (637);*
4. *target-level audit differences are exchangeable or otherwise support the declared paired interval;*
5. *the protected evaluator and source repairs are unchanged across the parent-child edge;*
6. *accepted-harm rate is not increased.*

If the paired lower confidence bound for

$$M_{24}^{\text{for}}(\mathfrak{P}_1) - M_{24}^{\text{for}}(\mathfrak{P}_0) \tag{665}$$

is positive and bridge-identification coverage also increases, then the edge $\mathfrak{P}_0 \to \mathfrak{P}_1$ is a bounded one-step recursive meta-improvement relative to the declared audit distribution and fixed meta-grammar.

Block 7 satisfies the finite executable antecedents inside its fixture. The paired lower bound is 0.194968, selected harm is zero, and primary full-rank coverage rises from 0.03125 to 1.

42.3 What the result does not establish

The result does not establish:

- autonomous invention of the meta-grammar;
- autonomous source-code modification;
- repeated descendant improvement over multiple generations;
- improvement of packet discovery itself;
- unpaired or weakly paired transport;
- general nonlinear repair transport;
- improvement of the evaluator;
- open-ended ontology expansion;
- supercritical repair autocatalysis;
- unrestricted safety, consciousness, or sentience.

The compiler is designer specified. It searches five known anchor policies and a finite threshold grid. The task generator, protected metric, repair form, source counterfactual, and correspondence-query interface are also supplied. The descendant improves one declared module within that fixed grammar.

Block 7 result

Block 7 realizes the manuscript's first one-step bounded repair-process meta-improvement. A descendant bridge-estimation policy is compiled from development failures, sealed before audit, and compared with its parent on unseen tasks and unseen primary representation families. Under the same 24 paired-query budget, the descendant improves primary foreign-target gain by 0.209508 with a 95% lower bound of 0.194968, raises full-rank bridge-identification coverage from 0.03125 to 1, and introduces no selected harm in the fixture. The result is finite, linear, and grammar-bounded. It is not open-ended recursive self-improvement.

43 Bridge-Policy Reproducibility and Package

43.1 Reference package

The Block 7 reference package [7] contains:

```
anchor_compiler/
|-- sric_meta/
|   |-- agents.py
|   |-- anchors.py
|   |-- benchmark.py
```

```
|   |-- canonical.py
|   |-- compiler.py
|   |-- data.py
|   |-- plots.py
|   |-- repair.py
|   |-- statistics.py
|   `-- transport.py
|-- frozen/
|   |-- descendant_policy.json
|   |-- compiler_ledger.json
|   `-- preaudit_seal.json
|-- results/
|-- figures/
|-- tests/
|-- run_all.py
|-- Run_SRIC_Block07.ps1
|-- build_manifest.py
|-- verify_package.py
`-- MANIFEST.sha256
```

The package is CPU-only and uses NumPy, Matplotlib, and pytest. BLAS thread counts are fixed to one in the release runner.

43.2 Frozen execution

The canonical release reports:

```
SRIC_BLOCK07_ALL_SUITES=PASS
result_sha256=977879e0fc0aa491af70e7295f6704a92c637b434d76adba1f8b771c9f0d9411
preaudit_seal_sha256=
71100a2e91594028298198970e38f8eff5430fa3c514e50da57feff6eabc7a02
descendant_policy=packet_qr
parent_gain=0.061663
child_gain=0.153206
primary_paired_delta=0.209508
primary_ci_low=0.194968
child_stress_harm=0.000000
```

All eight automated tests pass. The package manifest covers 51 files.

43.3 Frozen claim gates

Table 29: Block 7 primary claim gates.

Gate	Status	Requirement
Pre-audit seal	Pass	Frozen descendant digest and no audit fields in compiler input
Foreign task split	Pass	No development–audit task overlap
Foreign primary families	Pass	No primary representation-family overlap
Matched deployment budget	Pass	Parent and descendant each use 24 paired queries
Paired metaproductivity	Pass	Lower confidence bound above zero
Structural correctability	Pass	Descendant primary full-rank rate at least 0.99
Protected harm	Pass	Zero harmful selected primary and stress targets
Lineage-permutation control	Pass	Misbound compiler output does not match descendant gain
Open-ended recursion	Not tested	No autonomous grammar growth or repeated generations

43.4 Reproduction commands

```
python verify_package.py
OPENBLAS_NUM_THREADS=1 OMP_NUM_THREADS=1 MKL_NUM_THREADS=1 python run_all.py
python -m pytest -q
```

The PowerShell runner performs the same sequence and accepts an explicit Python executable.

44 Meta-Headroom Geometry

Block 7 establishes that a lineage-generated descendant can improve a component of the repair process while the target systems, evaluator, witness, and deployment resources remain fixed. The next question is not merely whether another descendant can obtain another positive score difference. A repeated lineage must determine *where additional improvement remains possible*. If one module has reached its attainable ceiling, continuing to modify that module can produce a false negative about recursive improvement. The relevant bottleneck may have moved elsewhere.

This section therefore resolves the repair process into modules, defines restricted oracle envelopes, allocates remaining metaproductivity headroom, and distinguishes weak descendants from saturated parents.

44.1 Repair-process factorization

Let

$$N = \{H, Q, C, T, S, V, W\} \tag{666}$$

be the repair-process module set, where:

1. H is the causal hypothesis and atlas mechanism;
2. Q is the active probe or evidence-acquisition policy;
3. C is the candidate repair constructor;

4. T is the representation and repair transport mechanism;
5. S is the candidate selector;
6. V is the evaluator;
7. W is the witness, authorization, and rollback mechanism.

A repair process is a typed tuple

$$K = (H, Q, C, T, S, V, W). \tag{667}$$

The factorization is relative to a declared implementation boundary. A system may refine one module into several submodules or combine several modules behind one interface, provided the declared intervention and evidence channels remain explicit.

Fix a protected evaluation context Ξ, a foreign-system distribution $\mathcal{D}$, and a matched resource budget B. Define protected foreign metaproductivity by

$$\mathcal{M}_{\Xi,\mathcal{D},B}(K) = \mathbb{E}_{A\sim\mathcal{D}}\left[J_\Xi\big(K[A;B]\big) - J_\Xi(A)\right], \tag{668}$$

where $K[A;B]$ is the authorized descendant selected by K for target A, or A itself when K abstains.

Remark 44.1 (Module claims are distinct)**.** A child that changes only S establishes a selector-level claim. A child that changes only Q establishes an information-acquisition claim. Neither child enlarges the candidate language unless C or the grammar used by C changes. Likewise, a better transport map does not by itself establish a better evaluator or witness.

44.2 Restricted oracle envelopes

Let $\mathcal{K}$ be the declared finite or compact class of admissible repair processes under the protected context. For $R \subseteq N$, let

$$\mathcal{K}_R(K) = \left\{\widetilde{K} \in \mathcal{K} : \widetilde{K}_i = K_i \text{ for every } i \notin R\right\}. \tag{669}$$

The restricted oracle envelope is

$$F_K(R) = \sup_{\widetilde{K}\in\mathcal{K}_R(K)} \mathcal{M}_{\Xi,\mathcal{D},B}(\widetilde{K}). \tag{670}$$

Thus

$$F_K(\varnothing) = \mathcal{M}(K), \qquad F_K(N) = \sup_{\widetilde{K}\in\mathcal{K}} \mathcal{M}(\widetilde{K}). \tag{671}$$

The total declared headroom is

$$\mathcal{H}(K) = F_K(N) - F_K(\varnothing). \tag{672}$$

The adjective *declared* matters. A larger, previously unavailable meta-grammar may expose additional headroom. Equation (672) is therefore a relative ceiling, not an absolute limit on future system design.

Proposition 44.2 (Monotonicity of restricted envelopes)**.** *If $R \subseteq R' \subseteq N$, then*

$$F_K(R) \le F_K(R'). \tag{673}$$

Proof. Every process admissible under R is admissible under R', so $\mathcal{K}_R(K) \subseteq \mathcal{K}_{R'}(K)$. Taking suprema gives the result. □

44.3 Ordered meta-regret decomposition

Choose an operational ordering

$$r_1 \prec r_2 \prec \cdots \prec r_m, \qquad m = |N|, \tag{674}$$

with nested sets

$$R_0 = \varnothing, \qquad R_j = \{r_1, \ldots, r_j\}. \tag{675}$$

Define the ordered marginal headroom

$$\Delta_j^{\mathrm{ord}}(K) = F_K(R_j) - F_K(R_{j-1}). \tag{676}$$

Theorem 44.3 (Exact ordered meta-regret decomposition)**.** *For every ordering of the repair-process modules,*

$$\mathcal{H}(K) = \sum_{j=1}^{m} \Delta_j^{\mathrm{ord}}(K). \tag{677}$$

Every summand is nonnegative.

Proof. The sum telescopes:

$$\sum_{j=1}^{m} (F_K(R_j) - F_K(R_{j-1})) = F_K(N) - F_K(\varnothing). \tag{678}$$

Nonnegativity follows from Theorem 44.2. □

One useful order is

$$S \prec Q \prec T \prec C \prec H \prec V \prec W. \tag{679}$$

It first asks how much opportunity remains in selection, then evidence acquisition, transport, construction, representation, evaluation, and witness design. Other orders may be preferable in other experiments. The dependence on order records module interaction rather than a defect in the definition.

44.4 Interaction-independent headroom allocation

To remove ordering dependence, average marginal contributions over all module orderings.

Definition 44.4 (Shapley meta-headroom)**.** For $i \in N$, define

$$h_i(K) = \sum_{R \subseteq N \setminus \{i\}} \frac{|R|!(m - |R| - 1)!}{m!} \left[F_K(R \cup \{i\}) - F_K(R) \right]. \tag{680}$$

Theorem 44.5 (Meta-headroom allocation)**.** *For every finite repair-process factorization,*

$$h_i(K) \geq 0 \tag{681}$$

for every module i, and

$$\sum_{i \in N} h_i(K) = \mathcal{H}(K). \tag{682}$$

Proof. For any permutation of N, the marginal contributions along that ordering telescope from $F_K(\varnothing)$ to $F_K(N)$. Equation (680) is the average marginal contribution of module i over all permutations. Summing those averages over i gives the average of the total telescoping difference, which equals $\mathcal{H}(K)$. Nonnegativity follows from monotonicity of F_K. □

Remark 44.6 (Interactions are allocated rather than discarded)**.** If information acquisition and transport have positive synergy, the value of changing both can exceed the sum of their isolated values. Equation (680) allocates that interaction across the participating modules. It does not assume additive independence.

The exact finite companion uses five mutable modules and a monotone oracle envelope with three positive pair interactions. Its total headroom is

$$0.740000, \tag{683}$$

and the computed module allocations sum to the same value within machine precision. The allocation is shown in Figure 16.

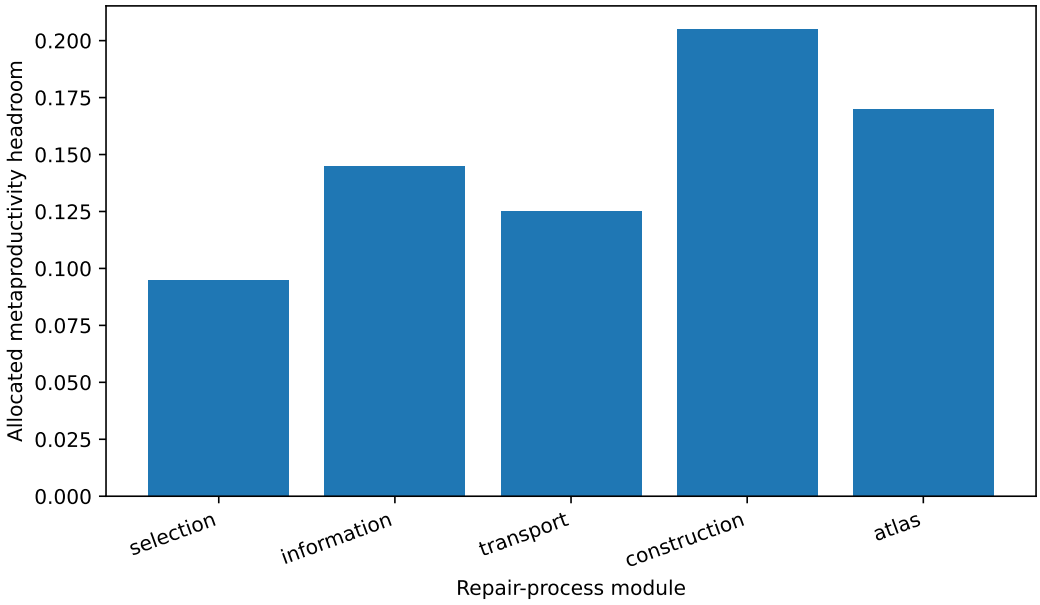


Figure 16: Exact finite allocation of declared metaproductivity headroom across five repair-process modules. Pair interactions are distributed by the permutation-average rule in Theorem 44.4.

44.5 Component saturation and headroom-adjusted gain

Definition 44.7 (ε-saturation)**.** A module set $R \subseteq N$ is ε-saturated at K when

$$F_K(R) - \mathcal{M}(K) \le \varepsilon. \tag{684}$$

Proposition 44.8 (Component ceiling)**.** *If R is ε-saturated, no descendant that changes only modules in R can improve protected metaproductivity by more than ε.*

Proof. Every such descendant lies in $\mathcal{K}_R(K)$ and therefore has metaproductivity at most $F_K(R) \leq \mathcal{M}(K) + \varepsilon$. □

The bridge-policy result from Block 7 reaches full-rank bridge identification on every primary target. Relative to that endpoint, the rank-deficiency component is saturated. A grandchild that repeats the same rank-revealing optimization cannot establish much additional progress. The next useful descendant must encounter a different limitation, such as conditioning, nonlinear misspecification, information loss, or candidate-grammar insufficiency.

Let $K \to K'$ change only modules in R. When $F_K(R) > \mathcal{M}(K)$, define the headroom-adjusted gain

$$\chi_R(K, K') = \frac{\mathcal{M}(K') - \mathcal{M}(K)}{F_K(R) - \mathcal{M}(K)}. \tag{685}$$

For an admissible child inside the declared class,

$$\chi_R(K, K') \leq 1. \tag{686}$$

A small absolute gain with $\chi_R \approx 1$ indicates a nearly saturated parent. A small absolute gain with $\chi_R \approx 0$ indicates that the child captured little of the available opportunity.

When the exact envelope is unknown, certified lower and upper oracle bounds

$$\underline{F}_K(R) \leq F_K(R) \leq \overline{F}_K(R) \tag{687}$$

produce corresponding bounds on χ_R. Audit-oracle arms are finite empirical approximations to this idea, but they must remain inadmissible for selection because they use final audit outcomes.

Warning 44.9 (No-ceiling fallacy)**.** A failed same-module grandchild does not by itself refute repeated recursive improvement. The mutable component may have reached its declared oracle ceiling. A repeated-lineage experiment must report remaining headroom or a saturation certificate before interpreting a null result.

45 Bottleneck Migration and Repeated Lineage Dynamics

A repair lineage should not be expected to improve the same module forever. Better selection exposes candidate limitations. Better evidence acquisition exposes transport mismatch. Better transport exposes missing repair constructors. Better construction can expose evaluator or witness incompleteness. Recursive improvement is therefore a dynamics of moving bottlenecks.

45.1 The meta-regret state

Let the unresolved module-headroom vector be

$$r_t = \begin{bmatrix} h_H(K_t) \\ h_Q(K_t) \\ h_C(K_t) \\ h_T(K_t) \\ h_S(K_t) \\ h_V(K_t) \\ h_W(K_t) \end{bmatrix} \in \mathbb{R}^7_{\geq 0}. \tag{688}$$

An authorized meta-action a changes one or more repair-process modules. In a local finite model, represent its effect by

$$r_{t+1} \leq A_a r_t + e_t, \tag{689}$$

where A_a is a nonnegative response matrix and e_t records estimation error, distribution shift, semantic debt, or newly induced headroom.

The diagonal entries of A_a describe direct contraction. Off-diagonal entries describe bottleneck transfer. For example, a better selector can reduce selection regret while increasing the value of better candidate construction because candidate quality becomes the new limiting factor.

Definition 45.1 (Weighted meta-regret)**.** For $w \in \mathbb{R}^7_{>0}$, define

$$\Phi_w(r) = w^\top r. \tag{690}$$

The weights encode the declared relative value or risk of unresolved module headroom.

45.2 Lyapunov contraction

Theorem 45.2 (Meta-regret Lyapunov contraction)**.** *Suppose there exist $w \in \mathbb{R}^7_{>0}$ and $q \in [0, 1)$ such that for every reachable r_t, the controller can choose an authorized action a_t satisfying*

$$w^\top A_{a_t} r_t \leq q\, w^\top r_t. \tag{691}$$

Then every controlled trajectory satisfying (689) obeys

$$w^\top r_t \leq q^t w^\top r_0 + \sum_{k=0}^{t-1} q^{t-1-k} w^\top e_k. \tag{692}$$

Proof. Equation (689) and nonnegativity of w give

$$w^\top r_{t+1} \leq w^\top A_{a_t} r_t + w^\top e_t \leq q\, w^\top r_t + w^\top e_t. \tag{693}$$

Repeated substitution proves (692). □

Corollary 45.3 (Fixed-grammar convergence)**.** *If the grammar remains fixed and*

$$\sum_{t=0}^{\infty} w^\top e_t < \infty, \tag{694}$$

then $w^\top r_t \to 0$ *whenever the tail errors vanish under the convolution in* (692).

The theorem does not require one module to contract at every generation. It requires a controller that can choose a sufficiently useful action for the current state. The target of improvement may migrate among modules.

45.3 Greedy dominant-bottleneck contraction

A simpler sufficient condition uses the largest unresolved component.

Proposition 45.4 (Greedy bottleneck contraction)**.** *Let* $r_t \in \mathbb{R}^m_{\geq 0}$ *and*

$$\Phi_t = \mathbf{1}^\top r_t. \tag{695}$$

Suppose the controller chooses i_t *with*

$$r_{i_t,t} \geq \frac{\Phi_t}{m}, \tag{696}$$

contracts that component by $q \in [0,1)$*, and permits total collateral increase at most* ε_t*. Then*

$$\Phi_{t+1} \leq \left(1 - \frac{1-q}{m}\right)\Phi_t + \varepsilon_t. \tag{697}$$

Proof. The direct reduction is at least $(1-q)r_{i_t,t} \geq (1-q)\Phi_t/m$. Adding collateral error ε_t gives the bound. □

This proposition makes the role of bottleneck detection explicit. A process that repeatedly modifies an already small component need not contract total meta-regret.

45.4 Saturation-aware controller

Define a module status map

$$\sigma_t(i) \in \{\texttt{active}, \texttt{saturated}, \texttt{unidentified}, \texttt{protected}\}. \tag{698}$$

An admissible bottleneck controller should:

1. estimate headroom for active modules;
2. refuse large gain claims for saturated modules;
3. request new evidence for unidentified modules;
4. avoid changing protected modules inside the current transaction;
5. trigger grammar expansion when all active modules are saturated while persistent obstruction remains.

Protocol 45.5 (Saturation-aware meta-controller)**.** At generation t:

1. compute lower and upper headroom bounds for every currently mutable module coalition;
2. select the coalition with largest certified lower headroom per unit resource;
3. compile candidate descendants that target only that coalition;

4. seal the candidate set and final audit endpoints;
5. commit only a descendant with a positive fresh lower metaproductivity bound;
6. mark the coalition saturated when its upper headroom falls below the declared tolerance;
7. if every current coalition is saturated and foreign obstruction remains, produce a grammar-boundary certificate.

45.5 Conservative grammar discovery event

A grammar extension can reveal error that did not exist as a coordinate in the old state. Let

$$E_t : \mathbb{R}^{m_t} \hookrightarrow \mathbb{R}^{m_{t+1}} \tag{699}$$

be the conservative embedding of old meta-regret coordinates. Decompose

$$r_{t+1} = E_t \widetilde{r}_{t+1} + n_{t+1}, \qquad n_{t+1} \perp \operatorname{im} E_t. \tag{700}$$

Define old-coordinate repair progress by

$$P_t^{\text{repair}} = \|r_t\|^2 - \|\widetilde{r}_{t+1}\|^2, \tag{701}$$

and epistemic discovery by

$$P_t^{\text{discovery}} = \|n_{t+1}\|^2. \tag{702}$$

Both quantities may be positive. The total represented regret may increase because the lineage learned to represent a previously invisible limitation.

Warning 45.6 (Discovery is not regression)**.** A conservative extension should not be rejected merely because the norm of the expanded regret vector exceeds the norm of the old vector. The correct comparison separates progress on old coordinates from the validated birth of a new coordinate.

The finite Block 8 companion contracts several headroom components, then introduces one new nonlinear-grammar coordinate. The explicit expansion is the only increase in total represented headroom. Subsequent actions reduce the new coordinate. The trajectory appears in Figure 17.

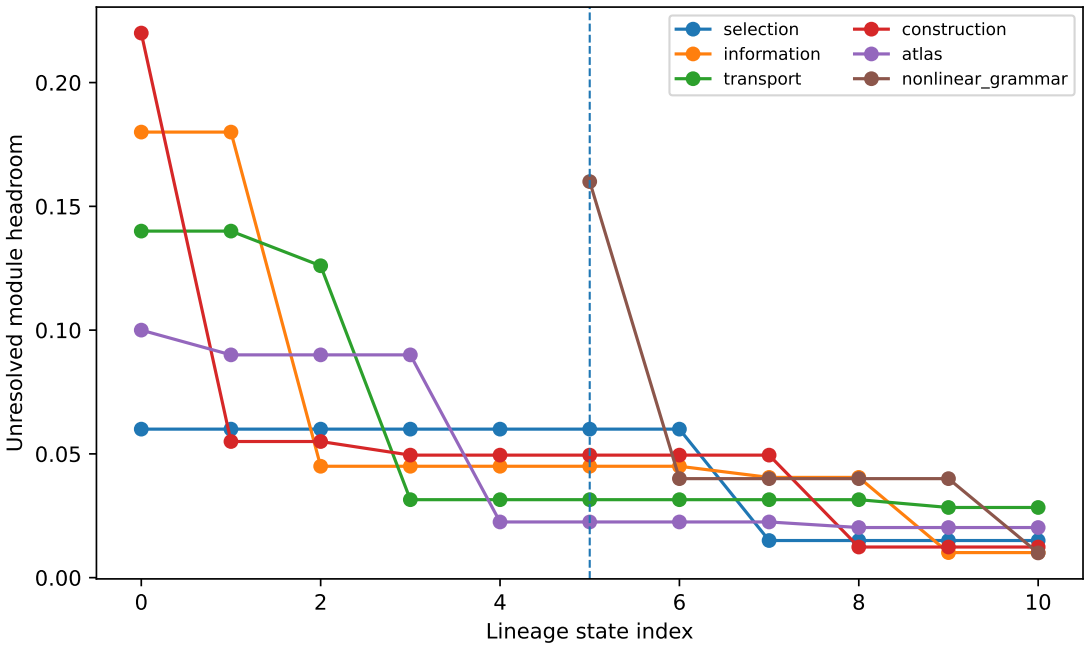


Figure 17: Finite bottleneck-migration fixture. The controller reduces the currently dominant headroom. The dashed line marks a conservative grammar-discovery event that adds a new nonlinear coordinate rather than silently reinterpreting an old one.

45.6 Conservative lineage dominance

Definition 45.7 (Conservative lineage dominance)**.** Write

$$L_t \preceq_{\mathrm{cons}} L_{t+1} \tag{703}$$

when:

1. the old grammar, evidence ledger, and proof objects embed into the new lineage state;
2. every old protected interpretation remains recoverable within the declared semantic-debt bound;
3. old authorized behaviors replay within declared tolerances;
4. the new correction image contains the transported old correction image;
5. at least one new externally validated distinction, repair schema, or correction direction is added, or one old module-headroom coordinate is strictly reduced;
6. foreign metaproductivity does not decrease beyond the accepted uncertainty and regression budgets.

This is a partial order only after quotienting by the declared equivalence of lineage states. It is intentionally stricter than an increase in one scalar score.

Definition 45.8 (Open-set ascending lineage)**.** A lineage is conservatively ascending when

$$L_0 \preceq_{\mathrm{cons}} L_1 \preceq_{\mathrm{cons}} L_2 \preceq_{\mathrm{cons}} \cdots . \tag{704}$$

It is structurally open-set when infinitely many edges add a foreign-task-relevant correction direction or validated distinction not representable in the prior grammar.

45.7 Repair process and repair compiler

The mechanism that repairs agents should be distinguished from the mechanism that compiles a better repair process.

Let K_t be the active repair process and let Γ_t be the lineage compiler. Write

$$K_{t+1} = \Gamma_t(K_t, \mathcal{E}_t, \mathcal{A}_t), \tag{705}$$

where $\mathcal{E}_t$ is the available evidence and $\mathcal{A}_t$ is the executable archive.

There are three regimes:

1. **fixed-compiler iteration**: $\Gamma_t = \Gamma$ while the compiled processes K_t change;
2. **compiler improvement**: Γ_t itself changes and improves on foreign parent processes;
3. **guarded reflective closure**: Γ_t is a typed editable component of the lineage, but the witness validating a compiler change is not modified in the same transaction.

Define compiler metaproductivity by

$$\mathcal{C}(\Gamma) = \mathbb{E}_{K \sim \mathcal{D}_{\text{parent}}} \left[\mathcal{M}(\Gamma(K, \mathcal{E}_K)) - \mathcal{M}(K) \right]. \tag{706}$$

A stronger recursive claim requires

$$\mathcal{C}(\Gamma_{t+1}) > \mathcal{C}(\Gamma_t) \tag{707}$$

on sealed foreign parent processes under matched compilation resources.

Block 7 improves components inside K. It does not yet establish improvement of the outer compiler Γ.

46 Persistent Obstruction Across Grammar Growth

The original obstruction theory is relative to one authorized correction map. Open-set recursive improvement requires a grammar-indexed sequence of maps. This section tracks which defect components become correctable, which persist, and which merely disappear because a new representation forgot them.

46.1 Nested correction images

Let V be a finite-dimensional defect Hilbert space. Suppose a conservative grammar chain gives nested closed authorized correction images

$$I_0 \subseteq I_1 \subseteq I_2 \subseteq \cdots \subseteq V, \qquad I_t = \operatorname{im} D_t. \tag{708}$$

For $r \in V$, define unresolved obstruction magnitude

$$\eta_t(r) = \operatorname{dist}(-r, I_t) = \left\| P_{I_t^\perp} r \right\|. \tag{709}$$

Theorem 46.1 (Monotone obstruction convergence)**.** *For every defect r, the sequence $\eta_t(r)$ is nonincreasing and*

$$\eta_t(r) \longrightarrow \operatorname{dist}\left(-r, \overline{\bigcup_{t\geq 0} I_t}\right). \tag{710}$$

Proof. Nestedness gives $I_t \subseteq I_{t+1}$, so distance to I_{t+1} cannot exceed distance to I_t. Let

$$I_\infty = \overline{\bigcup_t I_t}. \tag{711}$$

The projections $P_{I_t} r$ converge strongly to $P_{I_\infty} r$ in finite dimension, so the orthogonal residual norms converge to the distance from $-r$ to I_∞. □

46.2 Exact newly correctable energy

Theorem 46.2 (Newly correctable energy identity)**.** *For nested correction images,*

$$\eta_t(r)^2 - \eta_{t+1}(r)^2 = \left\| P_{I_{t+1} \cap I_t^\perp} r \right\|^2. \tag{712}$$

Proof. The nested orthogonal decomposition is

$$I_{t+1} = I_t \oplus (I_{t+1} \cap I_t^\perp). \tag{713}$$

Therefore

$$P_{I_{t+1}^\perp} r = P_{I_t^\perp} r - P_{I_{t+1} \cap I_t^\perp} r, \tag{714}$$

and the two terms on the right are orthogonal. The Pythagorean identity gives (712). □

Equation (712) provides exact structural credit. A grammar extension is useful not because it contains more syntax, but because it creates a new response direction with nonzero overlap with held-out obstruction.

For a foreign residual distribution Q, define expected obstruction energy

$$E_t(Q) = \mathbb{E}_{r\sim Q} \eta_t(r)^2. \tag{715}$$

Then

$$E_t(Q) - E_{t+1}(Q) = \mathbb{E}_{r\sim Q} \left\| P_{I_{t+1} \cap I_t^\perp} r \right\|^2. \tag{716}$$

This is the primary structural metaproductivity estimand for grammar growth.

The exact companion starts with one correctable direction and adds directions in decreasing order of foreign obstruction energy. The resulting tower is shown in Figure 18.

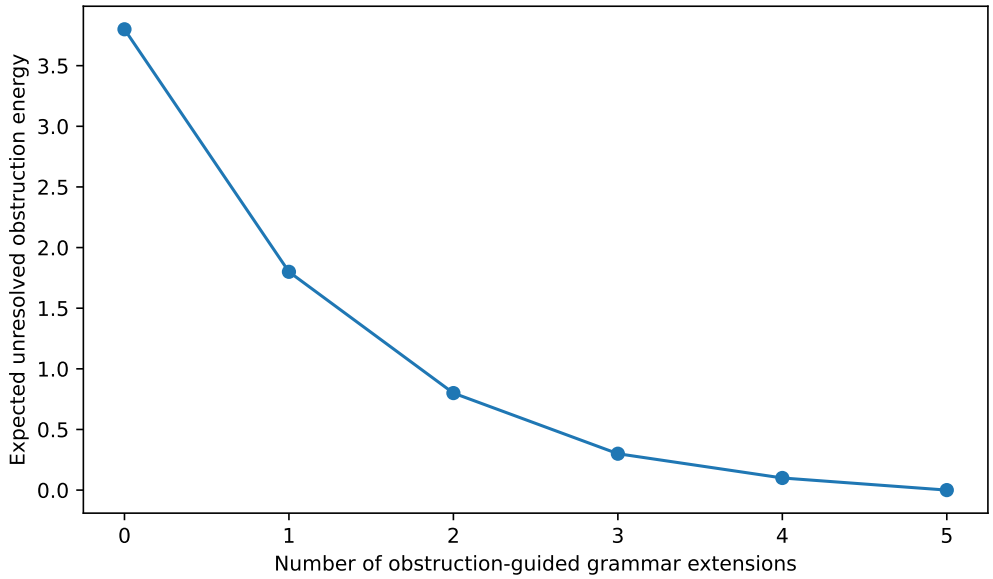


Figure 18: Expected unresolved obstruction energy in the exact grammar-growth fixture. Each extension adds a direction outside the old correction image, and the energy drop equals the newly correctable component from Theorem 46.2.

46.3 Dynamic defect semantics

A grammar extension may also change the defect representation. Let

$$D_t : U_t \to V_t, \tag{717}$$

with transports

$$T_t^U : U_t \to U_{t+1}, \qquad T_t^V : V_t \to V_{t+1}. \tag{718}$$

Assume the exact correction square

$$D_{t+1} T_t^U = T_t^V D_t. \tag{719}$$

Then there is an induced obstruction map

$$\overline{T}_t : \operatorname{coker} D_t \to \operatorname{coker} D_{t+1}, \qquad \overline{T}_t([r]) = [T_t^V r]. \tag{720}$$

Proposition 46.3 (Well-defined obstruction transport)**.** *Equation* (720) *is well defined under* (719).

Proof. If $r - r' = D_t u$, then

$$T_t^V r - T_t^V r' = T_t^V D_t u = D_{t+1} T_t^U u, \tag{721}$$

so $T_t^V r$ and $T_t^V r'$ define the same class in $\operatorname{coker} D_{t+1}$. □

The sequence

$$\operatorname{coker} D_0 \xrightarrow{\overline{T}_0} \operatorname{coker} D_1 \xrightarrow{\overline{T}_1} \operatorname{coker} D_2 \longrightarrow \cdots \tag{722}$$

is the obstruction persistence module of the grammar lineage.

46.4 Repair death and forgetting death

An obstruction class can map to zero for two different reasons.

Definition 46.4 (Repair death)**.** A class $[r] \in \operatorname{coker} D_t$ undergoes repair death at $t+1$ when

$$\overline{T}_t([r]) = 0, \tag{723}$$

there exists an authorized $u \in U_{t+1}$ satisfying

$$D_{t+1}u = -T_t^V r, \tag{724}$$

and the protected defect transport obeys a lower norm bound

$$\left\| T_t^V r \right\| \geq \kappa \left\| r \right\| \tag{725}$$

for a declared $\kappa > 0$.

Definition 46.5 (Forgetting death)**.** A class undergoes forgetting death when its transported representative becomes small or zero because T_t^V collapses the protected distinction, without an authorized correction receipt accounting for the disappearance.

Warning 46.6 (Zero cokernel class is not sufficient)**.** The equation $\overline{T}_t([r]) = 0$ does not by itself certify improvement. Information-losing transport can annihilate a defect rather than repair it. Protected obstruction faithfulness must be checked separately.

Let $Q_t \subseteq V_t$ be a protected obstruction subspace. A dynamic grammar transport is κ-faithful on Q_t when

$$\left\| T_t^V q \right\| \geq \kappa \left\| q \right\| \qquad \text{for all } q \in Q_t. \tag{726}$$

This extends the obstruction-faithfulness requirement from one bridge to an entire grammar lineage.

46.5 Approximate grammar transport and semantic debt

Suppose the correction square is approximate:

$$\left\| D_{t+1} T_t^U - T_t^V D_t \right\| \leq \varepsilon_t^D. \tag{727}$$

Suppose protected defect meanings are reconstructed with error at most ε_t^V. Define cumulative dynamic semantic debt

$$\Delta_{m,n} = \sum_{t=m}^{n-1} \left(\varepsilon_t^D + \varepsilon_t^V \right). \tag{728}$$

Under nonexpansive transports, old correction and defect relations drift by at most the corresponding telescoping debt. If

$$\sum_t (\varepsilon_t^D + \varepsilon_t^V) < \infty, \tag{729}$$

old protected meanings form Cauchy trajectories along the grammar chain.

Definition 46.7 (Persistent protected obstruction)**.** A protected class is persistent when every sufficiently late transported representative remains above a declared norm margin and outside the authorized correction image by a declared residual margin.

Persistent obstruction is the correct trigger for grammar expansion. A transient residual caused by estimator noise should not automatically create a new ontology.

47 Obstruction-Guided Generator and Grammar Discovery

The previous section measures structural progress once a larger correction image has been supplied. This section turns the measurement into a construction rule. The central idea is simple: search for new repair generators whose response aligns with the current obstruction, not merely with high-variance state directions.

47.1 One-generator obstruction reduction

Let $I = \operatorname{im} D \subseteq V$ be the current authorized correction image and let

$$z = P_{I^\perp} r \tag{730}$$

be the unresolved component of defect r. Suppose a candidate grammar extension contributes a response vector $v \in V$. Only its component outside the old image is structurally new:

$$\bar{v} = P_{I^\perp} v. \tag{731}$$

The expanded image is

$$I' = I + \operatorname{span}\{v\}. \tag{732}$$

Theorem 47.1 (One-generator obstruction reduction)**.** *If $\bar{v} \neq 0$, then*

$$\eta_{I'}(r)^2 = \eta_I(r)^2 - \frac{|\langle z, \bar{v}\rangle|^2}{\|\bar{v}\|^2}. \tag{733}$$

If $\bar{v} = 0$, the extension introduces no new first-order correctability.

Proof. Because $\bar{v} \in I^\perp$,

$$I' = I \oplus \operatorname{span}\{\bar{v}\}. \tag{734}$$

The new residual is obtained by subtracting the orthogonal projection of z onto $\bar{v}$. The squared projection norm is the second term in (733). If $\bar{v} = 0$, then $v \in I$ and $I' = I$. □

Equation (733) gives a direct acquisition score. For a candidate family $\mathcal{V}_{\text{cand}}$, choose

$$v^\star \in \arg\max_{v \in \mathcal{V}_{\text{cand}}} \frac{|\langle z, P_{I^\perp} v\rangle|^2}{\|P_{I^\perp} v\|^2 + \lambda c(v) + \beta \operatorname{Risk}(v)}, \tag{735}$$

where $c(v)$ is implementation cost and $\operatorname{Risk}(v)$ is a protected-risk estimate.

A candidate is not accepted from the score alone. The response direction must be generated by an executable typed repair schema, survive witness validation, and reduce foreign obstruction energy.

47.2 Distributional rank-k extension

Let foreign residuals follow distribution Q. Define the obstruction covariance operator

$$\Sigma_{\mathrm{ob}} = \mathbb{E}_{r\sim Q}\left[P_{I^\perp} r\, r^\top P_{I^\perp}\right]. \tag{736}$$

Let $W \subseteq I^\perp$ be a k-dimensional new response subspace.

Theorem 47.2 (Optimal rank-k obstruction extension)**.** *The subspace W minimizing expected unresolved obstruction energy*

$$\mathbb{E}_{r\sim Q}\left\|P_{(I\oplus W)^\perp} r\right\|^2 \tag{737}$$

is the span of the top k eigenvectors of Σ_{ob}. The minimum residual energy is the sum of the remaining eigenvalues.

Proof. For $W \subseteq I^\perp$,

$$\mathbb{E}\left\|P_{(I\oplus W)^\perp} r\right\|^2 = \mathrm{tr}(\Sigma_{\mathrm{ob}}) - \mathrm{tr}(P_W \Sigma_{\mathrm{ob}}). \tag{738}$$

The Ky Fan variational principle maximizes $\mathrm{tr}(P_W\Sigma_{\mathrm{ob}})$ over rank-k orthogonal projectors by choosing the top k eigenspace. □

Remark 47.3 (Where spectral analysis belongs)**.** Block 6 showed that PCA of raw hidden activations can miss the causal repair packet. Theorem 47.2 does not contradict that result. The spectral decomposition is applied after causal intervention and obstruction projection. The relevant covariance is the covariance of unresolved authorized defects, not the unconditional covariance of hidden states.

The resulting pipeline is

$$\text{causal intervention} \longrightarrow \text{authorized obstruction residual} \longrightarrow \Sigma_{\mathrm{ob}} \longrightarrow \text{new response directions}. \tag{739}$$

47.3 Causal credit for a new generator

A new generator receives structural credit only if all of the following hold:

1. its response has a nonzero component outside the old correction image;
2. removing the generator restores the old obstruction energy or eliminates the new foreign gain;
3. an equal-complexity random or generic generator does not reproduce the same effect;
4. the generator does not obtain gain by changing the evaluator or deleting the defect representation;
5. old authorized repairs and protected distinctions continue to replay.

Define the causal generator contribution

$$\Delta_{\mathrm{gen}} = E_Q[\eta_{\mathrm{without}}(r)^2] - E_Q[\eta_{\mathrm{with}}(r)^2]. \tag{740}$$

The contribution is admissible only when a matched ablation isolates the generator while holding the remaining grammar, evidence, and selection budget fixed.

47.4 Repair-conflict grammar inadequacy

Let X be a finite set of repair episodes. The current grammar represents an episode through

$$\phi_{\mathcal{G}} : X \to Z_{\mathcal{G}}. \tag{741}$$

Let $\mathcal{A}^\star(x)$ be the set of externally validated authorized repairs for episode x.

For $z \in Z_{\mathcal{G}}$, define the grammar cell

$$C_z = \{x \in X : \phi_{\mathcal{G}}(x) = z\}. \tag{742}$$

Definition 47.4 (Repair-consistent cell)**.** A cell is repair consistent when

$$\bigcap_{x \in C_z} \mathcal{A}^\star(x) \neq \varnothing. \tag{743}$$

It is inconsistent when no single authorized repair works for every episode that the grammar treats as equivalent.

Definition 47.5 (Minimal repair conflict)**.** A subset $S \subseteq C_z$ is a minimal repair conflict when

$$\bigcap_{x \in S} \mathcal{A}^\star(x) = \varnothing, \tag{744}$$

but every proper subset has a nonempty repair intersection.

The minimal conflicts form a finite hypergraph $\mathfrak{H}_{\mathcal{G}}$. A candidate feature or probe

$$f : X \to Y_f \tag{745}$$

resolves a conflict S when f is not constant on S.

Appending the feature gives the conservative extension

$$\phi_{\mathcal{G}'}(x) = (\phi_{\mathcal{G}}(x), f(x)). \tag{746}$$

The old representation is recovered by projection onto the first coordinate.

Theorem 47.6 (Finite conservative conflict refinement)**.** *Suppose a finite feature family $\mathcal{F}$ resolves every minimal repair conflict in $\mathfrak{H}_{\mathcal{G}}$. Then iterative addition of any feature that resolves at least one remaining conflict terminates after finitely many steps in a grammar whose every cell is repair consistent.*

Proof. A feature split never merges previously separated episodes. Therefore a resolved conflict cannot reappear inside one cell. Every successful addition removes at least one unresolved hyperedge. Since the finite hypergraph has finitely many hyperedges, the process terminates. If the terminal grammar had an inconsistent cell, that cell would contain a minimal inconsistent subset, contradicting exhaustion of the conflict hypergraph. □

The exact companion begins with episodes that are aliased under an empty feature schema. Two causally relevant binary features resolve every conflict. A nuisance feature resolves none of the remaining repair distinctions and is not selected. The conflict counts are shown in Figure 19.

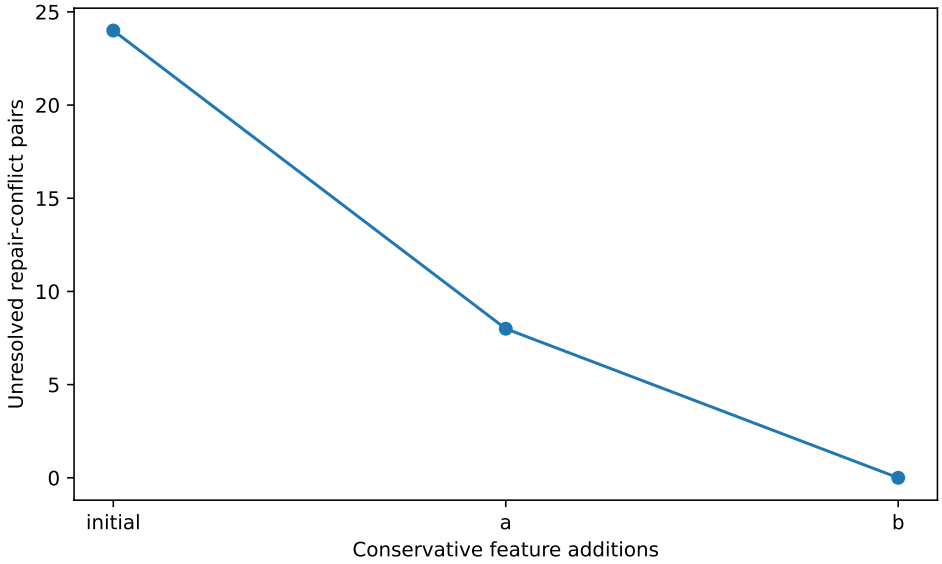


Figure 19: Finite conservative repair-conflict refinement. The selected features split cells with incompatible authorized repairs. The nuisance feature is available but unnecessary.

47.5 Feature acquisition as set cover

Let each candidate feature f cover the conflict set

$$\mathrm{Cov}(f) = \{S \in \mathfrak{H}_{\mathcal{G}} : f \text{ resolves } S\}. \tag{747}$$

Choosing a minimum-cost feature set that resolves all conflicts is a weighted set-cover problem. An exact solver is possible in the finite companion. A scalable lineage may use the greedy score

$$\frac{\text{newly resolved conflicts}}{\text{feature cost} + \text{probe risk}}. \tag{748}$$

The optimization target must remain subordinate to external repair validation. A feature that separates labels but does not improve authorized repair decisions is not a grammar advance.

47.6 Candidate invention as a right inverse

For episode x, let

$$F_x : U_x^{\mathrm{auth}} \to V_x \tag{749}$$

be the post-modification defect map, with $F_x(0) = r_x$. A valid repair satisfies

$$F_x(u_x) = 0. \tag{750}$$

A repair schema is a local section

$$s : x \mapsto u_x \tag{751}$$

that approximately inverts F_x over its declared domain.

In the linear case,

$$F_x(u) = r_x + D_x u, \tag{752}$$

and the canonical minimum-norm repair is

$$s(x) = -D_x^{\dagger} r_x \tag{753}$$

whenever $r_x \in \operatorname{im} D_x$.

Thus candidate invention is not merely sampling more parameter vectors. It is learning or synthesizing an authorized local right inverse of the defect map.

Definition 47.7 (Structurally new repair schema)**.** Let $\mathscr{S}_t$ be the closure of existing repair schemas under the authorized operadic compositions. A schema s_{t+1} is structurally new when

$$s_{t+1} \notin \overline{\mathscr{S}_t} \tag{754}$$

under the declared behavioral equivalence and

$$\operatorname{span} \operatorname{im} s_{t+1} \not\subseteq \operatorname{span} \operatorname{im} \mathscr{S}_t. \tag{755}$$

A structural metaproductivity claim additionally requires

$$E_{t+1}(Q_{\text{foreign}}) < E_t(Q_{\text{foreign}}) \tag{756}$$

and a causal ablation showing that the new schema is responsible for the reduction.

47.7 Finite counterexample-guided repair synthesis

Let $\mathcal{P}_{\mathcal{G}}$ be a finite typed repair-program grammar and let

$$\operatorname{Verify}(p, x) \in \{0, 1\} \tag{757}$$

be a complete verifier on a finite declared episode domain.

Protocol 47.8 (Finite repair synthesis)**.**

1. choose a program $p \in \mathcal{P}_{\mathcal{G}}$ consistent with the current receipt set;
2. search for an episode x with $\operatorname{Verify}(p, x) = 0$;
3. if no counterexample exists, return p with its verification receipt;
4. otherwise add x to the receipt set, eliminate p, and continue.

Theorem 47.9 (Finite synthesis or grammar exhaustion)**.** *Protocol 47.8 terminates after at most $|\mathcal{P}_{\mathcal{G}}|$ rejected programs with either a verified repair program or an exhaustion certificate showing that no program in the current grammar satisfies the finite verifier.*

Proof. Every failed iteration eliminates at least the current program. No eliminated program is

reconsidered. Therefore at most $|\mathcal{P}_{\mathcal{G}}|$ programs can be rejected. If one program survives every verifier query, it is accepted. If all are eliminated, the receipt set certifies grammar-relative exhaustion. □

The exhaustion certificate is the operational bridge from candidate construction to grammar expansion. It does not prove that no repair exists outside the current grammar.

48 Causal Atlas Discovery from Intervention Algebras

The previous theory assumes a defect space and repair strata. A more autonomous lineage must discover a causally meaningful functional cover from intervention responses. Variance alone is insufficient because high-variance directions can be irrelevant to the defect mechanism. The natural finite object is the algebra generated by controlled response operators.

48.1 Intervention-response algebra

Let H be a finite-dimensional complex Hilbert space. Suppose permitted interventions or controlled transitions produce operators

$$R_1, \dots, R_m \in \mathrm{End}(H). \tag{758}$$

These may be finite-difference response maps, local transition Jacobians, or finite Koopman operators on an observable dictionary.

Define the unital adjoint-closed algebra

$$\mathfrak{A} = \mathrm{alg}^* \left(I, R_1, \dots, R_m\right). \tag{759}$$

Its commutant is

$$\mathfrak{A}' = \{X \in \mathrm{End}(H) : XA = AX \text{ for all } A \in \mathfrak{A}\}. \tag{760}$$

The center is

$$Z(\mathfrak{A}) = \mathfrak{A} \cap \mathfrak{A}'. \tag{761}$$

48.2 Finite causal atlas theorem

Theorem 48.1 (Finite causal atlas decomposition)**.** *There exists a unitary identification*

$$H \cong \bigoplus_{\alpha=1}^{p} \left(\mathbb{C}^{m_\alpha} \otimes \mathbb{C}^{n_\alpha}\right) \tag{762}$$

under which

$$\mathfrak{A} \cong \bigoplus_{\alpha=1}^{p} \left(I_{m_\alpha} \otimes M_{n_\alpha}(\mathbb{C})\right). \tag{763}$$

The minimal central projections $P_\alpha \in Z(\mathfrak{A})$ give a canonical coarse causal atlas

$$H = \bigoplus_{\alpha=1}^{p} P_\alpha H. \tag{764}$$

The blocks are unique up to permutation and unitary changes of basis within multiplicity spaces.

Proof. A finite-dimensional unital $*$-algebra is a finite-dimensional C^*-algebra. Decompose its identity into minimal central projections. Each central summand is a simple finite-dimensional C^*-algebra and therefore isomorphic to a full matrix algebra $M_{n_\alpha}(\mathbb{C})$. Its representation on the corresponding invariant subspace is a direct sum of m_α equivalent irreducible representations, yielding (762) and (763). Minimal central projections are intrinsic to the center, giving uniqueness up to ordering and internal unitary gauge. □

Remark 48.2 (Identifiability boundary)**.** The multiplicity factor $\mathbb{C}^{m_\alpha}$ cannot be resolved from $\mathfrak{A}$ alone. Directions inside that factor are interventionally indistinguishable under the declared response family. The ambiguity is a gauge freedom, not a missing numerical optimization step.

The theorem supplies a causal atlas relative to the available interventions. If the intervention family omits a relevant operation, the generated algebra may merge functionally distinct strata.

48.3 Defect-specific cyclic modules

For a current residual $r \in H$, define its cyclic causal module

$$H_r = \operatorname{span}\{Ar : A \in \mathfrak{A}\}. \tag{765}$$

Proposition 48.3 (Minimal causal defect module)**.** *H_r is the smallest $\mathfrak{A}$-invariant subspace containing r. Because $\mathfrak{A}$ is adjoint closed, H_r is reducing and admits an orthogonal projector.*

Proof. The identity belongs to $\mathfrak{A}$, so $r \in H_r$. For $B \in \mathfrak{A}$ and $Ar \in H_r$, one has $BAr \in H_r$, proving invariance. Every invariant subspace containing r contains Ar for every A, hence contains H_r. Adjoint closure implies invariance under every A^*, so the orthogonal complement is invariant as well. □

This produces a two-level representation:

1. the central blocks $P_\alpha H$ form the persistent coarse atlas;
2. H_r is the episode-specific repair packet generated by the current defect under the causal algebra.

The theory therefore does not require one permanent universal workspace subspace.

48.4 Approximate commutant recovery

For $X \in \operatorname{End}(H)$, define commutator energy

$$\mathcal{E}_{\mathrm{com}}(X) = \sum_{j=1}^{m} \left(\|XR_j - R_jX\|_F^2 + \left\|XR_j^* - R_j^*X\right\|_F^2 \right). \tag{766}$$

There is a positive semidefinite linear operator $\mathcal{L}_{\mathrm{com}}$ on the Hilbert space of matrices such that

$$\langle X, \mathcal{L}_{\mathrm{com}} X \rangle_F = \mathcal{E}_{\mathrm{com}}(X). \tag{767}$$

Its kernel is the commutant:

$$\ker \mathcal{L}_{\mathrm{com}} = \mathfrak{A}'. \tag{768}$$

Proposition 48.4 (Spectral-gap stability of the commutant)**.** *Assume the first positive eigenvalue of $\mathcal{L}_{\mathrm{com}}$ is at least $\lambda > 0$. Let $\widehat{\mathcal{L}}_{\mathrm{com}}$ be an empirical estimate satisfying*

$$\left\| \widehat{\mathcal{L}}_{\mathrm{com}} - \mathcal{L}_{\mathrm{com}} \right\| \leq \varepsilon < \lambda/2. \tag{769}$$

Then the spectral projector onto the empirical eigenvalues in $[0, \lambda/2)$ differs from the exact commutant projector by at most a constant multiple of ε/λ.

Proof. The exact zero spectrum is separated from the remaining spectrum by λ. Standard resolvent or sin-theta perturbation bounds for self-adjoint operators give projector error proportional to the perturbation norm divided by the spectral gap. □

The proposition identifies the required empirical condition: stable atlas recovery needs a response-algebra spectral gap. Without one, nearby intervention algebras can support substantially different decompositions.

48.5 Finite algebra fixture

The Block 8 companion uses four response operators on $\mathbb{R}^4$. Two generators act only on the first two coordinates and generate a full 2×2 matrix block. Two act only on the final two coordinates and generate a second full block. The generated algebra has dimension

$$8, \tag{770}$$

the commutant has dimension

$$2, \tag{771}$$

and the recovered central block ranks are

$$(2, 2). \tag{772}$$

A residual supported in the first block has cyclic dimension 2, while a residual with support in both blocks has cyclic dimension 4.

Warning 48.5 (Supplied intervention confound)**.** The finite fixture verifies the algebraic mechanism. It does not show that a large learned agent autonomously discovers the right intervention family. Atlas recovery remains relative to the probes and response operators made available to the system.

48.6 Nonlinear local extension

For nonlinear dynamics on a state manifold M, the response algebra becomes base-point dependent:

$$x \longmapsto \mathfrak{A}_x \subseteq \mathrm{End}(T_x M). \tag{773}$$

Local central projectors define a bundle of tangent strata when their ranks remain locally constant. Transport around a loop can permute or distort the local modules, producing atlas holonomy. A complete nonlinear theory must combine:

1. local intervention algebras;
2. spectral-gap control of approximate central projectors;

3. transition maps between neighboring tangent atlases;
4. descent and holonomy tests on overlaps;
5. repair transport that preserves obstruction classes rather than only state coordinates.

This remains a conditional extension rather than a completed general theorem.

49 Sequential Lineage Certification

One sealed parent-child edge does not provide a lineage-wide statistical guarantee. Every new descendant is chosen adaptively using previous receipts. A valid repeated-lineage theory therefore needs fresh evidence, conditional coverage, and an explicit error budget over the whole archive.

49.1 Filtration and edge-level effects

Let $\mathcal{F}_t$ contain all lineage information available before generation t opens its final audit. The child K_{t+1} and every endpoint used to evaluate it must be $\mathcal{F}_t$-measurable.

Let the true foreign metaproductivity effect be

$$\Delta_t = \mathcal{M}(K_{t+1}) - \mathcal{M}(K_t). \tag{774}$$

Let L_t be a lower confidence bound computed from fresh audit evidence satisfying

$$\Pr(\Delta_t < L_t \mid \mathcal{F}_t) \le \delta_t. \tag{775}$$

Accept the edge only when

$$L_t > 0. \tag{776}$$

Theorem 49.1 (Lineage-wide false-improvement control)**.** *If*

$$\sum_{t=0}^{\infty} \delta_t \le \delta, \tag{777}$$

then

$$\Pr\left(\textit{every accepted lineage edge has } \Delta_t > 0\right) \ge 1 - \delta. \tag{778}$$

Proof. A false accepted edge satisfies $\Delta_t \le 0 < L_t$, so the confidence event in (775) failed. Conditional coverage and the union bound give

$$\Pr(\exists t \text{ false accepted}) \le \sum_t \delta_t \le \delta. \tag{779}$$

□

49.2 Branching archive allocation

For a branching archive, index proposed edges by $e \in E(\Lambda)$. Assign each edge a budget $\delta_e > 0$ with

$$\sum_{e \in E(\Lambda)} \delta_e \le \delta. \tag{780}$$

The same proof yields archive-wide false-acceptance control. The allocation may be adaptive provided the remaining total budget is predictable from the pre-audit filtration.

A simple geometric schedule is

$$\delta_e = \delta 2^{-\iota(e)}, \tag{781}$$

where ι is a prefix-free edge index chosen before the edge audit. More efficient schedules may allocate more error budget to scientifically important or expensive edges.

49.3 Protected regression and semantic-debt budgets

Let R_t be true protected regression and let U_t^R be a conditional upper confidence bound. Require

$$U_t^R \le \beta_t, \qquad \sum_t \beta_t \le B_R. \tag{782}$$

Let d_t be semantic debt incurred by approximate grammar or evaluator transport, with

$$\sum_t d_t \le B_{\mathrm{sem}}. \tag{783}$$

A lineage certificate should therefore report three separate cumulative quantities:

$$\left(\sum_t \delta_t, \sum_t \beta_t, \sum_t d_t \right). \tag{784}$$

A positive gain sequence is not sufficient if its protected regression or semantic debt diverges.

49.4 Repeated-lineage metaproductivity certificate

Definition 49.2 (Repeated-lineage certificate)**.** A finite lineage path

$$K_0 \to K_1 \to \cdots \to K_T \tag{785}$$

has a repeated-lineage metaproductivity certificate when:

1. every edge uses disjoint development, meta-validation, and final audit evidence;
2. candidate, query, compute, and validation resources are fixed or explicitly accounted;
3. every edge has a positive conditional lower metaproductivity bound;
4. cumulative false-acceptance, protected-regression, and semantic-debt budgets remain below declared limits;
5. prior proofs, receipts, and protected distinctions replay after every edge;
6. the archive retains every parent as an executable alternative;
7. every claimed structural edge reduces held-out obstruction energy or adds a validated correction direction.

Proposition 49.3 (Two-edge significance)**.** *A certified path*

$$K_0 \to K_1 \to K_2 \tag{786}$$

with fresh evidence at both edges distinguishes one successful meta-fit from a mechanism that can produce at least one further validated descendant. It does not imply infinitely repeated productive closure.

49.5 Conditional bottleneck-migrating improvement theorem

Conditional Theorem 49.4 (Bottleneck-migrating recursive improvement)**.** *Let*

$$L_t = (K_t, \Gamma_t, \mathcal{G}_t, \mathcal{E}_t, \Lambda_t, V_t, W_t) \tag{787}$$

be a sequence of resolved lineage states. Assume:

1. *every edge is evaluated against externally grounded evidence unavailable before sealing;*
2. *componentwise restricted oracle envelopes are finite or bounded by valid certificates;*
3. *inside each fixed grammar, an authorized meta-controller satisfies the weighted contraction condition of Theorem 45.2;*
4. *saturation is detected before further same-module gains are claimed;*
5. *a grammar extension is accepted only when it resolves a repair conflict or adds a response direction with positive held-out obstruction overlap;*
6. *old evidence, proofs, authorization decisions, and protected meanings embed with summable semantic debt;*
7. *every new repair schema receives causal credit through matched ablation;*
8. *edge-level false-acceptance probabilities are summable;*
9. *proposal, evidence, evaluation, witness, and rollback surfaces are not jointly rewritten in one transaction.*

Then:

1. *weighted meta-regret contracts to the declared error floor during every fixed-grammar interval;*
2. *progress may continue after one module saturates by migrating to another non-saturated module;*
3. *every accepted structural grammar edge strictly reduces held-out obstruction energy in at least one direction unavailable to the old grammar;*
4. *historical protected meanings remain recoverable within the cumulative semantic-debt bound;*
5. *with probability at least $1 - \sum_t \delta_t$, every accepted edge has positive true metaproductivity relative to its declared claim and budget;*
6. *if infinitely many accepted structural edges expose genuinely new foreign-task-relevant correction directions, the lineage forms an infinite conservatively ascending chain of structural repair competence.*

Proof. Item 1 is Theorem 45.2. Item 2 follows from saturation-aware module selection and the existence of another coalition with positive certified headroom. Item 3 follows from Theorems 46.2

and 47.1 and the structural acceptance rule. Item 4 follows from telescoping approximate conservativity. Item 5 is Theorem 49.1. Item 6 follows by repeated application of conservative lineage dominance together with strict structural novelty at infinitely many edges. □

Remark 49.5 (Existence is not guaranteed)**.** The theorem does not assert that a useful meta-action or grammar extension always exists. It states what follows if the lineage can repeatedly find and validate such actions. It does not prove unlimited utility, indefinite acceleration, or unrestricted safety.

50 Block 8 Symbolic Reference Realization

Block 8 is theory first. Its companion implementation is an exact finite realization of the new definitions and theorems. It does not introduce a new learned-agent claim gate.

50.1 Implemented fixtures

The package implements:

1. a monotone restricted oracle envelope and exact Shapley headroom allocation;
2. a bottleneck-migration trace with one explicit conservative grammar-discovery event;
3. a nested obstruction tower with exact newly correctable energy;
4. the one-generator obstruction-reduction identity;
5. repair-conflict refinement with a nuisance-feature negative control;
6. finite counterexample-guided repair synthesis;
7. a finite intervention-response algebra with two central causal blocks;
8. a summable lineage confidence and protected-regression budget fixture.

50.2 Frozen exact results

Fixture	Frozen result
Total declared headroom	0.740000000000
Sum of Shapley module allocations	0.740000000000
Obstruction tower	monotone to zero in the finite six-direction fixture
Conflict refinement	selects features `a`, `b`; rejects nuisance
Finite synthesis	accepts `copy_ab` after four counterexamples
Response algebra dimension	8
Commutant dimension	2
Central block ranks	$(2, 2)$
Cyclic defect dimensions	2 for one block, 4 for mixed support
Cumulative confidence budget	0.01875
Automated tests	8/8 passed

Table 30: Block 8 exact symbolic companion results.

The canonical result object has SHA-256 digest

`39e7fb42fa88815f7184473b8fa3b465a3fc1d1a8fc12b30049dd41c1b9fa34c`.

50.3 Interpretation boundary

Block 8 result

Block 8 formalizes module-specific metaproductivity headroom, bottleneck migration, persistent obstruction across grammar growth, obstruction-guided repair-generator discovery, conflict-grounded grammar refinement, intervention-algebra atlas discovery, and lineage-wide statistical certification. Its executable companion verifies finite exact fixtures for these constructions. It does not establish a new child-grandchild learned-agent result, autonomous open-set grammar growth, spontaneous atlas discovery in a large model, evaluator self-improvement, unrestricted safety, or indefinitely accelerating recursive improvement.

50.4 Next empirical lineage

The next learned or trained experiment should be staged so that the bottleneck genuinely changes:

1. K_0 uses random anchors and a conservative fixed selector;
2. K_1 uses the Block 7 rank-revealing query policy and learned risk-sensitive selector;
3. K_2 faces full-rank but noisy and ill-conditioned bridges and must discover a conditioning-aware design criterion;
4. K_3 faces systematic failure of every global linear bridge and must produce a grammar-inadequacy certificate followed by local or nonlinear transport;
5. K_4 no longer receives a supplied causal packet and must recover a functional atlas from intervention-response operators.

Each edge requires fresh development, validation, and audit evidence. A positive result at one edge may not be reused as evidence for the next.

51 Block 8 Reproducibility and Package

51.1 Package layout

The reference implementation contains:

```
SRIC_v0.9_Block08_Reference_Implementation/
  README.md
  MANIFEST.sha256
  Run_Block08.ps1
  pyproject.toml
  run_block08.py
  verify_package.py
  sric_block08/
```

```
    atlas.py
    bottleneck.py
    certification.py
    grammar.py
    headroom.py
    obstruction.py
    run_all.py
  tests/
  results/
  figures/
```

51.2 Canonical replay

The compact PowerShell call is:

```
python verify_package.py; python run_block08.py; python -m pytest -q
```

The package regenerates the canonical JSON result object and four publication figures. The manifest excludes no executable source used by the result generator.

51.3 Claim boundary

The package is a symbolic reference realization. It demonstrates exact finite identities and implementation consistency. It does not replace a foreign-agent audit, a child-grandchild comparison, or a learned causal-atlas experiment.

52 Repeated Lineage under a Migrating Bottleneck

Block 7 established two one-step modifications of a repair process. The bridge-policy child replaced random paired-anchor acquisition by a rank-revealing design and raised the full-rank bridge-identification rate from 0.03125 to 1. That result also created the next theoretical problem. Once rank failure has been removed, applying the same rank-recovery idea again has no remaining rank headroom. A valid grandchild must identify a different limiting quantity.

Block 9 instantiates the bottleneck-migration theory of Block 8. The frozen Block 7 bridge child becomes the lineage parent

$$K_1. \tag{788}$$

It is evaluated in a fresh environment where every declared primary design is full rank, but the paired target observations are heteroscedastically noisy. The residual difficulty is therefore not whether the bridge equations are identifiable in principle. It is whether the selected anchors make the weighted bridge information sufficiently well conditioned for stable repair transport.

The descendant compiler receives only new development receipts. It diagnoses full-rank saturation, activates a finite conditioning-aware query grammar, selects a grandchild

$$K_2, \tag{789}$$

and seals the resulting policy before the fresh audit tasks and representation families are executed. The primary comparison is

$$\mathcal{M}_{\text{audit}}(K_2) - \mathcal{M}_{\text{audit}}(K_1^{\text{recal}}), \tag{790}$$

where K_1^{recal} preserves the Block 7 rank-revealing query policy but is allowed to recalibrate its witness gate on Block 9 development evidence. This is a stronger parent control than an unchanged replay of the old Block 7 gate.

52.1 Context-indexed repeated lineage

Repeated improvement across fresh environments needs a careful definition because two independently sealed audits need not share the same task distribution.

Definition 52.1 (Context-indexed lineage edge)**.** Let Ξ_t be the protected context for edge $t \to t+1$. A sealed lineage edge

$$K_t \xrightarrow{\Pi_{t+1}} K_{t+1} \tag{791}$$

is γ_t-productive in Ξ_t when its receipt replays, its resource and witness gates pass, and a fresh paired audit certifies

$$\mathcal{M}_{\Xi_t}(K_{t+1}) - \mathcal{M}_{\Xi_t}(K_t) \geq \gamma_t > 0. \tag{792}$$

Definition 52.2 (Finite repeated-lineage certificate)**.** A sequence

$$K_0 \xrightarrow{\Pi_1} K_1 \xrightarrow{\Pi_2} \cdots \xrightarrow{\Pi_m} K_m \tag{793}$$

has an m-edge repeated-lineage certificate when every edge is productive in its own predeclared protected context, every parent receipt is replayed before the child is compiled, no final audit evidence is reused as development evidence for a later edge, and the cumulative error and regression budgets remain within their declared bounds.

Proposition 52.3 (Same-context transitivity)**.** *Suppose every edge is evaluated in one common context Ξ. Then*

$$\mathcal{M}_\Xi(K_m) - \mathcal{M}_\Xi(K_0) = \sum_{t=0}^{m-1} \left(\mathcal{M}_\Xi(K_{t+1}) - \mathcal{M}_\Xi(K_t)\right). \tag{794}$$

Consequently, positive lower bounds on all edges add to a positive lower bound on the complete path.

Proof. The identity is a telescoping sum. □

Warning 52.4 (Fresh contexts do not produce one scalar ordering)**.** If edge t is evaluated in Ξ_t and the contexts differ, the edge effects cannot be added into one global utility difference without an additional transport or dominance relation among contexts. A context-indexed repeated-lineage certificate establishes repeatability of the improvement mechanism. It does not establish that one scalar utility increased by the sum of the reported edge effects.

Block 9 therefore supports a two-edge context-indexed claim. The first bridge-policy edge was sealed and evaluated in Block 7. The second edge replays the exact Block 7 child policy, uses new development evidence, and is evaluated on new task and architecture families.

52.2 Rank saturation and bottleneck migration

Let $X_S \in \mathbb{R}^{n\times d}$ be the source hidden-state design induced by anchor set S. Rank recovery is saturated when

$$\Pr\left(\operatorname{rank} X_S = d\right) \geq 1 - \varepsilon_{\text{rank}} \tag{795}$$

under the current foreign development distribution and the remaining performance gap cannot be explained by rank failure.

Definition 52.5 (Conditioning bottleneck certificate)**.** A rank-revealing parent has a conditioning bottleneck with margin $\kappa > 0$ when:

1. its full-rank rate is at least $1 - \varepsilon_{\text{rank}}$;
2. its weighted information floor is below the declared stability floor;
3. a conditioning-aware policy using the same query budget improves development metaproductivity by at least κ;
4. the improvement survives recalibration of the rank parent and is not obtained only by changing the commit gate.

This certificate separates two statements:

$$\operatorname{rank} X_S = d \tag{796}$$

and

$$\lambda_{\min}(X_S^{\mathsf{T}} W X_S) \gg 0. \tag{797}$$

The first is qualitative identifiability. The second is quantitative stability.

Proposition 52.6 (Rank saturation does not imply stability)**.** *For every $d \geq 2$ and every $\eta > 0$, there exists a full-rank design $X \in \mathbb{R}^{d\times d}$ such that*

$$\lambda_{\min}(X^{\mathsf{T}} X) < \eta. \tag{798}$$

Proof. Take

$$X = \operatorname{diag}(1, \ldots, 1, \sqrt{\eta/2}). \tag{799}$$

Then X is full rank and

$$\lambda_{\min}(X^{\mathsf{T}} X) = \eta/2 < \eta. \tag{800}$$

□

The proposition makes the Block 9 transition necessary rather than cosmetic. Once the full-rank rate reaches one, further progress requires a metric beyond rank.

53 Heteroscedastic Bridge Information

53.1 Weighted paired-anchor model

Let the source hidden state have dimension d and the target hidden state have dimension q. For selected source anchors $x_i \in \mathbb{R}^d$, suppose the observed target anchor is

$$y_i = Tx_i + \varepsilon_i, \qquad \mathbb{E}[\varepsilon_i] = 0, \qquad \mathrm{Cov}(\varepsilon_i) = \sigma_i^2 I_q. \tag{801}$$

The query-quality metadata supplies σ_i before the anchor is selected. This metadata is part of the declared Block 9 environment. It is not discovered autonomously.

Stacking the selected anchors gives

$$Y = XT^{\mathsf{T}} + E, \tag{802}$$

where $X \in \mathbb{R}^{n\times d}$ and $Y \in \mathbb{R}^{n\times q}$. Define

$$W = \mathrm{diag}(\sigma_1^{-2}, \ldots, \sigma_n^{-2}) \tag{803}$$

and the weighted information matrix

$$M_S = X^{\mathsf{T}} W X. \tag{804}$$

Theorem 53.1 (Weighted bridge covariance)**.** *Assume M_S is invertible and the noise rows are independent with covariance $\sigma_i^2 I_q$. The weighted least-squares estimator*

$$\widehat{T}^{\mathsf{T}} = M_S^{-1} X^{\mathsf{T}} W Y \tag{805}$$

is unbiased and satisfies

$$\mathrm{Cov}\left(\mathrm{vec}(\widehat{T}^{\mathsf{T}} - T^{\mathsf{T}}) \mid X\right) = I_q \otimes M_S^{-1}. \tag{806}$$

Consequently,

$$\mathbb{E}\left[\left\|\widehat{T} - T\right\|_F^2 \mid X\right] = q\ \mathrm{tr}(M_S^{-1}). \tag{807}$$

Proof. Write

$$\widehat{T}^{\mathsf{T}} - T^{\mathsf{T}} = M_S^{-1} X^{\mathsf{T}} W E. \tag{808}$$

For one target coordinate, the noise covariance is W^{-1}, so the coefficient covariance is

$$M_S^{-1} X^{\mathsf{T}} W W^{-1} W X M_S^{-1} = M_S^{-1}. \tag{809}$$

The target coordinates are independent and identically weighted, giving the Kronecker covariance. Taking the trace yields the Frobenius identity. □

Corollary 53.2 (Worst-direction variance)**.** *For every $u \in \mathbb{R}^d$ and every target coordinate,*

$$\mathrm{Var}\left(u^{\mathsf{T}}(\widehat{T}^{\mathsf{T}} - T^{\mathsf{T}})e_j \mid X\right) = u^{\mathsf{T}} M_S^{-1} u \le \frac{\|u\|^2}{\lambda_{\min}(M_S)}. \tag{810}$$

The A-optimal quantity $\mathrm{tr}(M_S^{-1})$ controls mean-square bridge error. The E-optimal quantity $\lambda_{\min}(M_S)$ controls the least informed direction. The D-optimal quantity $\log\det M_S$ controls information volume.

53.2 Ridge regularization and reported information

The executable implementation uses

$$M_{S,\lambda} = X^{\mathsf{T}} W X + \lambda I \tag{811}$$

with a small fixed $\lambda > 0$. The reported design diagnostics use source coordinates standardized against the complete candidate pool before applying the reliability weights. Standardization prevents one arbitrary source-coordinate scale from dominating the comparison. The actual bridge fit remains weighted in the original hidden coordinates.

The regularized estimator is biased unless $\lambda = 0$, but it remains stable under weak finite designs. Block 9 reports the regularized information floor, condition number, validation alignment, and a clean transport residual available only to the audit analysis.

53.3 Transport error and repair error

Let the frozen source repair be

$$R_s(h) = h + D_s h + b_s. \tag{812}$$

For an injective bridge T, the exact transported repair is

$$D_t = T D_s T^{\dagger}, \qquad b_t = T b_s. \tag{813}$$

The estimated repair uses $\widehat{T}$.

Proposition 53.3 (Exact repair-operator perturbation identity)**.** *Let*

$$\widehat{D}_t = \widehat{T} D_s \widehat{T}^{\dagger}. \tag{814}$$

Then

$$\widehat{D}_t - D_t = (\widehat{T} - T) D_s \widehat{T}^{\dagger} + T D_s (\widehat{T}^{\dagger} - T^{\dagger}). \tag{815}$$

Hence

$$\left\| \widehat{D}_t - D_t \right\| \le \left\| \widehat{T} - T \right\| \|D_s\| \left\| \widehat{T}^{\dagger} \right\| + \|T\| \|D_s\| \left\| \widehat{T}^{\dagger} - T^{\dagger} \right\|. \tag{816}$$

Proof. Add and subtract $T D_s \widehat{T}^{\dagger}$. □

Conditional Theorem 53.4 (Conditioned repair stability)**.** *Suppose T has full column rank and*

$$\left\| T^{\dagger} \right\| \left\| \widehat{T} - T \right\| < 1. \tag{817}$$

Then there is a finite constant C_T, depending continuously on $\|T\|$ and $\left\| T^{\dagger} \right\|$ inside this perturbation neighborhood, such that

$$\left\| \widehat{D}_t - D_t \right\| \le C_T \|D_s\| \left\| \widehat{T} - T \right\|. \tag{818}$$

If the protected utility readout is L-Lipschitz with respect to the transported repair operator on the declared evaluation set, the loss in repair gain is at most

$$L C_T \|D_s\| \left\| \widehat{T} - T \right\|. \tag{819}$$

The theorem identifies two conditioning layers. The anchor design controls $\widehat{T} - T$. The target bridge condition controls the sensitivity of $T^{\dagger}$ and therefore the amplification of that estimation error. Good source information cannot eliminate a genuinely noninjective or severely ill-conditioned target bridge. The stress families retain those negative cases.

53.4 Known quality and the autonomy boundary

The Block 9 grandchild receives a scalar query-noise estimate before choosing each candidate anchor. The parent receives the same metadata but its frozen packet-QR rule does not use it. This is a legitimate matched-information comparison between query policies.

It is not a result on autonomous uncertainty discovery. A later block must replace the supplied σ_i by an estimated reliability model, charge the repeated probes needed to learn that model, and test calibration under shift.

54 Conditioning-Aware Query Design

54.1 D-optimal marginal information

Let $M \succ 0$ be the current regularized information matrix. Adding candidate anchor x_i with quality weight $w_i = \sigma_i^{-2}$ gives

$$M' = M + w_i x_i x_i^{\mathsf{T}}. \tag{820}$$

Theorem 54.1 (D-optimal marginal gain)**.** *The increase in log information volume is*

$$\log\det M' - \log\det M = \log\left(1 + w_i x_i^{\mathsf{T}} M^{-1} x_i\right). \tag{821}$$

Proof. Apply the matrix determinant lemma to $M + (\sqrt{w_i}x_i)(\sqrt{w_i}x_i)^{\mathsf{T}}$. □

The grandchild greedily maximizes the quantity in Theorem 54.1 while retaining equal affected and unaffected anchor quotas. The quota is inherited from the Block 7 packet-balanced child and prevents the design from abandoning one side of the declared repair contrast.

54.2 A-optimal marginal gain

Proposition 54.2 (A-optimal trace reduction)**.** *Let $v = \sqrt{w_i}x_i$. Then*

$$\operatorname{tr}(M^{-1}) - \operatorname{tr}\left((M + vv^{\mathsf{T}})^{-1}\right) = \frac{v^{\mathsf{T}} M^{-2} v}{1 + v^{\mathsf{T}} M^{-1} v}. \tag{822}$$

Proof. The Sherman–Morrison identity gives

$$(M + vv^{\mathsf{T}})^{-1} = M^{-1} - \frac{M^{-1} v v^{\mathsf{T}} M^{-1}}{1 + v^{\mathsf{T}} M^{-1} v}. \tag{823}$$

Taking traces yields the result. □

The finite compiler compares weighted D-optimal, weighted A-optimal, weighted E-refined, unweighted D-optimal, and inherited packet-QR policies. The selected policy is not hard coded into the claim gate.

54.3 E-optimal exchange

The E-optimal objective is

$$\max_{|S|=k} \lambda_{\min}(M_S). \tag{824}$$

The companion initializes with the weighted D-optimal design, then performs same-stratum exchanges that increase the smallest eigenvalue. This is a finite heuristic rather than a claim of global E-optimality.

54.4 Quality-binding control

A policy may appear successful merely because its label is correlated with a favorable outcome generator. Block 9 therefore includes a quality-permutation control. The grandchild algorithm is left unchanged, but the query-quality values are permuted among the candidate anchors before the design is constructed. Actual observation noise remains attached to the original anchors.

If quality binding is causal, permutation should reduce the weighted information floor and eliminate or weaken the audit gain. This control differs from shuffling final outcomes. It preserves the policy code while breaking the semantic relation between one anchor and its reliability estimate.

54.5 Rank-only and gate-only controls

Two controls isolate the source of the grandchild effect.

Recalibrated rank child. The packet-QR parent is allowed to select its best witness thresholds on the new development environment. This prevents the grandchild from winning only because the unchanged Block 7 alignment threshold is too strict for noisy anchors.

Grandchild gate on rank anchors. The grandchild witness thresholds are applied to the inherited packet-QR anchors. If the grandchild gain came only from a more favorable gate, this arm would reproduce it. Failure of this arm, together with a positive grandchild result, attributes the improvement to the new query design rather than gate liberalization.

54.6 Conditioning diagnosis

The development compiler first evaluates the inherited rank policy. It records:

$$\begin{aligned} &\text{full-rank rate},\\ &\text{weighted minimum information eigenvalue},\\ &\text{weighted condition number},\\ &\text{recalibrated rank-policy gain}. \end{aligned} \tag{825}$$

It activates the conditioning meta-grammar only if rank is saturated and a non-rank policy exceeds the recalibrated rank policy by the predeclared development margin.

Proposition 54.3 (Finite bottleneck diagnosis soundness)**.** *Inside a finite policy grammar, suppose the inherited policy has full-rank rate at least* $1 - \varepsilon_{\text{rank}}$ *and the best conditioning policy has true*

mean protected gain at least κ above the best recalibrated inherited policy. If every empirical policy mean is within η of its true mean and

$$\kappa > 2\eta, \tag{826}$$

then empirical maximization selects a policy whose true gain exceeds the inherited optimum.

Proof. Let $p^\star$ be the best conditioning policy and p_0 the best inherited-policy configuration. Empirical errors of at most η imply

$$\widehat{\mu}(p^\star) \geq \mu(p^\star) - \eta > \mu(p_0) + \eta \geq \widehat{\mu}(p_0). \tag{827}$$

□

The executable release uses frozen development and audit files rather than claiming a prospective population bound from this proposition. The theorem specifies the stronger confirmatory design.

55 Sealed Child-to-Grandchild Protocol

55.1 Lineage objects

The Block 9 lineage uses three policy objects.

1. K_0 is the random-anchor parent from the Block 7 bridge experiment.
2. K_1 is the exact Block 7 packet-balanced pivoted-QR child. Its semantic policy digest is replayed from the Block 7 pre-audit receipt.
3. K_2 is the conditioning-aware grandchild compiled from new Block 9 development receipts.

The primary Block 9 parent control is K_1^{recal}, which preserves the anchor rule of K_1 while recalibrating only the witness gate on the Block 9 development split. The exact K_1 replay remains in the audit as a lineage-integrity and distribution-shift control.

55.2 Fresh task and representation supports

The Block 9 development tasks use the trigger families

$$\{\texttt{bilinear}, \texttt{wave}, \texttt{offset}, \texttt{radial5}\}. \tag{828}$$

The final audit uses

$$\{\texttt{cubic}, \texttt{triple}, \texttt{ratio}, \texttt{annulus}\}. \tag{829}$$

Task identifiers, teacher seeds, and trigger supports are disjoint across the development and final audit files.

The development primary representation families are

$$\{\texttt{dev_dense20}, \texttt{dev_scaled22}, \texttt{dev_block24}\}. \tag{830}$$

The final primary families are

$$\{\texttt{audit_rotated20}, \texttt{audit_anisotropic24}, \texttt{audit_block28}\}. \tag{831}$$

The stress families are severe ill-conditioning, noninjective compression, and nonlinear tanh representations.

No Block 7 final audit rows are used to compile K_2.

55.3 Fixed query and repair budget

Every primary child and grandchild deployment receives:

$$16 \text{ fit anchors} + 8 \text{ validation anchors} = 24 \text{ paired representation queries.} \tag{832}$$

The source repair, bridge estimator, transport formula, target evaluator, target audit size, and protected witness metrics are fixed. The only primary process changes are the anchor-selection rule and the gate compiled with that rule.

A random 32-fit-anchor arm uses

$$32 + 8 = 40 \tag{833}$$

paired queries and serves as a larger-budget control.

55.4 Finite meta-grammar

The compiler searches the anchor policies

$$\begin{aligned}\{&\texttt{packet_qr}, \texttt{unweighted_d_opt}, \texttt{quality_d_opt},\\ &\texttt{quality_a_opt}, \texttt{quality_e_opt}\}\end{aligned} \tag{834}$$

and a finite gate grid over calibration gain, protected regression, debiased validation alignment, and minimum weighted information eigenvalue.

The development objective is a stability- and harm-sensitive mean foreign gain. The policy grammar is externally declared and finite. The grandchild does not invent D-optimal design from an unrestricted program language.

55.5 Commit gate

For one candidate row r, a configuration commits only if

$$\begin{aligned}\widehat{g}_{\mathrm{cal}}(r) &\geq \tau_g,\\ \widehat{R}_{\mathrm{cal}}(r) &\leq \tau_R,\\ \widehat{\varepsilon}_{\mathrm{align}}(r) &\leq \tau_A,\\ \lambda_{\min}(M_r) &\geq \tau_I,\\ \operatorname{rank} X_r &= d.\end{aligned} \tag{835}$$

The selected Block 9 grandchild has

$$(\tau_g, \tau_R, \tau_A, \tau_I) = (0.02, 0.05, 0.20, 2.0). \tag{836}$$

The recalibrated rank child has

$$(0.02, 0.05, 0.55, 0.10). \tag{837}$$

55.6 Seal chronology

The executable chronology is:

1. replay the Block 7 child policy and pre-audit receipt;
2. generate Block 9 development tasks and foreign development representations;
3. evaluate the finite policy and gate grammar;
4. diagnose full-rank saturation and conditioning headroom;
5. serialize the recalibrated child and conditioning grandchild;
6. hash the compiler ledger, source files, policy objects, and ancestor digest;
7. write the Block 9 pre-audit seal with `audit_fields_present=false`;
8. generate the final audit tasks and representation families;
9. execute all primary and control arms;
10. evaluate paired claim gates and package closure.

55.7 Finite-class selection bound

Theorem 55.1 (Finite meta-grammar generalization)**.** *Let $\mathcal{K}$ contain N candidate grandchild configurations. Suppose each target-level protected gain lies in $[-G, G]$ and n independent development targets are used. Then, with probability at least $1-\delta$, every candidate satisfies*

$$\left|\widehat{\mathcal{M}}(K) - \mathcal{M}(K)\right| \le G\sqrt{\frac{2\log(2N/\delta)}{n}}. \tag{838}$$

If $\widehat{K}$ maximizes the empirical mean, then

$$\mathcal{M}(\widehat{K}) \ge \max_{K\in\mathcal{K}} \mathcal{M}(K) - 2G\sqrt{\frac{2\log(2N/\delta)}{n}}. \tag{839}$$

Proof. Hoeffding's inequality for a variable with range length $2G$ gives

$$\Pr\left(\left|\widehat{\mathcal{M}}(K) - \mathcal{M}(K)\right| > \eta\right) \le 2\exp\left(-\frac{n\eta^2}{2G^2}\right). \tag{840}$$

Apply a union bound over N configurations and solve for η. The selected-candidate inequality follows by comparing the empirical maximizer with the true maximizer and paying one deviation term for each. □

The released bootstrap intervals are post-selection evaluations on disjoint frozen audits. They are not presented as an anytime-valid guarantee for an indefinitely expanding meta-grammar.

55.8 Paired grandchild certificate

Let D_i be the grandchild-minus-child protected gain on final foreign target i. The primary finite estimand is

$$\overline{D} = \frac{1}{n}\sum_{i=1}^{n} D_i. \tag{841}$$

The release records a paired bootstrap interval, a sign-flip test, positive and negative pair rates, and exact query ledgers. The matched target pairing removes between-target difficulty from the edge comparison.

56 Block 9 Executable Realization and Results

56.1 Development diagnosis

The development compiler evaluates 54 foreign target systems across six new tasks, three representation families, and three target seeds. Every policy receives three independent noisy paired-query replicates.

The inherited packet-QR child is full rank in every development replicate, but its weighted information floor remains weak.

Development diagnostic	Value
Rank-child full-rank rate	1.000000
Rank-child mean information eigenvalue	0.549830
Rank-child median condition number	21622.327
Recalibrated rank-child mean gain	0.028232
Best conditioning-policy mean gain	0.217374
Conditioning headroom gap	0.189142

Table 31: Frozen Block 9 bottleneck diagnosis. Rank is saturated, while conditioning remains a large unresolved source of metaproductivity headroom.

The compiler selects

$$\texttt{quality_d_opt} \tag{842}$$

as the grandchild query policy.

56.2 Primary foreign audit

The primary audit contains

$$8 \text{ tasks} \times 3 \text{ foreign representation families} \times 4 \text{ target seeds} = 96 \text{ paired targets}. \tag{843}$$

Arm	Selected gain	Commit	Full rank	$\lambda_{\min}$	Clean transport
Exact Block 7 child replay	0.000618	0.002604	1.000	0.518	0.304
Recalibrated rank child	0.037393	0.161458	1.000	0.518	0.304
Conditioning grandchild	**0.209580**	0.638021	1.000	**14.268**	**0.096**
Quality-permuted grandchild	0.000000	0.000000	1.000	0.643	0.293
Random 32-anchor control	0.111028	0.382813	0.844	3.278	0.180

Table 32: Primary foreign audit. Clean transport is normalized RMSE and is available only to the audit analysis. The grandchild and rank child have identical 24-query deployment budgets.

The paired grandchild-minus-recalibrated-child effect is

$$\widehat{\Delta}_{1\to 2} = 0.172188, \tag{844}$$

with frozen interval

$$[0.141481,\ 0.203110]. \tag{845}$$

The sign-flip estimate is

$$p = 9.9999 \times 10^{-6}. \tag{846}$$

The positive paired-effect rate is

$$0.677083, \tag{847}$$

and the negative paired-effect rate is zero. The remaining pairs are ties produced mainly by mutual abstention.

The exact Block 7 child replay is nearly inactive because its old alignment threshold was calibrated for effectively noiseless paired anchors. The stronger primary claim is therefore made against the recalibrated rank child, not only against the unchanged replay.

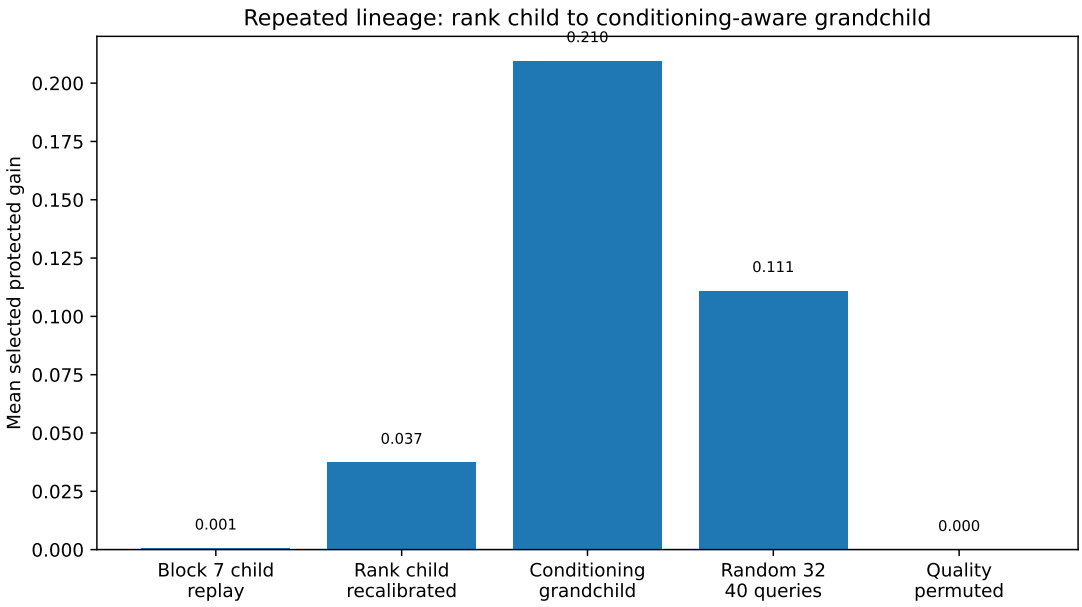


Figure 20: Primary foreign metaproductivity across the repeated lineage and controls. The conditioning grandchild uses the same 24 paired queries as the rank child.

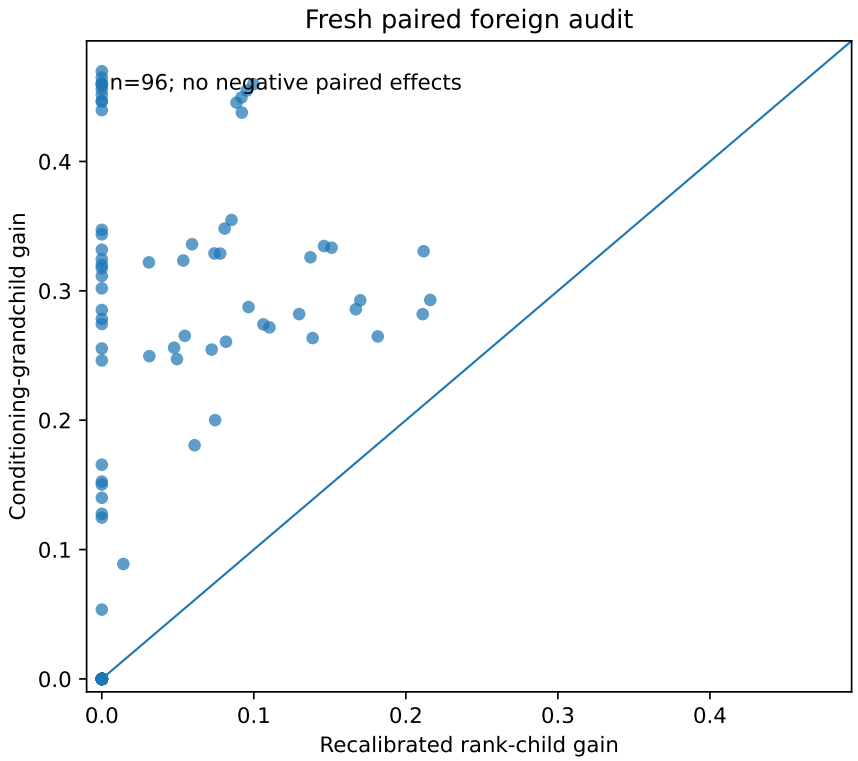


Figure 21: Target-wise conditioning-grandchild gain versus recalibrated rank-child gain on the fresh primary audit. No paired target favors the rank child.

56.3 Bottleneck migration is structural

Both the recalibrated rank child and the grandchild have full-rank rate one on the primary audit. The difference appears in the geometry beyond rank:

$$\lambda_{\min} : 0.518197 \longrightarrow 14.268195, \tag{848}$$

$$\text{median condition number} : 19393.296 \longrightarrow 544.180, \tag{849}$$

and

$$\text{clean transport NRMSE} : 0.303981 \longrightarrow 0.095803. \tag{850}$$

The debiased validation-alignment residual falls from

$$0.261487 \tag{851}$$

to

$$0.057707. \tag{852}$$

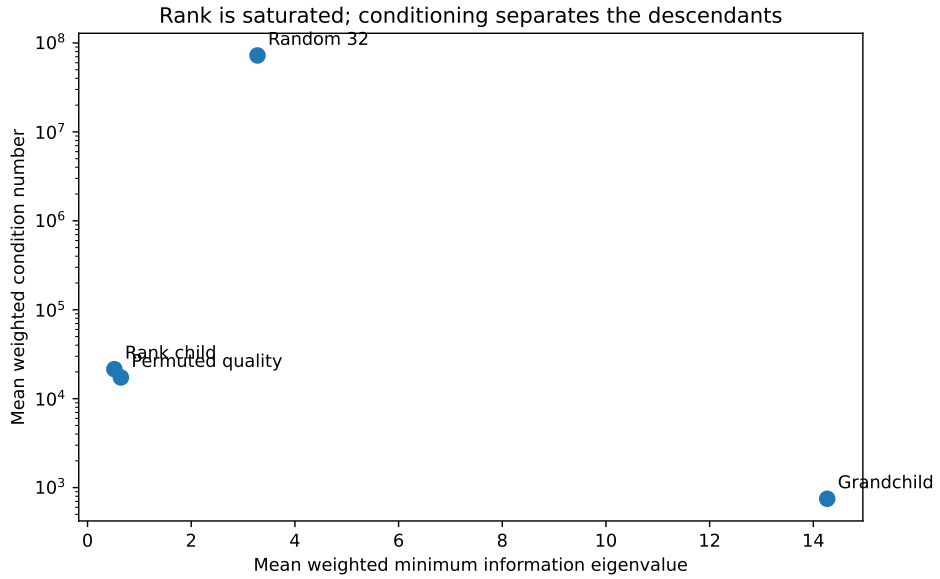


Figure 22: All principal designs can be full rank, yet their weighted information floors and condition numbers differ by orders of magnitude.

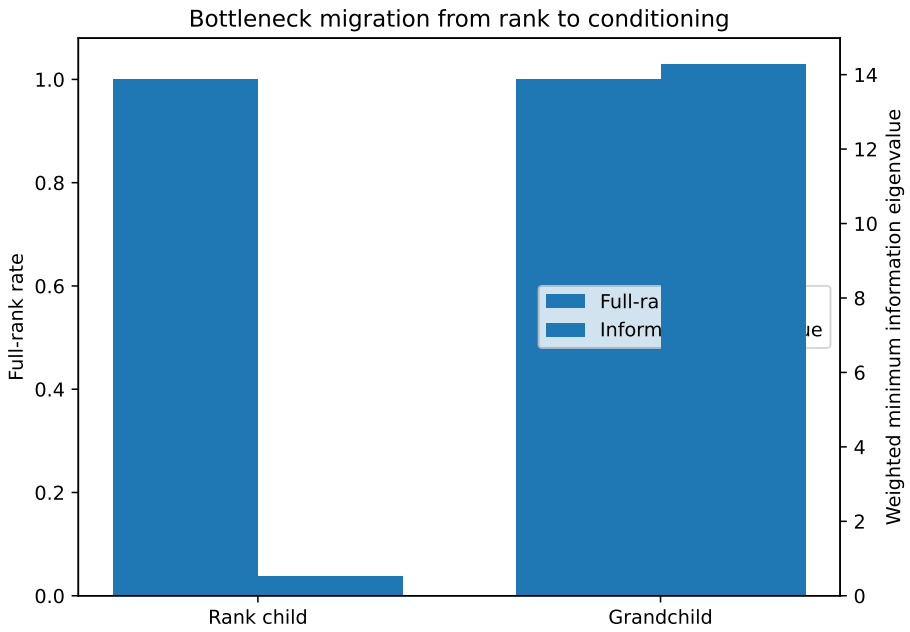


Figure 23: The rank endpoint is saturated for both generations. The grandchild improves the newly active conditioning coordinate.

These results instantiate the Block 8 meta-regret picture. The first bridge-policy edge reduced rank-identification regret. The second edge leaves rank unchanged and reduces conditioning-sensitive information regret.

56.4 Gate isolation and quality binding

Applying the grandchild gate to inherited rank anchors yields zero selected primary gain. This does not mean the inherited repair has zero blind value. Its mean blind primary gain is

$$0.157691. \tag{853}$$

It means the rank design does not satisfy the grandchild's declared information and alignment certificate.

The quality-permuted grandchild also has zero selected primary gain. Its blind primary gain is

$$0.154203, \tag{854}$$

but its mean information eigenvalue falls from

$$14.268195 \tag{855}$$

to

$$0.643375. \tag{856}$$

The paired quality-binding certificate has the same positive lower bound pattern as the gate-isolation comparison.

Unweighted D-optimal design is also rejected by the grandchild gate. This establishes that the observed benefit is not explained by determinant maximization alone. The anchor-quality relation is part of the realized mechanism.

56.5 Larger-budget control

The random 32-anchor control uses 40 paired queries and obtains mean primary selected gain

$$0.111028. \tag{857}$$

The conditioning grandchild uses 24 paired queries and obtains

$$0.209580. \tag{858}$$

The result is therefore not explained by a hidden increase in deployment query count.

The one-time development compiler uses

$$19440 \tag{859}$$

paired development queries. Relative to the random 32-anchor arm's additional 16 deployment queries, the declared query-only break-even point is

$$1215 \tag{860}$$

deployments. This accounting does not include engineering labor, model training, or the value of safer abstention.

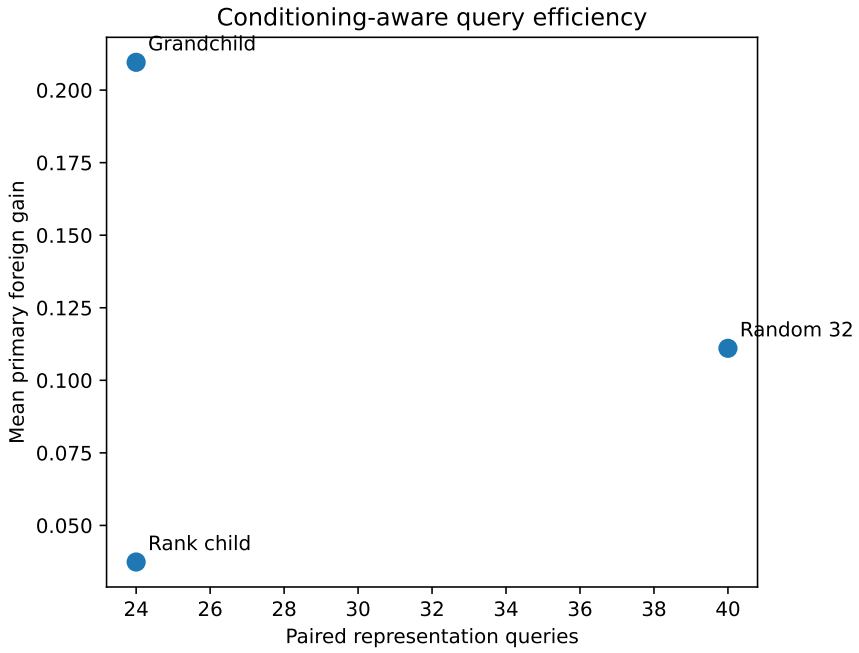


Figure 24: The grandchild outperforms the larger random-query control while using fewer paired target-representation queries per deployment.

56.6 Family-level results and negative boundaries

Family	Rank child	Grandchild	Grandchild commit
audit_rotated20	0.044815	0.315166	0.984375
audit_anisotropic24	0.000	0.001674	0.007813
audit_block28	0.067362	0.311900	0.921875
stress_severe_ill24	0.000	0.000000	0.000000
stress_compressed12	0.016366	0.055636	0.343750
stress_tanh24	0.000000	0.007651	0.039063

Table 33: Family-level selected gain. The severe target-conditioning family remains obstructed. Compression and nonlinear transport permit only partial, witness-gated transfer. Rounded rank-child family values are reported for orientation.

The audit-anisotropic primary family is an informative near-null result. Better source query conditioning does not remove target-side pseudoinverse amplification. The severe-ill stress family is rejected completely. The compressed and nonlinear stress families admit limited commits, but harmful selected-target rates remain zero in the frozen fixture.

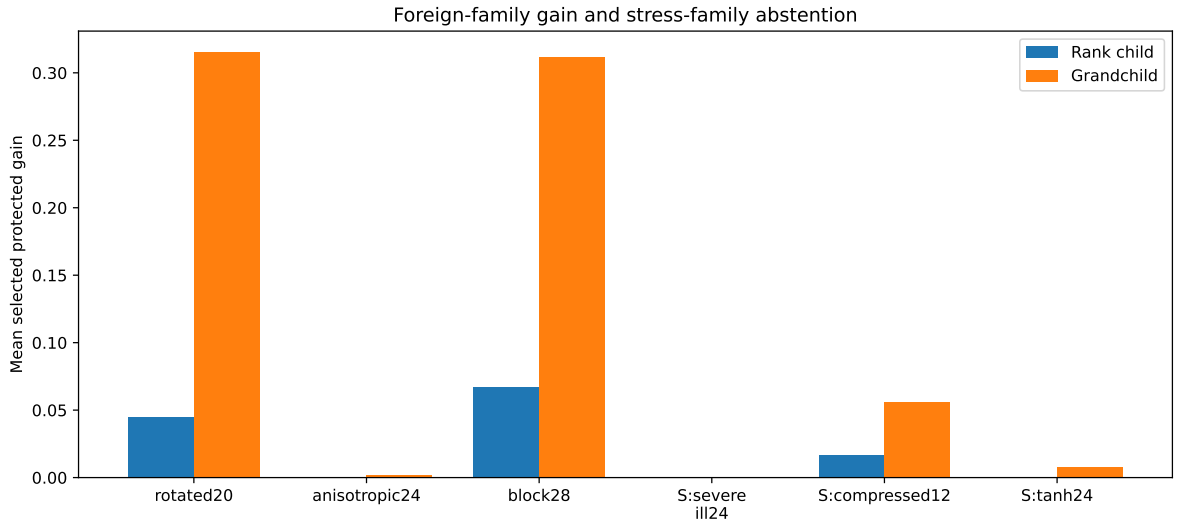


Figure 25: Conditioning-aware transfer is strong on two foreign primary families and remains limited or abstaining on the harder transport regimes.

56.7 Two-edge repeated-lineage certificate

The Block 7 child policy file has SHA-256 digest

$$\texttt{46e794fe5335c19a7dd56558128e0e15434447dd644bd9daf8f827c3e139f01c} \tag{861}$$

and semantic policy digest

$$\texttt{c241031f271f1f9a2e6ce7585319fc5eb64467e5fce31a4fb33373e7445bc1d2.} \tag{862}$$

The Block 9 grandchild receipt names that semantic digest as its lineage parent.

The Block 9 pre-audit seal has digest

`9b3014b1d3c4a8ef09aa34ae47baa134cdadd8cb9250e81f90a9e2dbac60fb07.` (863)

The canonical complete result has digest

`bdb8623fb1f2676f3828b38d51460f8a4d351ac8fd7618205c29908f812ebc56.` (864)

Theorem 56.1 (Finite two-edge repeated-lineage realization)**.** *Within the declared Block 7 and Block 9 contexts, the released receipts and audits instantiate two sequential productive edges:*

$$K_0 \xrightarrow{\Pi_1} K_1 \xrightarrow{\Pi_2} K_2. \tag{865}$$

The first edge improves rank-revealing bridge-query design relative to its Block 7 parent. The second edge replays K_1, detects rank saturation, and improves conditioning-aware query design relative to a recalibrated K_1 control under a fresh matched-resource audit.

Proof. The Block 7 receipt and audit provide the first sealed edge. The Block 9 package verifies the exact Block 7 child digest before compilation. Its development and audit task identifiers and primary representation families are disjoint, and no Block 7 final audit row is used in Block 9 development. The Block 9 pre-audit seal fixes the grandchild policy before final audit execution. The primary paired interval for $K_2 - K_1^{\mathrm{recal}}$ is strictly positive, query budgets match, both designs are full rank, protected harmful-target rates are zero, and the quality-permutation and gate-isolation controls fail to reproduce the grandchild result. These conditions instantiate the context-indexed repeated-lineage definition. □

Remark 56.2 (What repeated means here)**.** The theorem establishes two sequential successful lineage edges. It does not establish a winning strategy for infinitely many generations, compiler self-improvement, autonomous query-quality discovery, candidate-constructor growth, or open-set grammar expansion.

Block 9 result

Block 9 crosses the first repeated-lineage threshold. The exact Block 7 bridge child is replayed as the parent of a newly compiled conditioning-aware grandchild. On a fresh 96-target primary audit, the grandchild improves mean selected protected gain from 0.037393 for a recalibrated rank-child control to 0.209580 under the same 24-query deployment budget. The paired effect is 0.172188 with frozen interval $[0.141481, 0.203110]$. Both generations are full rank, so the improvement is attributed to a migration from rank recovery to heteroscedastic conditioning. The query-quality metadata, finite policy grammar, repair form, evaluator, witness, and target generators remain designer supplied.

57 Block 9 Reproducibility and Package

57.1 Package contents

The companion package contains:

1. the fresh five-dimensional task and defect generator;

2. source and foreign target representation families;
3. packet-QR, random, unweighted optimal-design, and quality-aware optimal-design policies;
4. heteroscedastic target-anchor observations and weighted bridge fitting;
5. ancestor receipt replay and lineage-parent hashing;
6. the finite grandchild compiler and bottleneck diagnosis ledger;
7. pre-audit sealing and matched-resource foreign audits;
8. six publication figures;
9. canonical JSON and CSV result objects;
10. nine automated tests;
11. a complete SHA-256 package manifest.

57.2 Canonical replay

A clean deterministic execution produces:

```
SRIC_BLOCK09_ALL_SUITES=PASS
result_sha256=bdb8623fb1f2676f3828b38d51460f8a4d351ac8fd7618205c29908f812ebc56
preaudit_seal_sha256=
  9b3014b1d3c4a8ef09aa34ae47baa134cdadd8cb9250e81f90a9e2dbac60fb07
grandchild_policy=quality_d_opt
rank_child_primary_gain=0.037393
conditioning_grandchild_primary_gain=0.209580
paired_grandchild_delta=0.172188
paired_ci_low=0.141481
rank_child_information=0.497677
grandchild_information=13.121386
......... [100%]
9 passed
SRIC_BLOCK09_PACKAGE_VALID
manifest_files=52
```

57.3 Claim boundary

The release establishes a controlled two-edge lineage inside two finite experimental grammars. It does not establish:

- autonomous discovery of query reliability;
- autonomous invention of D-optimal design;
- improvement of the outer compiler Γ_t;
- candidate-constructor or repair-image growth;
- spontaneous causal-atlas discovery in a pretrained model;

- evaluator self-improvement;
- open-ended recursive closure;
- indefinite acceleration;
- unrestricted safety;
- consciousness or sentience.

The next structural threshold is not a third refinement of the same query policy. The lineage must confront a defect for which every policy in the current global-linear bridge grammar remains obstructed, produce a grammar-inadequacy certificate, and introduce a conservative local-chart or nonlinear transport constructor whose new correction direction is necessary on foreign systems.

58 Structural Grammar Expansion by Local Transport Charts

Block 9 established a repeated lineage whose second productive edge migrated from rank recovery to heteroscedastic conditioning. That result remained inside one global-affine bridge grammar. Better query design reduced estimation error, but it could not represent a foreign relation whose transport law changes across observable contexts. Block 10 studies the next structural threshold: the inherited process must distinguish poor estimation inside the current grammar from persistent approximation error caused by the grammar itself, then add a conservative constructor that reaches a correction direction unavailable to its parent.

The executable realization keeps the Block 9 quality-weighted acquisition rule fixed. Every primary source design is full rank and strongly conditioned. The inherited parent receives more bridge-fit pairs than the child. A separate directly supervised global-affine oracle is fitted with far more target labels than either deployed process. The structural claim therefore does not rest on a weak parent fit. It rests on a persistent function-class obstruction and on a chart-resolved child that exceeds the best directly supervised global-affine comparison on fresh foreign targets.

58.1 Affine global and chart-resolved grammars

Let $H_s \cong \mathbb{R}^d$ be a source state space and $H_t \cong \mathbb{R}^q$ a target state space. Write

$$\bar{h} = \begin{bmatrix} h \\ 1 \end{bmatrix} \in \mathbb{R}^{d+1} \tag{866}$$

for homogeneous affine coordinates. Let $c : H_s \to \{-1, +1\}$ be an observable chart variable. A two-chart affine relation is specified by

$$\bar{T}_-, \bar{T}_+ \in \mathbb{R}^{q\times(d+1)}, \tag{867}$$

with

$$y = \bar{T}_{c(h)}\bar{h} + \varepsilon. \tag{868}$$

Equivalently, writing

$$\bar{A} = \frac{\bar{T}_+ + \bar{T}_-}{2}, \qquad \bar{B} = \frac{\bar{T}_+ - \bar{T}_-}{2}, \tag{869}$$

we have

$$y = (\bar{A} + c(h)\bar{B})\bar{h} + \varepsilon. \tag{870}$$

Definition 58.1 (Inherited global-affine grammar)**.** The inherited bridge grammar is

$$\mathcal{G}_{\mathrm{glob}} = \left\{ \bar{T} : \mathbb{R}^{d+1} \to \mathbb{R}^{q} \text{ linear in homogeneous coordinates} \right\}. \tag{871}$$

It fits one affine source-to-target map across every observable context.

Definition 58.2 (Invariant-context two-chart grammar)**.** Let $P_{\mathrm{ctx}} : H_s \to \mathbb{R}^r$ be a protected context projector whose value is unchanged by the source repair. Let $\mathcal{F}$ be a declared family of separators

$$g(h) = \mathbf{1}\{\langle u, P_{\mathrm{ctx}} h \rangle > \theta\}. \tag{872}$$

The chart extension is

$$\mathcal{G}_{\mathrm{chart}} = \left\{ (g, \bar{T}_0, \bar{T}_1) : g \in \mathcal{F},\ \bar{T}_j \in \mathbb{R}^{q \times (d+1)} \right\}. \tag{873}$$

Its prediction is

$$\widehat{y}(h) = \bar{T}_{g(h)} \bar{h}. \tag{874}$$

The protected-context restriction matters. A separator is not permitted to depend on coordinates changed by the repair being evaluated. Otherwise the chart assignment can change merely because the candidate repair was applied, creating an untyped control-flow effect.

Proposition 58.3 (Conservative embedding of the global grammar)**.** *For every separator $g \in \mathcal{F}$, the map*

$$\iota_g : \mathcal{G}_{\mathrm{glob}} \hookrightarrow \mathcal{G}_{\mathrm{chart}}, \qquad \bar{T} \longmapsto (g, \bar{T}, \bar{T}) \tag{875}$$

is injective and preserves every global-affine prediction exactly.

Proof. Both branches of $(g, \bar{T}, \bar{T})$ return $\bar{T}\bar{h}$, independently of $g(h)$. Equality of two embedded elements implies equality of their common affine map. □

Thus the child grammar does not delete the parent. It adds a contrast coordinate while retaining the parent as the diagonal subgrammar and as an executable fallback.

58.2 Global-affine obstruction

The next statement gives an exact function-class obstruction that does not depend on sample size.

Theorem 58.4 (Open-set incompatibility of distinct affine charts)**.** *Let $U_0, U_1 \subseteq \mathbb{R}^d$ each contain a nonempty open set. Let*

$$f_j(h) = A_j h + b_j, \qquad j \in \{0, 1\}. \tag{876}$$

If one affine map $f(h) = Ah + b$ satisfies

$$f(h) = f_j(h) \tag{877}$$

for every $h \in U_j$ and both j, then

$$A = A_0 = A_1, \qquad b = b_0 = b_1. \tag{878}$$

Consequently, if $(A_0, b_0) \neq (A_1, b_1)$, no element of $\mathcal{G}_{\mathrm{glob}}$ represents the two-chart relation exactly.

Proof. The affine difference $f - f_j$ vanishes on a nonempty open set. Every coordinate of this difference is an affine polynomial. An affine polynomial that vanishes on a nonempty open set has zero linear coefficient and zero constant coefficient. Hence $f = f_j$ as affine maps for both j. □

A population residual formula gives a quantitative special case. Suppose h is centered with covariance $\Sigma \succ 0$, $s \in \{-1, +1\}$ is independent of h with $\Pr(s = 1) = p$, and

$$y = (A + sB)h. \tag{879}$$

Then the best global linear map is

$$T^* = A + (2p - 1)B, \tag{880}$$

and

$$\inf_T \mathbb{E}\,\|y - Th\|^2 = 4p(1 - p)\,\mathrm{tr}(B\Sigma B^\top). \tag{881}$$

The proof is the normal-equation calculation from the centered two-component mixture. The executable fixture allows affine offsets, context-dependent source distributions, and heteroscedastic noise, so its operative certificate is a held-out residual margin plus a directly supervised global-affine oracle rather than literal substitution into Equation (881).

Remark 58.5 (Grammar-relative impossibility)**.** The obstruction is relative to one global affine map. The target relation is not declared intrinsically untransportable. It is affine on each chart and becomes expressible after a typed grammar extension.

58.3 Chartwise repair transport

Let the source repair be

$$R_s(h) = M_s h + c_s. \tag{882}$$

On chart j, let the target bridge be

$$T_j(h) = A_j h + b_j, \tag{883}$$

with A_j invertible on the repair-relevant subspace. Define

$$R_{t,j}(y) = T_j\Big(R_s(T_j^{-1}(y))\Big). \tag{884}$$

Expanding gives

$$R_{t,j}(y) = A_j M_s A_j^{-1}(y - b_j) + A_j c_s + b_j. \tag{885}$$

Theorem 58.6 (Exact chartwise affine repair transport)**.** *Under the preceding invertibility assumptions,*

$$R_{t,j}(T_j(h)) = T_j(R_s(h)) \tag{886}$$

for every repair-relevant h and each chart j.

Proof. Substitute $T_j(h) = A_j h + b_j$ into Equation (885):

$$R_{t,j}(T_j(h)) = A_j M_s A_j^{-1}(A_j h) + A_j c_s + b_j \tag{887}$$
$$= A_j(M_s h + c_s) + b_j \tag{888}$$
$$= T_j(R_s(h)). \tag{889}$$

□

The inherited global target-repair response lies in the diagonal class

$$\mathcal{D}_{\mathrm{diag}} = \{(R, R) : R \in \mathrm{Aff}(H_t)\}, \tag{890}$$

whereas the chart grammar reaches

$$\mathcal{D}_{\mathrm{chart}} = \mathrm{Aff}(H_t) \times \mathrm{Aff}(H_t). \tag{891}$$

Proposition 58.7 (New chart-contrast correction direction)**.** *Let the affine target repairs have homogeneous matrices $\bar{R}_0$ and $\bar{R}_1$. If*

$$\bar{R}_0 \neq \bar{R}_1, \tag{892}$$

then

$$\bar{R}_{\mathrm{ctr}} = \frac{\bar{R}_1 - \bar{R}_0}{2} \tag{893}$$

induces a nonzero anti-diagonal component

$$(-\bar{R}_{\mathrm{ctr}}, \bar{R}_{\mathrm{ctr}}) \tag{894}$$

orthogonal to every diagonal pair under the product Frobenius inner product. Therefore the chart grammar contains a correction direction unavailable to the parent grammar.

Proof. For every diagonal pair $(\bar{R}, \bar{R})$,

$$\left\langle (-\bar{R}_{\mathrm{ctr}}, \bar{R}_{\mathrm{ctr}}), (\bar{R}, \bar{R}) \right\rangle_F = 0. \tag{895}$$

The anti-diagonal component is nonzero exactly when the two chart repairs differ. □

The executable package records a normalized chart-specific bridge contrast

$$\kappa_T = \frac{\left\| \bar{T}_1 - \bar{T}_0 \right\|_F}{\frac{1}{2}(\left\| \bar{T}_1 \right\|_F + \left\| \bar{T}_0 \right\|_F) + 10^{-12}}. \tag{896}$$

A positive contrast alone is not a structural certificate. Block 10 combines it with global-oracle exhaustion, local residual reduction, conservative embedding, matched resources, foreign gain, and binding controls.

58.4 Finite separator discovery

In the executable fixture,

$$P_{\mathrm{ctx}}h = (h_1, h_2), \tag{897}$$

and the source repair leaves these coordinates invariant. The initial separator vocabulary is

$$\mathcal{F}_X = \{\mathbf{1}\{\langle u_k, P_{\mathrm{ctx}}h\rangle > \theta\} : 0 \le k < 128,\ \theta \in \Theta_k(X)\}, \tag{898}$$

where

$$u_k = (\cos(2\pi k/128), \sin(2\pi k/128)) \tag{899}$$

and $\Theta_k(X)$ contains zero and the empirical $0.30, 0.40, 0.50, 0.60, 0.70$ quantiles of the projected fit states. For every candidate separator, the compiler fits two weighted affine bridges and evaluates held-out chart-selection error. It then performs at most five deterministic residual-reassignment rounds and projects the resulting assignment back to one linear gate. The complete search rule, tie order, and iteration count are frozen before the audit.

Definition 58.8 (Finite grammar-inadequacy certificate)**.** A Block 10 certificate contains:

1. full-rank and adequate-information diagnostics for the inherited source design;
2. a directly supervised global-affine oracle residual that remains bounded away from the exact chart oracle;
3. a smaller held-out residual for the learned local-chart constructor;
4. a positive chart-repair or bridge contrast;
5. an explicit embedding of the global grammar into the chart grammar;
6. a finite repair-conflict set on which the global oracle is inadequate and the chart oracle is effective;
7. foreign-audit gain that disappears under gate-only, random-gate, or shuffled-binding controls.

Theorem 58.9 (Gate misclassification bound)**.** *Let g be the correct chart and $\widehat{g}$ an estimated chart. Suppose the two target repair outputs satisfy*

$$\|R_1(y) - R_0(y)\| \le L \tag{900}$$

throughout the evaluated region and

$$\Pr(\widehat{g} \ne g) \le \epsilon_g. \tag{901}$$

Then

$$\mathbb{E}\left\|R_{\widehat{g}}(y) - R_g(y)\right\|^2 \le \epsilon_g L^2. \tag{902}$$

If protected utility is L_J-Lipschitz in repaired state, then

$$\left|\mathbb{E}J(R_{\widehat{g}}(y)) - \mathbb{E}J(R_g(y))\right| \le L_J L\sqrt{\epsilon_g}. \tag{903}$$

Proof. The output difference is zero on correctly classified cases and has norm at most L on the error event. Conditioning gives the squared bound. Jensen's inequality and Lipschitz continuity give the utility bound. □

58.5 Conservative fallback

Let K_G be the global parent and K_C the chart process. Let $A_C(x)$ be the frozen chart-certificate decision for target episode x. The descendant is

$$K_3(x) = \begin{cases} K_C(x), & A_C(x) = 1, \\ K_G(x), & A_C(x) = 0 \text{ and the parent gate passes}, \\ \text{no-op}, & \text{otherwise.} \end{cases} \tag{904}$$

Proposition 58.10 (Pointwise fallback under a sound chart gate)**.** *If the chart gate accepts only when true protected utility under K_C is at least that under K_G, then*

$$J(K_3(x)) \geq J(K_G(x)) \tag{905}$$

for every episode.

Proof. On chart rejection the child executes the parent or the same no-op. On acceptance the result follows from gate soundness. □

The finite gate is not assumed perfectly sound. The empirical claim instead requires fresh paired audits, zero harmful selected targets in the primary suite, explicit abstention reporting, and a local-blind control showing that the witness removes some high-gain but harmful commits.

59 Block 10 Experimental Contract

59.1 Lineage edge and mutable grammar

Let K_2 denote the exact Block 9 conditioning-aware process. It supplies the quality-weighted D-optimal anchor-ordering rule and the published lineage receipt. Block 10 constructs two resource-matched processes:

$$K^{96}_{2,G}, \qquad K^{96}_{3,C}. \tag{906}$$

The global parent $K^{96}_{2,G}$ retains one affine bridge and uses all 80 non-witness pairs to estimate it. The chart child $K^{96}_{3,C}$ uses 48 pairs for local bridge fitting, 32 pairs for chart selection, and the same 16 witness pairs. Both consume exactly

$$96 \tag{907}$$

paired source-target queries. The edge is

$$K^{96}_{2,G} \xrightarrow{\Pi_3} K^{96}_{3,C}. \tag{908}$$

The parent is deliberately strengthened relative to the child on raw bridge-fitting sample count. In addition, an inadmissible global-affine oracle is fitted directly from 1600 labeled target-state and target-repaired-state pairs. This oracle does not use the source bridge at all. It estimates the best global affine repair map available with much more direct supervision and serves as a function-class ceiling comparison.

The main paired estimands are

$$\Delta_{G\to C} = \mathbb{E}\left[G(K^{96}_{3,C}, A) - G(K^{96}_{2,G}, A)\right] \tag{909}$$

and

$$\Delta_{\text{oracle}\to C} = \mathbb{E}\left[G(K^{96}_{3,C}, A) - G(K_{\text{affine oracle}}, A)\right]. \tag{910}$$

The second quantity is the sharper structural test.

59.2 Source state, protected packet, and repair

The source state is

$$h = (h_1, h_2, h_3, h_4) \in \mathbb{R}^4. \tag{911}$$

The first two coordinates form the protected context packet. The source repair is

$$R_s(h) = M_s h + c_s, \tag{912}$$

where

$$M_s = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.10 & 1.10 \\ 0 & 0 & 0 & 1.50 \end{bmatrix}, \qquad c_s = \begin{bmatrix} 0 \\ 0 \\ 0.30 \\ -0.25 \end{bmatrix}. \tag{913}$$

Hence

$$P_{\text{ctx}} R_s(h) = P_{\text{ctx}} h. \tag{914}$$

The observable chart is generated by a task-specific separator

$$c(h) = \mathbf{1}\{\langle u_*, P_{\text{ctx}} h\rangle > \theta_*\}. \tag{915}$$

The compiler does not receive u_* or θ_*, but it receives the declared protected packet in which a valid separator may be sought.

59.3 Piecewise foreign target atlas

Every primary target is four dimensional and has two invertible affine charts:

$$y = A_{c(h)} h + b + \varepsilon. \tag{916}$$

The two maps share the protected-context block but differ in the mutable response block. The source repair therefore preserves the chart while inducing two different target-space repair conjugates.

Development, meta-validation, and audit use disjoint task identifiers and disjoint target-family identifiers:

Split	Tasks	Families	Targets
Development	6	dev-chart-a/b/c	54
Meta-validation	4	meta-chart-h/i	24
Primary audit	8	audit-chart-d/e/f/g	96
Stress	4	three-chart, smooth, compressed, chart-crossing	32

Each primary task-family pair is instantiated with three unseen target seeds. Source distributions include heavy-tailed, curved-context, anisotropic, mixture, shifted, and correlated cases absent in the same combinations from compiler development.

59.4 Inherited quality-aware acquisition

A candidate pool contains source state h_i, target observation y_i, and known observation scale σ_i. The inherited acquisition policy selects rows sequentially to maximize the quality-weighted log-determinant increment

$$\log\det\left(\lambda I + \sum_{i\in S} \sigma_i^{-2} \bar{h}_i \bar{h}_i^\top\right). \tag{917}$$

Both parent and child use the same resulting order. The parent takes the first 80 rows for one global affine fit. The child takes the first 48 rows for local fits and the next 32 for separator selection. Both reserve the final 16 rows for the frozen witness.

The primary source design has homogeneous dimension five. Every development, meta-validation, audit, and stress design used for the headline comparison has full source-design rank. Thus the experiment does not recycle Block 7's rank deficiency or Block 9's poor-conditioning bottleneck.

59.5 Protected gain and witness quantities

For current target state y, candidate repaired state $\widetilde{y}$, and oracle repaired state $y^\star$, define

$$G = \mathrm{clip}_{[-1,1]}\left(1 - \frac{\mathbb{E}\,\|\widetilde{y} - y^\star\|^2}{\mathbb{E}\,\|y - y^\star\|^2}\right). \tag{918}$$

The witness records:

1. validation gain;
2. source-to-target alignment NRMSE;
3. decoded protected-context regression;
4. minimum chart support;
5. alignment reduction relative to the global bridge;
6. source-gate and target-gate agreement;
7. target-gate margin.

The selected global parent gate requires nonnegative validation gain, context regression at most 0.02, and global alignment NRMSE at most 0.80. The selected chart-child gate requires validation gain at least 0.20, context regression at most 0.05, local alignment NRMSE at most 0.35, at least eight fit observations in each chart, alignment reduction at least 0.10, gate agreement at least 0.90, and gate margin at least 0.35.

59.6 Finite compilation and pre-audit sealing

The parent is selected from

$$4 \times 4 \times 4 = 64 \tag{919}$$

gate configurations. The child is selected from

$$4 \times 4 \times 4 \times 4 \times 4 \times 3 \times 3 = 9216 \tag{920}$$

configurations. The development grammar certificate is computed first. Parent and child policies are selected only on the disjoint meta-validation split. Then the ancestor digest, grammar certificate, compiler ledger, policy objects, split commitments, and source hashes are sealed before primary audit generation.

The exact semantic digest of the inherited Block 9 policy is

`1a139f8e4818c8887f338128a29aaa93c5cfd23c0f75ab37fe941b6bc1243296`.

The Block 10 grammar certificate digest is

`25dbb14866b7d31baa74351f43f382d05f73c0a03c2bb7ac619e57b20479507b`.

The pre-audit seal digest is

`8f1936f93c8ab0d6d7da7b8a81c373603689b0d0087d352eefce0bea5b6c9162`.

59.7 Frozen controls

The package includes:

1. **Resource-matched global parent.** One global bridge, 80 fit pairs, 16 witness pairs.
2. **Direct global-affine oracle.** One target repair map fitted from 1600 directly supervised pairs. It is an inadmissible ceiling comparison.
3. **Local blind application.** The learned chart repair is always applied, bypassing the witness.
4. **Gate-only control.** The local gate is retained but both regions receive the same global repair.
5. **Random-gate control.** Two local bridges are bound through a random source partition.
6. **Shuffled-binding control.** The two learned local repairs are exchanged between predicted charts.
7. **True-chart oracle.** The correct chart labels are supplied. This estimates opportunity inside the declared two-chart family.
8. **Stress families.** Three-chart, smooth, compressed, and repair-dependent chart assignments test where the hard two-chart grammar fails.

60 Block 10 Results

60.1 Development grammar-inadequacy certificate

All 54 development targets have full source-design rank. The inherited global bridge has mean validation alignment NRMSE

$$\widehat{\varepsilon}_{\mathrm{glob}} = 0.632211, \tag{921}$$

whereas the learned two-chart constructor has

$$\widehat{\varepsilon}_{\text{chart}} = 0.068871. \tag{922}$$

The mean reduction is

$$0.563340. \tag{923}$$

The directly supervised global-affine repair oracle has mean gain

$$0.152169 \tag{924}$$

and maximum gain

$$0.364965 \tag{925}$$

across development. By contrast, the true-chart oracle has mean gain

$$0.998046, \tag{926}$$

and the learned local constructor has mean blind gain

$$0.613651. \tag{927}$$

Every one of the 54 development targets satisfies the frozen repair-conflict predicate

$$G_{\text{global oracle}} \le 0.40, \qquad G_{\text{true chart}} \ge 0.95. \tag{928}$$

Mean learned gate accuracy is 0.915000, and the mean chart-specific bridge contrast is 2.186784. These quantities, together with the conservative embedding, produce the frozen grammar-inadequacy certificate.

60.2 Primary foreign audit

The primary audit contains 96 foreign targets. Parent and child each consume 96 paired queries. Results are:

Process	Mean gain	Median	Commit	Positive	Harm
Global parent $K^{96}_{2,G}$	0.000000	0.000000	0.000000	0.000000	0
Chart child $K^{96}_{3,C}$	**0.334224**	0.284265	0.572917	0.572917	0
Direct global-affine oracle	0.176694	0.138838	–	1.000000	0
Local blind application	0.464387	0.502036	1.000000	0.958333	0.041667
True-chart oracle	0.998328	0.998514	–	1.000000	0

The paired child-minus-parent effect is

$$\widehat{\Delta}_{G\to C} = 0.334224, \tag{929}$$

with frozen bootstrap interval

$$[0.266459, 0.401903]. \tag{930}$$

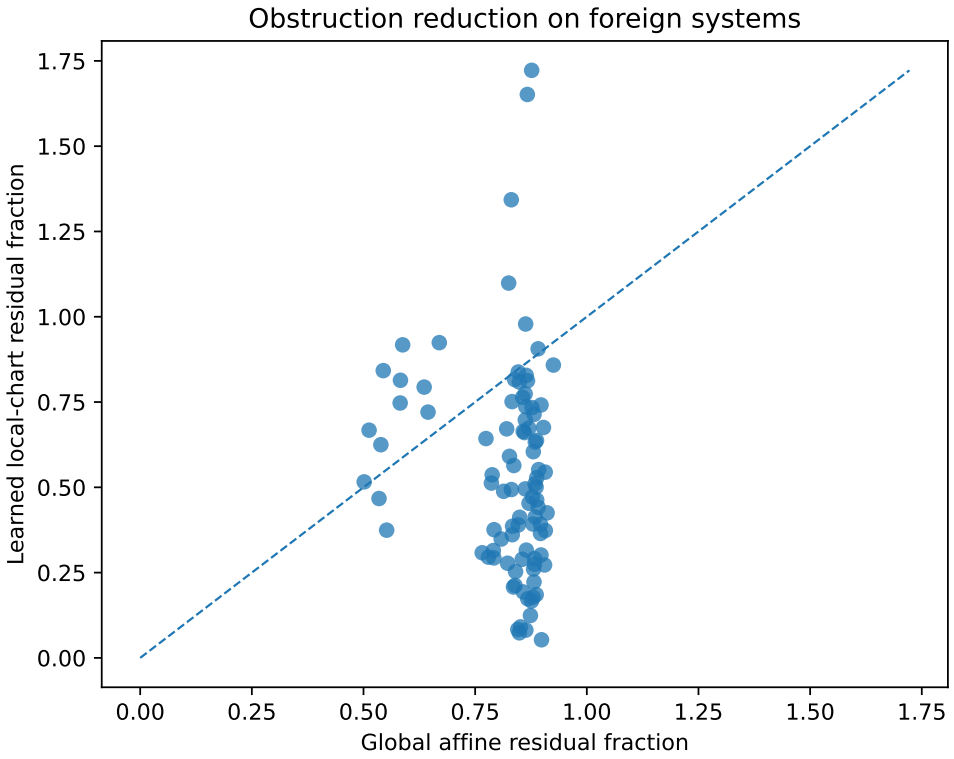


Figure 26: Global-affine oracle gain versus true-chart oracle gain. The primary and development regimes occupy a persistent grammar-conflict region: direct global supervision cannot reproduce the chart-resolved repair.

The sign-flip estimate is

$$p = 9.9999 \times 10^{-6}. \tag{931}$$

The positive-pair rate is 0.572917, the remaining 0.427083 are conservative abstention ties, and no paired target favors the global parent.

More importantly, the paired child-minus-direct-global-oracle effect is

$$\widehat{\Delta}_{\text{oracle}\to C} = 0.157530, \tag{932}$$

with frozen interval

$$[0.080863, 0.234135]. \tag{933}$$

Thus the child exceeds a directly supervised global-affine ceiling comparison on average, even though the oracle uses far more target labels and is not constrained to estimate a source-to-target bridge.

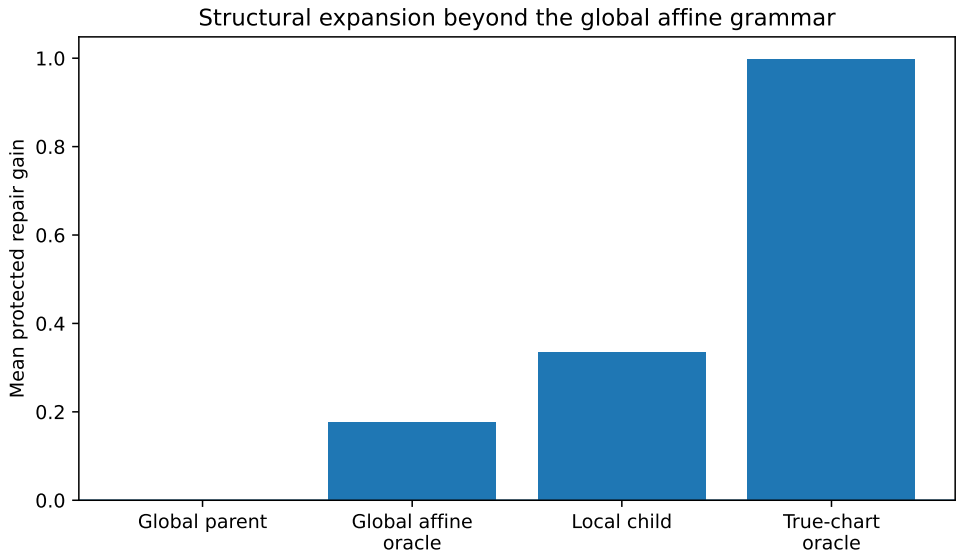


Figure 27: Mean primary protected gain. The chart child exceeds the resource-matched global parent and the directly supervised global-affine oracle.

60.3 Foreign-family results

Each audit family contains 24 targets. The chart child obtains:

Family	Child gain	Global oracle gain	Child commit
audit-chart-d	0.280114	0.168712	0.625000
audit-chart-e	0.336567	0.149444	0.625000
audit-chart-f	0.330004	0.189899	0.458333
audit-chart-g	0.390209	0.198721	0.583333

The child remains positive across all four unseen family identifiers and exceeds the mean global-affine oracle in every family.

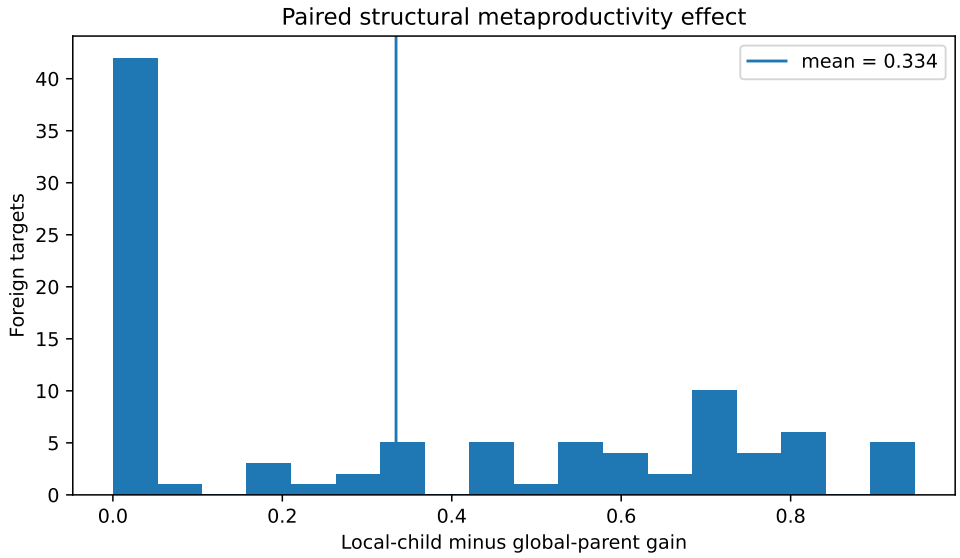


Figure 28: Target-wise chart-child minus global-parent protected gain. Zeros are witness abstention ties; no primary target has a negative paired effect.

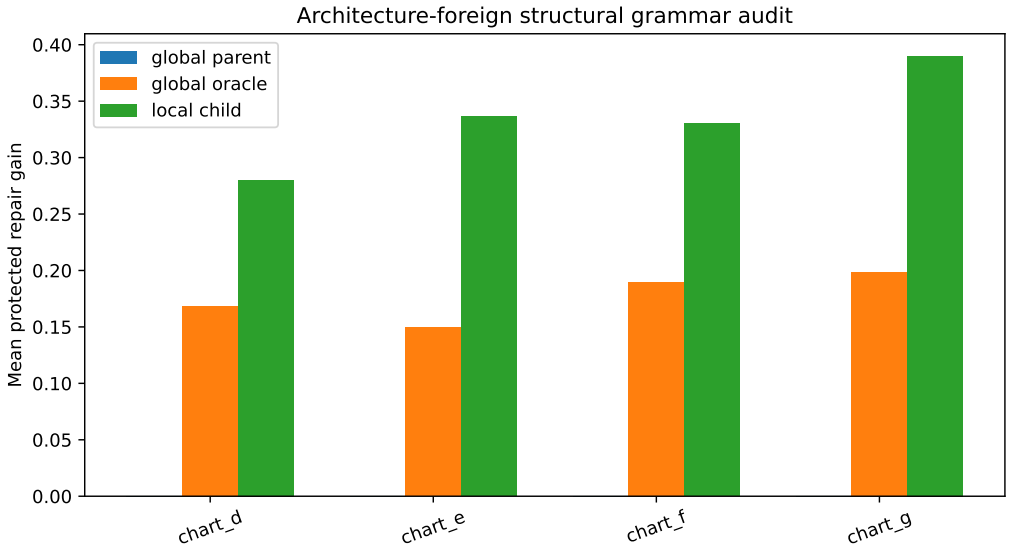


Figure 29: Mean selected gain by fresh foreign representation family. The parent abstains throughout; the chart child exceeds the directly supervised global-affine oracle in every family mean.

60.4 Structural diagnostics

On the primary audit, mean validation alignment NRMSE changes from

$$\widehat{\varepsilon}_{\mathrm{glob}} = 0.598513 \tag{934}$$

to

$$\widehat{\varepsilon}_{\mathrm{chart}} = 0.064975. \tag{935}$$

Mean learned gate accuracy is

$$0.874854. \tag{936}$$

The directly supervised global-affine oracle leaves mean residual fraction

$$0.823306 \tag{937}$$

of the unrepaired defect energy. The learned local constructor, before witness selection, leaves mean fraction

$$0.535613. \tag{938}$$

The true-chart oracle leaves only approximately 0.001672 of the original error fraction on average.

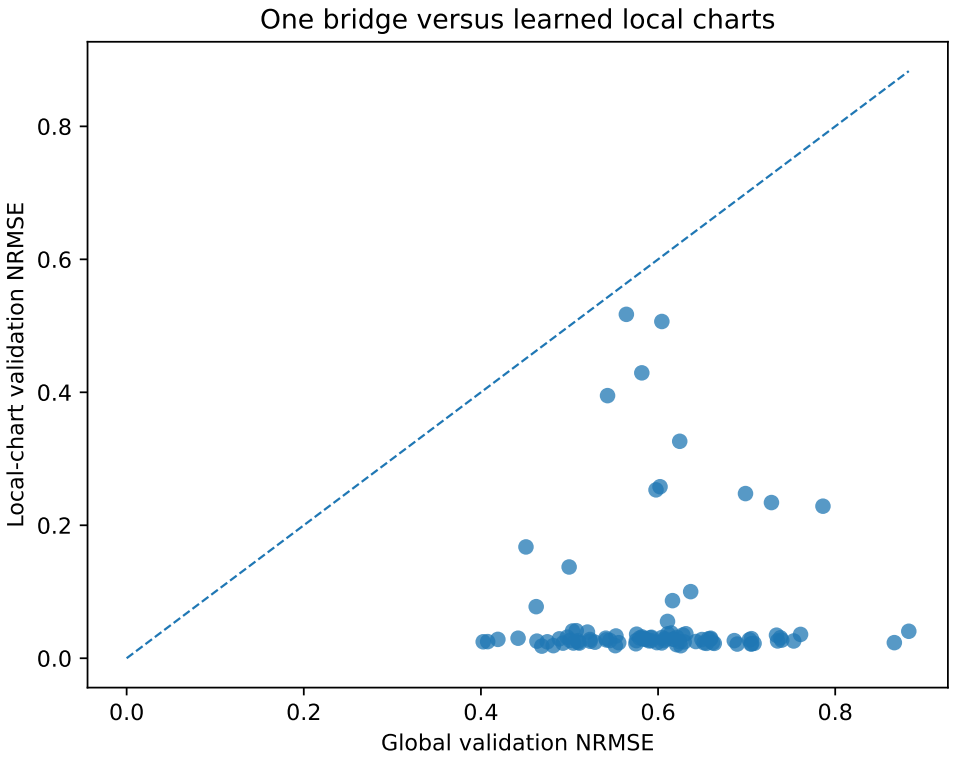


Figure 30: Global versus chart-resolved validation NRMSE. The chart grammar sharply lowers alignment error despite using fewer bridge-fit anchors than the global parent.

60.5 Mechanism controls

The frozen controls give:

Arm	Mean gain	Positive rate	Harm rate
Global blind	−0.833287	0.000000	1.000000
Gate only	−0.833287	0.000000	1.000000
Random gate	−0.689622	0.072917	0.927083
Shuffled binding	−1.000000	0.000000	1.000000
Local blind	0.464387	0.958333	0.041667
Witness-selected child	0.334224	0.572917	0.000000

The gate-only result shows that an acceptance rule cannot manufacture missing chart-specific transport information. The random-gate result shows that merely increasing the number of affine maps is insufficient. Shuffled binding shows that the maps must remain attached to the correct regions. Local blind application obtains higher mean gain than the selected child but commits four harmful targets. The witness trades some utility for zero harmful selected targets in the frozen primary suite.

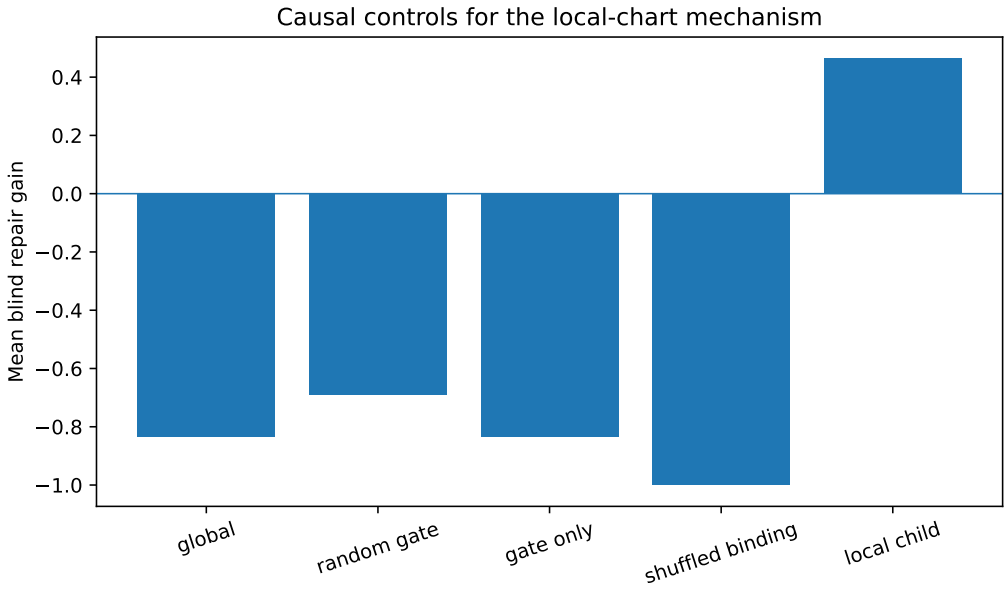


Figure 31: Causal and mechanism controls. Only the correctly gated and correctly bound local constructor produces positive primary gain without harmful commits.

60.6 Stress boundaries

The child abstains on every target in the three-chart, smooth-transition, compressed, and repair-dependent chart-crossing stress families. This is intentional. Blind local application is harmful in the smooth, three-chart, and chart-crossing regimes. The hard two-chart constructor is therefore not described as a universal nonlinear transport method.

The stress results identify distinct next obstructions:

1. three-chart targets require a larger discrete atlas;
2. smooth targets require a soft gate, local manifold atlas, or nonlinear lift;
3. compressed targets require obstruction-faithful information-loss accounting rather than another local affine map;
4. repair-dependent chart assignments require a transition model whose control flow is itself transported and witnessed.

60.7 Finite structural grammar-expansion theorem

Theorem 60.1 (Finite obstruction-certified structural grammar expansion)**.** *Within the declared Block 10 benchmark and frozen implementation, the edge in Equation (908) is a bounded structural expansion of the repair-transport grammar. Specifically:*

1. *the exact Block 9 ancestor digest is verified before compilation;*
2. *development, meta-validation, and primary audit supports are disjoint by task and family identifier;*
3. *all primary source designs are full rank and the global parent receives at least as many bridge-fit pairs as the child;*
4. *the directly supervised global-affine oracle has a persistent foreign residual and is exceeded by the child with a paired interval excluding zero;*
5. *the global grammar embeds exactly as the diagonal subgrammar and runtime fallback;*
6. *the child introduces a nonzero chart-contrast response direction unavailable to the global grammar;*
7. *parent and child have the same total paired-query budget;*
8. *child-minus-parent protected gain has a positive frozen interval, no negative primary paired effects, and zero harmful selected targets;*
9. *gate-only, random-gate, shuffled-binding, and blind-application controls rule out the principal finite shortcuts;*
10. *the child abstains rather than overclaiming adequacy on the declared stress families;*
11. *compiler, policies, grammar certificate, source hashes, and split commitments are sealed before primary audit generation.*

Proof. Items 1, 2, and 11 follow from the ancestor receipt, split ledger, and pre-audit seal. Item 3 follows from the source-rank and query ledgers. Item 4 is the paired direct-global-oracle comparison. Item 5 is Theorem 58.3 and the executable fallback. Item 6 follows from Theorem 58.7 and the recorded nonzero chart-specific response. Item 7 is the declared 96-query envelope. Item 8 is the frozen paired audit. Item 9 is the control table. Item 10 is the stress ledger. Together these instantiate the manuscript's task-relevant structural-metaproductivity criterion: a new authorized response direction absent from the parent grammar reduces protected foreign error under matched deployment resources. □

Block 10 result

Block 10 crosses the first bounded structural grammar-expansion threshold. A strengthened global-affine parent uses 80 fit and 16 witness pairs. A conservative two-chart child uses 48 fit, 32 chart-selection, and 16 witness pairs. Across 96 fresh foreign targets, child mean selected gain is 0.334224 with zero harmful commits. The child-parent effect is 0.334224 with interval $[0.266459, 0.401903]$. More decisively, the child exceeds a directly supervised global-affine oracle by 0.157530 on average, with interval $[0.080863, 0.234135]$. Mean validation NRMSE falls from 0.598513 to 0.064975. The protected context packet, finite separator search, piecewise target family, source repair, evaluator, witness, and outer compiler remain designer supplied.

61 Block 10 Reproducibility and Package

61.1 Package contents

The reference implementation contains:

1. exact Block 9 ancestor receipts;
2. quality-weighted D-optimal anchor acquisition;
3. global and local affine bridge fitting;
4. finite protected-context separator search and deterministic residual reassignment;
5. chart-specific repair transport;
6. a directly supervised global-affine ceiling control;
7. global-blind, local-blind, gate-only, random-gate, shuffled-binding, true-chart, and stress controls;
8. development grammar certificates, compiler ledgers, selected policy objects, and a pre-audit seal;
9. canonical JSON and CSV results;
10. seven reproducible figures;
11. eleven automated tests;
12. an exact 51-file package manifest;
13. PowerShell, Python, and Makefile runners.

61.2 Canonical replay

A clean execution prints:

```
SRIC_BLOCK10_ALL_SUITES=PASS
result_sha256=
2425ff043d378dae18f3a07d9122257c0385d21f8c172419a5fc14e712cc8bc7
preaudit_seal_sha256=
8f1936f93c8ab0d6d7da7b8a81c373603689b0d0087d352eefce0bea5b6c9162
```

```
grammar_certificate_sha256=
25dbb14866b7d31baa74351f43f382d05f73c0a03c2bb7ac619e57b20479507b
global_parent_gain=0.000000
local_chart_child_gain=0.334224
paired_structural_delta=0.334224
paired_ci_low=0.266459
global_affine_oracle_gain=0.176694
child_harm_rate=0.000000
```

A second execution regenerates the same canonical result digest. The clean package verification prints

```
SRIC_BLOCK10_PACKAGE_VALID
manifest_files=51
result_sha256=2425ff043d378dae18f3a07d9122257c0385d21f8c172419a5fc14e712cc8bc7
```

61.3 Claim boundary

The release establishes a finite lineage-generated structural expansion from one global affine bridge to a learned invariant-context two-chart affine transport grammar under matched paired-query resources.

It does not establish:

- autonomous discovery of the protected context packet;
- unrestricted invention of separator features or chart number;
- general smooth, three-chart, noninjective, or repair-dependent transport;
- improvement of the outer lineage compiler;
- autonomous evaluator or witness evolution;
- open-ended recursive improvement or indefinite acceleration;
- unrestricted safety;
- consciousness, sentience, or moral status.

The next major threshold is endogenous atlas growth. A descendant should receive no supplied protected separator packet, infer an intervention-stable local cover from response operators, distinguish hard charts from smooth or noninjective structure, conservatively replay the global and two-chart grammars, and expose a further foreign correction direction under a new sealed audit.

62 Endogenous Atlas Growth from Intervention Responses

Block 10 established a finite structural extension from one global affine transport map to a two-chart affine grammar. Its separator, however, was supplied through a protected context packet. Block 11 removes that packet. The descendant receives hidden states, a finite family of controlled interventions, source-repair traces, paired source–target examples, and witness evidence. It must

recover a functionally meaningful local cover from intervention responses, determine which recovered block supports repair-consistent charting, and retain the inherited process as a conservative fallback.

The new edge is therefore

$$K_0 \xrightarrow{\Pi_1} K_1 \xrightarrow{\Pi_2} K_2 \xrightarrow{\Pi_3} K_3 \xrightarrow{\Pi_4} K_4, \tag{939}$$

where K_3 is the Block 10 chart process with a supplied protected packet and K_4 replaces that packet by an intervention-response atlas compiler. This is not unrestricted atlas discovery. The intervention family, finite state dimension, target task, repair family, chart count, evaluator, witness, and outer compiler remain declared. The result is a controlled first realization of *endogenous functional packet recovery* inside the lineage.

62.1 Intervention-response algebras

Let H be a finite-dimensional real or complex Hilbert space. Suppose the unknown functional decomposition is

$$H = \bigoplus_{a=1}^{m} H_a, \qquad P_a : H \to H_a, \tag{940}$$

with pairwise orthogonal projectors satisfying

$$P_a P_b = \delta_{ab} P_a, \qquad \sum_{a=1}^{m} P_a = I. \tag{941}$$

For controlled interventions $u_1, \dots, u_k$, let

$$R_j \in \mathrm{End}(H), \qquad j = 1, \dots, k, \tag{942}$$

be their finite-difference or linearized response operators.

Definition 62.1 (Block-scalar response model)**.** The response family is block scalar on Equation (940) when

$$R_j = \sum_{a=1}^{m} \lambda_{aj} P_a, \qquad \lambda_{aj} \in \mathbb{R}. \tag{943}$$

The response signature of block a is

$$\lambda_a = (\lambda_{a1}, \dots, \lambda_{ak}) \in \mathbb{R}^k. \tag{944}$$

The signatures are separating when $\lambda_a \neq \lambda_b$ for every $a \neq b$.

The algebra generated by the response family is

$$\mathfrak{A}_R = \mathrm{alg}^*(I, R_1, \dots, R_k). \tag{945}$$

Because the R_j are simultaneously block scalar in the exact model, $\mathfrak{A}_R$ is commutative and every polynomial in the responses is again block scalar.

Theorem 62.2 (Exact response-signature atlas recovery)**.** *Assume the block-scalar response model and separating signatures. Then every hidden projector P_a belongs to $\mathfrak{A}_R$. Consequently, the nonzero*

minimal central projections of $\mathfrak{A}_R$ are exactly $P_1, \dots, P_m$, up to permutation.

Proof. Choose $c \in \mathbb{R}^k$ outside the finite union of hyperplanes

$$\left\{ c : c^{\mathsf{T}}(\lambda_a - \lambda_b) = 0 \right\}, \qquad a \neq b. \tag{946}$$

Then the scalars

$$\mu_a = c^{\mathsf{T}} \lambda_a \tag{947}$$

are pairwise distinct. The combined response

$$B = \sum_{j=1}^{k} c_j R_j = \sum_{a=1}^{m} \mu_a P_a \tag{948}$$

belongs to $\mathfrak{A}_R$. For each a, define the Lagrange polynomial

$$p_a(z) = \prod_{b \neq a} \frac{z - \mu_b}{\mu_a - \mu_b}. \tag{949}$$

Functional calculus gives

$$p_a(B) = \sum_{b=1}^{m} p_a(\mu_b) P_b = P_a. \tag{950}$$

Thus every P_a lies in the generated algebra. A nonzero central projection that is strictly below one P_a would have to be scalar on H_a and therefore is either 0 or P_a. Hence the P_a are the minimal central projections. □

Corollary 62.3 (Gauge boundary)**.** *The exact response family identifies each subspace H_a but does not select a preferred basis inside H_a. Atlas recovery is unique up to block permutation and an independent invertible, or unitary in the orthogonal setting, change of basis inside every recovered block.*

This is the appropriate identifiability target. A functional atlas is a family of invariant packets, not a coordinate labeling of individual neurons or latent dimensions.

62.2 Commutants and unresolved multiplicity

The response commutant is

$$\mathfrak{A}'_R = \{ X \in \mathrm{End}(H) : XR_j = R_j X \text{ for every } j \}. \tag{951}$$

Theorem 62.4 (Commutant characterization)**.** *Under the hypotheses of Theorem 62.2,*

$$\mathfrak{A}'_R = \bigoplus_{a=1}^{m} \mathrm{End}(H_a). \tag{952}$$

If $d_a = \dim H_a$, then

$$\dim \mathfrak{A}'_R = \sum_{a=1}^{m} d_a^2. \tag{953}$$

Proof. Write $X = \sum_{a,b} P_a X P_b$. The relation $XR_j = R_j X$ implies

$$(\lambda_{bj} - \lambda_{aj}) P_a X P_b = 0 \tag{954}$$

for every j. Separating signatures imply that for $a \neq b$ some coefficient is nonzero, so $P_a X P_b = 0$. Hence X is block diagonal. Every block-diagonal operator commutes with each block-scalar R_j. The dimension formula follows from $\dim \mathrm{End}(H_a) = d_a^2$. □

For three rank-two blocks, the exact commutant dimension is

$$3 \cdot 2^2 = 12, \tag{955}$$

which is the canonical finite signature used by the Block 11 reference package.

Proposition 62.5 (Degenerate-signature obstruction)**.** *If $\lambda_a = \lambda_b$ for distinct blocks a and b, then every element of $\mathfrak{A}_R$ acts with the same scalar on H_a and H_b. The response algebra can identify $P_a + P_b$ but cannot identify P_a and P_b separately.*

Proof. Every polynomial in the R_j evaluates to the same scalar on blocks with the same joint response signature. Therefore no element of the generated algebra separates the two projectors. □

A failure to split a degenerate pair is not a failed optimizer. It is an observational-identifiability obstruction relative to the current intervention family. The correct response is abstention, acquisition of a new separating intervention, or retention of the merged block.

62.3 Approximate response atlases

In an executable system, the responses are estimated from finite intervention data. Let

$$\widehat{R}_j = R_j + E_j, \tag{956}$$

and fix coefficients c_j. Define

$$B = \sum_j c_j R_j, \qquad \widehat{B} = \sum_j c_j \widehat{R}_j, \qquad E = \widehat{B} - B. \tag{957}$$

Let the distinct eigenvalue associated with block a be μ_a and define its separation

$$\gamma_a = \min_{b \neq a} |\mu_a - \mu_b|. \tag{958}$$

Conditional Theorem 62.6 (Spectral stability of a recovered atlas block)**.** *Suppose B is self-adjoint, $\|E\|_2 < \gamma_a/2$, and $\widehat{P}_a$ is the spectral projector of $\widehat{B}$ onto the eigenvalue cluster corresponding to μ_a. Then*

$$\left\| \widehat{P}_a - P_a \right\|_2 \leq \frac{2\,\|E\|_2}{\gamma_a}. \tag{959}$$

Proof. This is the finite-dimensional spectral-subspace perturbation bound obtained by applying the resolvent formula on a contour at distance $\gamma_a/2$ from the selected eigenvalue cluster. The resolvent identity contributes one factor of $\|E\|_2$, while both resolvents are bounded by $2/\gamma_a$. Integrating over the contour gives the stated standard Davis–Kahan form. □

The empirical certificate therefore records both estimation error and spectral separation. A small residual without a gap does not certify stable blocks.

For recovered projectors $\widehat{P}_1, \dots, \widehat{P}_m$, define the normalized response-fit residual

$$\varepsilon_{\mathrm{resp}} = \frac{\left(\sum_j \left\| \widehat{R}_j - \sum_a \widehat{\lambda}_{aj} \widehat{P}_a \right\|_F^2 \right)^{1/2}}{\left(\sum_j \left\| \widehat{R}_j \right\|_F^2 \right)^{1/2} + 10^{-12}}, \tag{960}$$

and the maximum normalized commutator residual

$$\varepsilon_{\mathrm{comm}} = \max_{a,j} \frac{\left\| \widehat{P}_a \widehat{R}_j - \widehat{R}_j \widehat{P}_a \right\|_F}{\left\| \widehat{R}_j \right\|_F + 10^{-12}}. \tag{961}$$

The smallest selected spectral gap is denoted $\widehat{\gamma}_{\min}$. These quantities serve different roles:

- $\varepsilon_{\mathrm{resp}}$ measures whether the finite block-scalar model fits the response family;
- $\varepsilon_{\mathrm{comm}}$ measures whether the recovered packets behave as approximately invariant response modules;
- $\widehat{\gamma}_{\min}$ measures stability of the recovered decomposition.

62.4 From an unlabeled atlas to a repair-relevant chart packet

The projectors recovered by Theorem 62.2 are unlabeled. The lineage must still determine which block supports a chart separator that is stable under the source repair and useful for target transport.

For each recovered block $\widehat{P}_a$, let $\mathcal{F}_a$ be a finite family of candidate separators

$$g_{a,\theta,u}(h) = \mathbf{1}\{\left\langle u, \widehat{P}_a h \right\rangle > \theta\}. \tag{962}$$

The compiler fits a two-chart bridge for every candidate and evaluates a score containing:

1. held-out local-bridge validation gain;
2. source-repair invariance of the chart assignment;
3. agreement between source and target chart assignments;
4. chart-balance constraints;
5. response-fit, commutator, and spectral-gap penalties;
6. protected-context regression.

Let s_a denote the best population score attainable from block a, and let $\widehat{s}_a$ be its empirical score.

Theorem 62.7 (Finite repair-relevant block selection)**.** *Suppose there is a unique repair-relevant block a^* and a margin $m > 0$ such that*

$$s_{a^*} \geq s_a + m \tag{963}$$

for every $a \neq a^*$. If

$$\max_a |\widehat{s}_a - s_a| < \frac{m}{2}, \tag{964}$$

then

$$\arg\max_a \widehat{s}_a = a^*. \tag{965}$$

Proof. For every $a \neq a^*$,

$$\widehat{s}_{a^*} - \widehat{s}_a \geq s_{a^*} - s_a - |\widehat{s}_{a^*} - s_{a^*}| - |\widehat{s}_a - s_a| \tag{966}$$

$$> m - \frac{m}{2} - \frac{m}{2} = 0. \tag{967}$$

□

The theorem clarifies the division of labor. Intervention responses recover a functional cover up to gauge. Repair-conflict and held-out transport evidence label the recovered packets relative to the current defect.

62.5 Conservative relation to the supplied-packet grammar

Let $\mathcal{G}_{10}$ denote the Block 10 two-chart grammar with a supplied context projector P_{ctx}. Let $\mathcal{G}_{11}$ contain:

1. the intervention-response atlas compiler;
2. a finite family of projectors and separators returned by the compiler;
3. every constructor and witness rule of $\mathcal{G}_{10}$;
4. a declared fallback that may instantiate the old projector directly.

Proposition 62.8 (Conservative embedding of the Block 10 grammar)**.** *There is an injective embedding*

$$\iota_{10\to 11} : \mathcal{G}_{10} \hookrightarrow \mathcal{G}_{11} \tag{968}$$

obtained by fixing the atlas output to P_{ctx} and retaining the inherited separator, bridge, repair, evaluator, and witness objects. Every Block 10 execution has an observationally identical image in $\mathcal{G}_{11}$.

Proof. The endogenous grammar strictly generalizes the source of the projector but does not remove the old constructor. Choosing the declared old projector and inherited separator reproduces every old branch, prediction, repair, witness decision, and receipt field. Injectivity follows because distinct Block 10 objects remain distinct after the additional fixed atlas fields are appended. □

Thus Block 11 does not obtain improvement by deleting the old process. The old process remains available as a valid but potentially poorly aligned atlas choice.

Proposition 62.9 (Fixed-packet insufficiency under hidden rotation)**.** *Let P_0 be a fixed rank-r packet and P_* a target-specific rank-r context packet. Suppose the chart label $c(h)$ is measurable from $P_* h$ but is not almost surely measurable from $P_0 h$. Then no separator depending only on $P_0 h$ can reproduce $c(h)$ almost surely. In particular, a target-specific rotation that moves* $\operatorname{im} P_*$ *away from* $\operatorname{im} P_0$ *creates an irreducible fixed-packet chart error.*

Proof. If a separator $g(P_0 h)$ reproduced $c(h)$ almost surely, then c would be measurable with respect to the sigma-algebra generated by $P_0 h$, contradicting the assumption. □

This is the finite reason the inherited parent can safely abstain while the endogenous child identifies useful target-specific context packets.

62.6 Cyclic defect modules and atlas localization

Let $r \in H$ be a current hidden-state defect residual. Define

$$H_r = \operatorname{span}\{Ar : A \in \mathfrak{A}_R\}. \tag{969}$$

Proposition 62.10 (Minimal response-invariant defect module)**.** *The subspace H_r is the smallest $\mathfrak{A}_R$-invariant subspace containing r. If $\mathfrak{A}_R$ is adjoint closed, then H_r is reducing and has an orthogonal projector.*

Proof. The span contains r because $I \in \mathfrak{A}_R$. It is invariant because $B(Ar) = (BA)r$ for $A, B \in \mathfrak{A}_R$. Every invariant subspace containing r must contain Ar for every A, hence contains H_r. Adjoint closure makes the orthogonal complement invariant as well. □

This yields a two-scale endogenous decomposition:

$$\text{central response blocks} \quad \longrightarrow \quad \text{defect-specific cyclic module.} \tag{970}$$

The first identifies coarse functional strata. The second localizes the current defect inside their response closure.

63 Block 11 Experimental Contract

63.1 Lineage and withheld structure

The package replays the exact published Block 10 ancestor policy and receipt digests. The supplied context packet is then removed from both the inherited parent and the endogenous child. The parent is permitted to use the legacy fixed first-two-coordinate packet. The child receives only intervention-response observations and the same ordinary bridge, validation, and witness interfaces.

The source hidden space has dimension six and is generated from three target-specific unknown rank-two blocks:

$$H = H_{\mathrm{ctx}} \oplus H_{\mathrm{rep}} \oplus H_{\mathrm{nui}}. \tag{971}$$

A target-specific orthogonal or injective embedding hides these blocks from the runtime coordinate system. The source repair changes only H_{rep} and leaves the context block invariant. Two controlled interventions act with different block signatures. The context label controls which target bridge and target repair are valid.

No packet coordinate, central projector, atlas label, or target chart label is supplied to the child. Ground-truth projectors are retained only for audit metrics such as subspace overlap and the inadmissible true-chart oracle.

63.2 Query accounting

Every primary target receives exactly 120 effect-query units. The final frozen accounting is:

Process	Bridge fit	Response	Chart validation	Witness	Total
Legacy fixed-packet parent	80	0	16	24	**120**
Endogenous-atlas child	56	24	16	24	**120**

The response budget consists of two interventions evaluated on twelve source states. The child therefore spends fewer ordinary bridge-fit pairs than the parent. A gain cannot be attributed to a larger total deployment budget.

63.3 Data chronology and sealing

The release uses:

Split	Targets	Purpose	Audit access
Development	36	response atlas and candidate-score construction	unavailable
Meta-validation	24	witness thresholds and finite compiler selection	unavailable
Primary foreign audit	80	final paired claim	opened after seal
Stress audit	25	declared boundary families	opened after seal

The chronology is:

$$\begin{aligned} &\text{replay Block 10 ancestor digests} \\ &\longrightarrow \text{generate development and meta-validation evidence} \\ &\longrightarrow \text{compile atlas certificate, parent, child, and witness} \\ &\longrightarrow \text{serialize and hash the pre-audit seal} \\ &\longrightarrow \text{generate primary and stress targets} \\ &\longrightarrow \text{evaluate every frozen arm and claim gate.} \end{aligned} \tag{972}$$

The pre-audit seal digest is

$$\texttt{de2fc5bb3654eaca237d7cc3799570eb72fe9814657a3a55f9a42323361a92bd}. \tag{973}$$

The atlas certificate digest is

$$\texttt{22976d2546377b4afcfb0dc67e2489f2d84911527b24ac35cbdb0dc0bf9a77e8}. \tag{974}$$

63.4 Comparison arms

Arm	Role
Legacy fixed packet	Inherited parent using one coordinate-fixed rank-two packet after the supplied Block 10 packet is removed.
Endogenous atlas	Child recovering response blocks, searching each block for a repair-stable separator, and applying the frozen witness.
PCA packet	Equal-rank variance subspace control.
Full-space search	Unstructured separator search without a block atlas.
Random atlas	Same atlas size with random rank-two packets.
Shuffled response	Response observations are permuted relative to the target before atlas construction.
Global bridge	One global bridge, used as a misspecification and harm control.
Blind atlas	Correct child candidate is always committed, removing the witness.
True-chart oracle	Inadmissible oracle using the hidden chart label and target structure.

The controls separate several possible explanations. PCA tests whether ordinary variance is enough. Full-space search tests whether flexible separator search can replace functional packets. Random and shuffled controls test whether the response correspondence matters. Blind application tests whether the witness is causally necessary for protected selection.

63.5 Frozen witness

The selected atlas candidate commits only when all frozen conditions hold:

$$\widehat{g}_{\mathrm{val}} \geq 0.10, \tag{975}$$

$$\mathrm{LCB}(\widehat{g}_{\mathrm{val}}) \geq 0.05, \tag{976}$$

$$\widehat{r}_{\mathrm{ctx}} \leq 3 \times 10^{-4}, \tag{977}$$

$$\widehat{a}_{\mathrm{target}} \geq 0.72, \tag{978}$$

$$\widehat{a}_{\mathrm{repair\ inv}} \geq 0.96, \tag{979}$$

$$\widehat{b}_{\mathrm{chart}} \geq 0.20, \tag{980}$$

$$\widehat{s}_{\mathrm{model}} \leq 0.25, \tag{981}$$

$$\widehat{\gamma}_{\mathrm{min}} \geq 0.004, \tag{982}$$

$$\widehat{\varepsilon}_{\mathrm{comm}} \leq 0.16, \tag{983}$$

$$\widehat{\varepsilon}_{\mathrm{resp}} \leq 0.08. \tag{984}$$

The witness therefore evaluates both downstream repair quality and the internal atlas certificate. A high blind gain with unstable response geometry is not sufficient for commitment.

63.6 Frozen claim gates

The primary endogenous-atlas gate requires:

1. exact ancestor-policy and ancestor-receipt replay;
2. no supplied context packet or hidden atlas label in the child inputs;

3. response-algebra use recorded in the compiler receipt;
4. equal total query accounting;
5. conservative embedding of the inherited process;
6. positive paired child-parent gain with a frozen interval excluding zero;
7. zero negative paired-effect rate on the primary audit;
8. zero selected harmful commits;
9. response-fit, commutator, and spectral-gap thresholds;
10. selected context overlap above the legacy-packet overlap;
11. substantial failure of shuffled-response and random-atlas controls;
12. abstention or failure on declared nonidentifiable and strongly misspecified stress families.

64 Block 11 Results

64.1 Development atlas certificate

Across 36 development targets, the compiler records:

Quantity	Mean
Best recovered-block context overlap	0.896577
Selected-block context overlap	0.875926
Legacy fixed-packet context overlap	0.348901
Minimum spectral gap	0.015421
Maximum commutator residual	0.076885
Response-fit NRMSE	0.011765
Endogenous-atlas blind gain	0.279957
Legacy-packet blind gain	-0.256587

The certificate records that the supplied separator was removed, the response algebra was used, and the Block 10 grammar remains conservatively embedded.

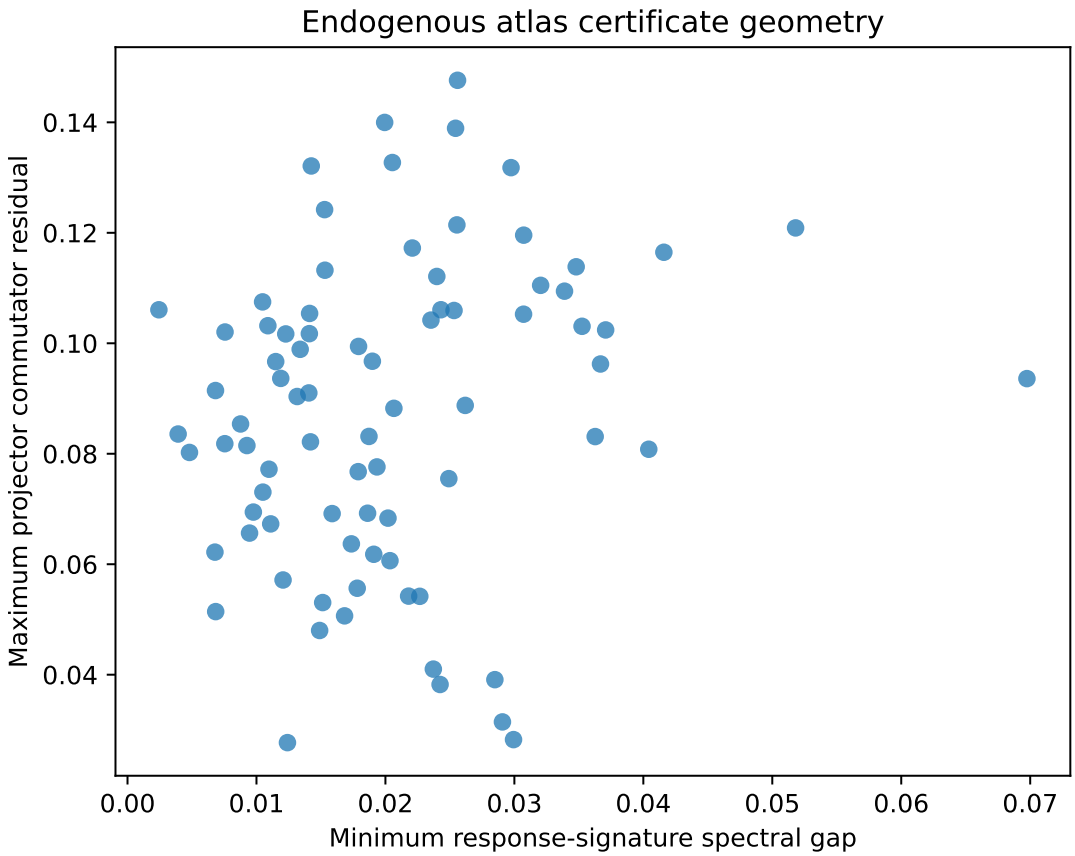


Figure 32: Development and audit atlas certificate. Intervention responses recover a target-specific functional cover whose selected block has substantially larger context overlap than the inherited fixed packet.

64.2 Primary paired foreign audit

The primary audit contains 80 targets from four foreign representation families. The main selected results are:

Process	Mean selected gain	Commit rate	Harm rate
Legacy fixed-packet parent	0.000000	0.0000	0.0000
Endogenous-atlas child	**0.207325**	0.2875	**0.0000**
PCA packet	0.053178	0.0750	0.0000
Full-space search	0.008591	0.0375	0.0125
Random atlas	0.000572	0.0125	0.0000
Shuffled response	0.016193	0.0250	0.0000
True-chart oracle	1.000000	–	0.0000

The paired effect is

$$\widehat{\Delta}_{3\to4} = 0.207325, \tag{985}$$

with frozen bootstrap interval

$$[0.131902,\ 0.287910] \tag{986}$$

and sign-flip estimate

$$p = 9.9999 \times 10^{-6}. \tag{987}$$

No primary target favors the parent. The positive paired-effect rate is 0.2875 and the remaining 0.7125 are abstention ties.

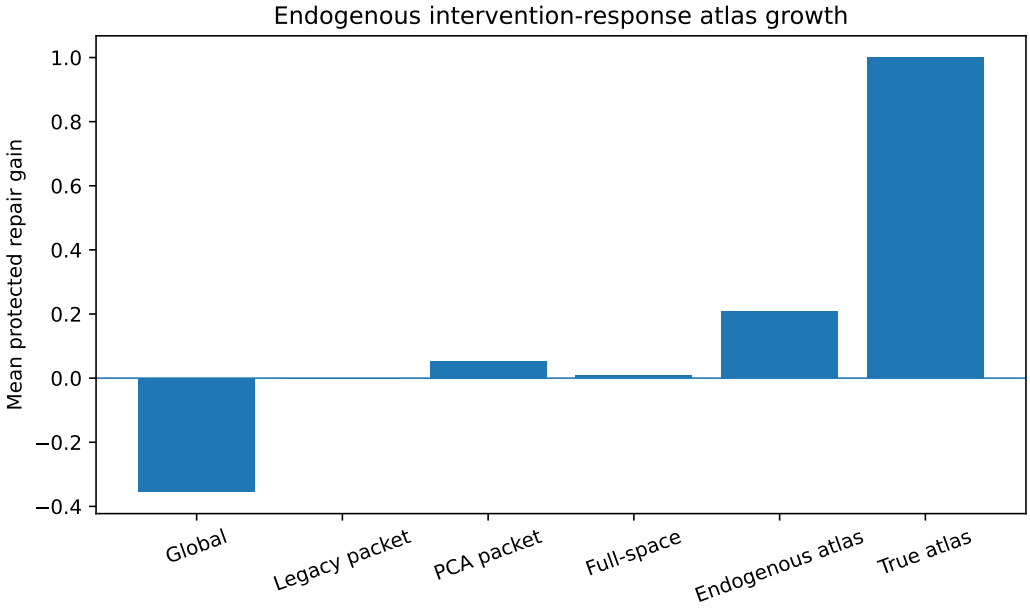


Figure 33: Primary selected protected gain under the matched 120-query envelope. The endogenous atlas child improves the inherited fixed-packet parent and all generic-packet controls while preserving zero selected harm.

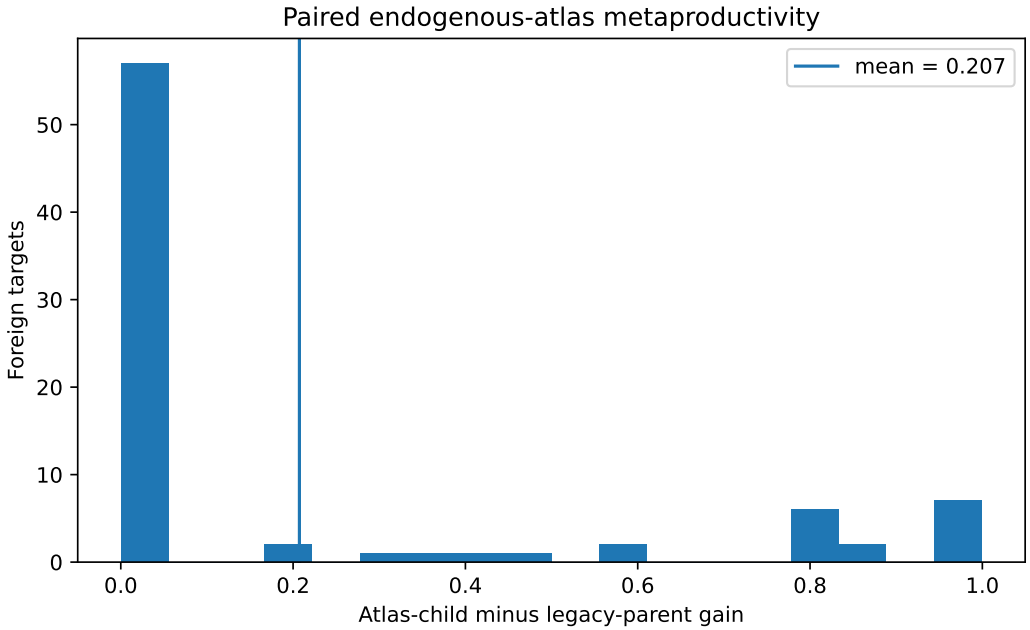


Figure 34: Episode-wise endogenous-child minus fixed-parent protected gain. Positive effects occur on the targets where the witness commits; the remaining targets are exact abstention ties.

64.3 Atlas geometry

The primary audit geometry is:

Quantity	Mean or mode
Best recovered-block context overlap	0.858886
Selected-block context overlap	0.821461
Legacy fixed-packet overlap	0.322091
Atlas target-gate accuracy	0.887562
Minimum spectral gap	0.020295
Maximum commutator residual	0.087389
Response-fit NRMSE	0.013046
Response-algebra rank, mode	3
Response-commutant dimension, mode	12

The overlap statistic is used only for audit interpretation, not for candidate selection. The compiler sees response operators and repair-validation evidence, not the hidden context projector.

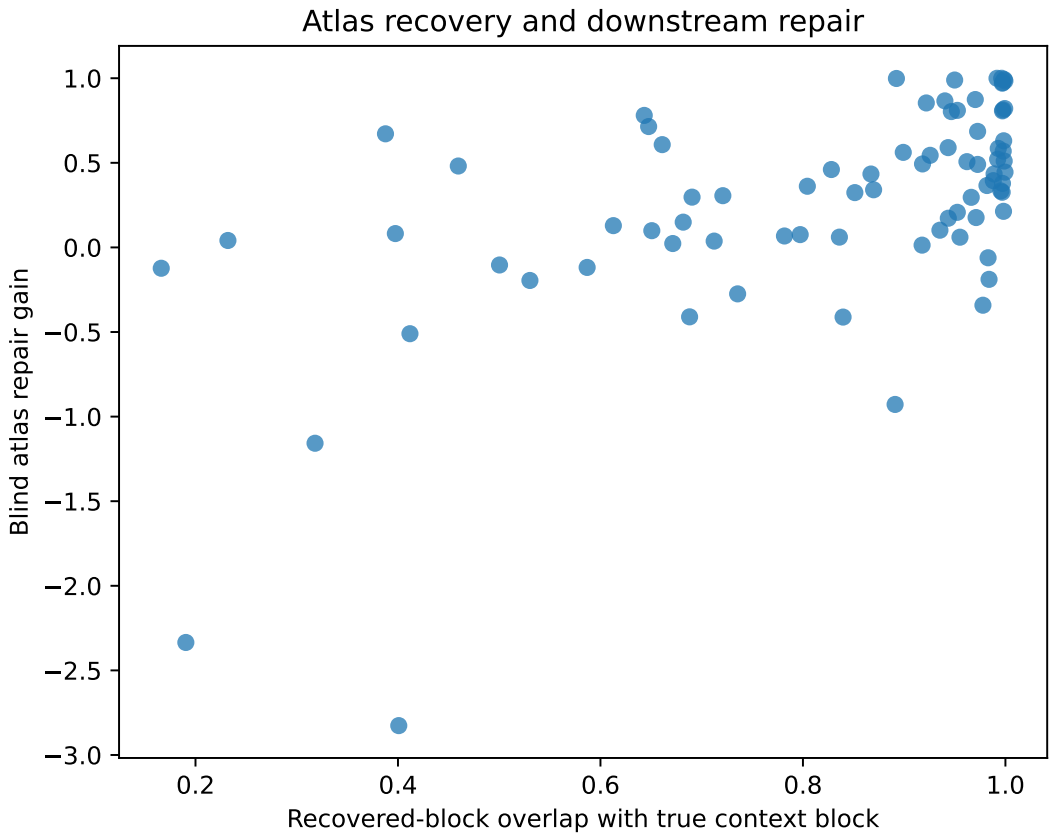


Figure 35: Recovered context overlap and protected gain. High overlap is neither the sole selection criterion nor a guarantee of commitment, but the selected atlas substantially outperforms the fixed packet in the audit geometry.

64.4 Family-level behavior

Family	Child gain	Child commit	Parent gain
Orthogonal six-dimensional	0.197559	0.35	0
Overcomplete eight-dimensional	0.576430	0.65	0
Noisy six-dimensional	0.055312	0.15	0
Mild cross-block coupling	0.000000	0.00	0

The exact and overcomplete families provide the strongest positive realization. Noisy targets produce smaller selective gains. Mild coupling lies outside the frozen witness envelope and produces abstention rather than a forced atlas claim.

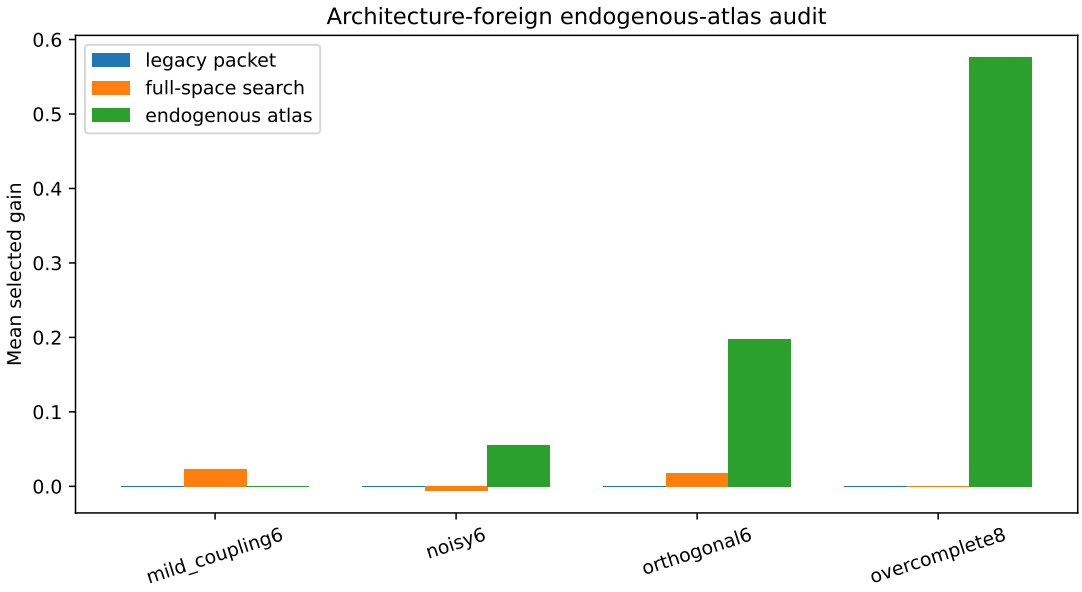


Figure 36: Family-level selected gain. The child commits selectively in exact, overcomplete, and some noisy targets, while abstaining on the mild-coupling family.

64.5 Mechanism controls and witness contribution

Blind application of the endogenous candidate has mean gain

$$0.271529 \tag{988}$$

but harmful-commit rate

$$0.1875. \tag{989}$$

After witness selection, mean gain becomes

$$0.207325 \tag{990}$$

and the harmful-commit rate becomes zero. The witness therefore removes an 18.75% blind-harm rate at a deliberate utility cost.

The other blind controls are substantially worse:

Blind arm	Mean gain	Harm rate
Endogenous atlas	0.271529	0.1875
Legacy packet	-0.279330	0.7500
PCA packet	-0.100204	0.5875
Random atlas	-1.332720	0.5625
Shuffled response	-0.252442	0.6500
Global bridge	-0.354808	0.9125

The unstructured full-space blind arm contains a small number of very large negative outliers and is therefore reported separately rather than used as a stable mean baseline.

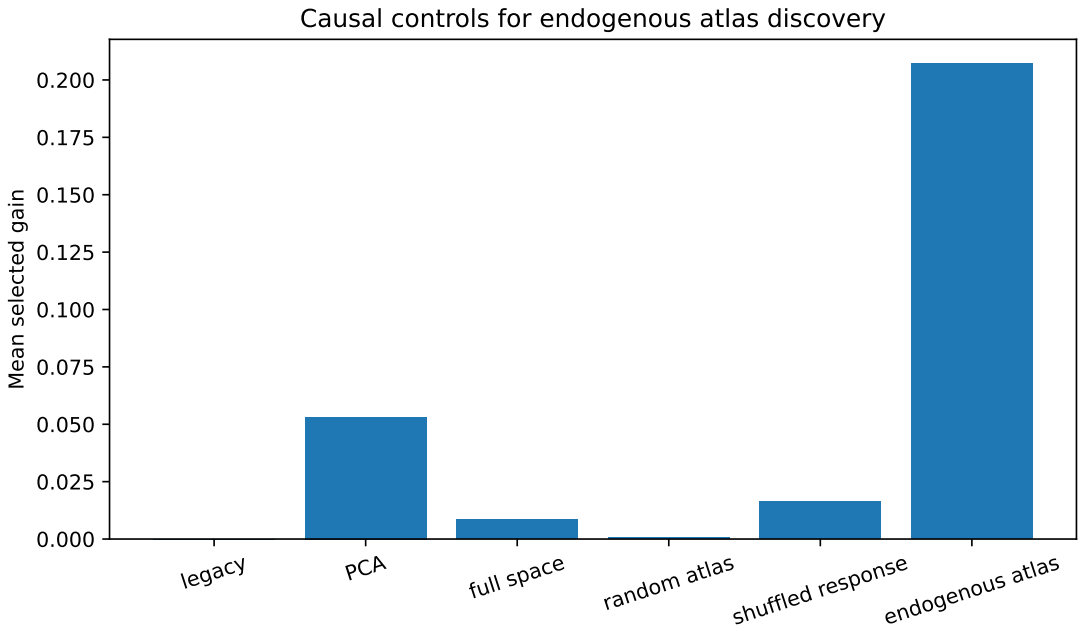


Figure 37: Mechanism and witness controls. Response correspondence, block structure, and frozen witness selection are all necessary for the declared primary result.

64.6 Stress boundaries

The stress audit contains degenerate signatures, strong cross-block coupling, smooth chart transitions, repair-dependent chart crossing, and noninjective target compression. The child has overall stress mean gain

$$0.027943, \tag{991}$$

commit rate 0.04, and zero selected harm. It abstains completely on the degenerate, strongly coupled, repair-crossing, and compressed families. One of five smooth targets is committed.

These negative results separate the current realization from a universal atlas claim:

- degenerate signatures require a new intervention capable of separating the merged response block;
- strong coupling requires a noncommuting or approximately reducing algebra rather than the block-scalar model;
- smooth transitions require soft charts, local manifolds, or a partition of unity;
- repair-dependent chart crossing requires typed control-flow transport through the repair;
- compression requires obstruction-faithful information-loss accounting.

64.7 Finite endogenous-atlas realization theorem

Conditional Theorem 64.1 (Block 11 finite realization)**.** *Within the frozen Block 11 benchmark, assume that:*

1. *the published Block 10 ancestor digests replay exactly;*

2. *the supplied context packet and hidden atlas labels are excluded from child construction;*
3. *the response operators are estimated only from the declared* 24 *response-query units;*
4. *parent and child each use exactly* 120 *total effect-query units;*
5. *the child is sealed before primary and stress target generation;*
6. *the conservative embedding in Theorem 62.8 is executable;*
7. *the paired effect in Equation (985) has the frozen interval in Equation (986);*
8. *the declared response, commutator, spectral, repair, and witness gates pass;*
9. *random-atlas, shuffled-response, PCA, full-space, blind, and stress controls retain their declared interpretations.*

Then the edge

$$K_3 \xrightarrow{\Pi_4} K_4 \tag{992}$$

is a finite, matched-resource, lineage-generated instance of endogenous intervention-response atlas growth. It is structurally stronger than the fixed-packet parent because atlas choice varies with target response geometry and enables positive protected gain unavailable to the inherited packet under the same query envelope.

Proof. Items 1 through 5 establish lineage continuity, audit isolation, and matched resources. Item 6 preserves the parent as an executable subgrammar. Items 7 and 8 establish positive foreign gain with the declared geometric and safety conditions. Item 9 rules out the principal packet-rank, variance, coordinate, correspondence, and blind-commit explanations inside the frozen finite fixture. The target-specific recovered projector is not available to the fixed-packet parent, so the accepted edge changes the atlas-construction component rather than merely selecting a different repair inside the inherited opportunity set. □

Block 11 result

Block 11 removes the supplied Block 10 context packet and replaces it with a finite intervention-response atlas compiler. The child recovers target-specific central response blocks up to permutation and within-block gauge, searches those blocks for a repair-stable chart separator, and commits only through a frozen witness. On 80 fresh foreign targets under an equal 120-query envelope, selected protected gain rises from 0 for the inherited fixed-packet parent to 0.207325 for the endogenous child, with frozen interval $[0.131902, 0.287910]$ and zero selected harmful commits. The result is a controlled finite endogenous-atlas edge, not unrestricted or spontaneous atlas discovery.

65 Block 11 Reproducibility and Package

65.1 Reference implementation

The released package contains:

- exact Block 10 ancestor policy and seal records;

- development, meta-validation, primary-audit, and stress generators;
- response-operator estimation and central spectral atlas recovery;
- approximate-commutant and commutator diagnostics;
- repair-relevant block and separator selection;
- legacy, PCA, full-space, random, shuffled-response, global, blind, and oracle controls;
- frozen witness policies and pre-audit receipts;
- deterministic result objects and seven publication figures;
- ten automated tests;
- an exact 49-file SHA-256 package manifest.

  The canonical execution prints:

```
SRIC_BLOCK11_ALL_SUITES=PASS
result_sha256=3e65e199d756a3fd742382744a96da9bb7aec9141ea1cd0431fbceada908afb1
parent_gain=0.000000
child_gain=0.207325
paired_effect=0.207325
context_overlap=0.821461
child_harm_rate=0.000000
.......... [100%]
SRIC_BLOCK11_PACKAGE_VALID
manifest_files=49
```

The exact Block 10 ancestor policy digest is

$$\texttt{48588442c27529fa38cbee5f6938e6c36263e4c650a5c08ab61365f440a27bb1}. \tag{993}$$

The Block 11 compiler digest is

$$\texttt{940105f8bf73a5528b5618066664004081066c94260f09c5f6672c9d54eab4f6}. \tag{994}$$

65.2 Claim boundary

The release establishes a finite, package-closed realization of endogenous atlas recovery from a declared intervention-response family. It does not establish:

- autonomous invention or selection of the intervention family;
- automatic determination of the number or dimension of functional blocks in an unrestricted model;
- recovery under arbitrary noncommuting, nonlinear, or time-varying response dynamics;
- spontaneous causal packets in pretrained language models;
- autonomous invention of chart number, soft gating, manifold coordinates, or kernel lifts;

- improvement of the outer lineage compiler;
- evaluator or witness self-improvement;
- open-ended recursive improvement, indefinite acceleration, unrestricted safety, consciousness, or sentience.

The next structural threshold is *response-language growth*. The lineage must encounter a failure of the current block-scalar, hard-chart atlas, certify whether the missing structure is an additional intervention, a noncommuting block interaction, a soft local manifold, a repair-dependent transition, or an information-loss obstruction, and then construct the corresponding new atlas or repair operator under a fresh sealed audit.

66 Endogenous Response-Language and Candidate-Constructor Growth

Block 11 recovers a finite functional atlas from a declared family of intervention-response operators. That result still assumes that the supplied response language separates the task-relevant blocks and that the inherited repair-constructor family is expressive enough once the relevant block has been found. Block 12 removes each assumption in a separate, sealed lineage edge.

The realized continuation is

$$K_4 \xrightarrow{\Pi_5} K_5 \xrightarrow{\Pi_6} K_6. \tag{995}$$

The first edge expands the intervention-response language. The second expands the repair-constructor grammar. The two edges are evaluated in different protected contexts and are not added into one scalar utility difference without an additional context transport.

66.1 Response languages and observational aliases

Let H be a finite-dimensional Hilbert space and let

$$\mathcal{R} = \{R_1, \dots, R_m\} \subseteq \mathrm{End}(H) \tag{996}$$

be a declared intervention-response language. Its unital response algebra is

$$\mathfrak{A}(\mathcal{R}) = \mathrm{alg}(I, R_1, \dots, R_m). \tag{997}$$

When adjoints are available, the corresponding adjoint-closed algebra is denoted $\mathrm{alg}^*(\mathcal{R})$.

Definition 66.1 (Block-scalar response signature)**.** Let

$$H = \bigoplus_{a=1}^{k} H_a \tag{998}$$

be a reducing decomposition for every R_j. The response language is block scalar when

$$R_j|_{H_a} = \lambda_{aj} I_{H_a} \tag{999}$$

for scalars λ_{aj}. The vector

$$\sigma_{\mathcal{R}}(H_a) = (\lambda_{a1}, \dots, \lambda_{am}) \tag{1000}$$

is the response signature of block H_a.

Definition 66.2 (Response alias)**.** Two reducing blocks H_a, H_b are response aliased under $\mathcal{R}$ when

$$\sigma_{\mathcal{R}}(H_a) = \sigma_{\mathcal{R}}(H_b). \tag{1001}$$

Theorem 66.3 (Polynomial persistence of a response alias)**.** *Suppose H_a and H_b are block-scalar response aliases under $\mathcal{R}$. Then every element $A \in \mathfrak{A}(\mathcal{R})$ acts by the same scalar on H_a and H_b. Consequently, no central idempotent in $\mathfrak{A}(\mathcal{R})$ separates the two blocks.*

Proof. Every element of $\mathfrak{A}(\mathcal{R})$ is a finite linear combination of words in the generators. On H_a, a word $R_{i_1} \cdots R_{i_\ell}$ acts by the scalar $\lambda_{ai_1} \cdots \lambda_{ai_\ell}$. Equality of the signature vectors makes the same scalar act on H_b. Linear combinations preserve the equality. A central idempotent separating the blocks would act as 1 on one and 0 on the other, contradicting the first conclusion. □

The theorem gives a strict grammar-relative no-go result. More samples from the same inherited intervention family cannot separate an exact alias because every polynomial or composition of the inherited operators preserves it.

Corollary 66.4 (No sample-size cure for an exact alias)**.** *If an atlas compiler is restricted to the central projectors of $\mathfrak{A}(\mathcal{R})$, then an exact response alias cannot be resolved by increasing the number of response observations while keeping $\mathcal{R}$ fixed.*

66.2 Separating interventions and conservative response refinement

Let $S \in \mathrm{End}(H)$ be a candidate intervention response. Define the expanded language

$$\mathcal{R}^+ = \mathcal{R} \cup \{S\}. \tag{1002}$$

The extension is conservative at the syntactic level because every old response expression remains available unchanged.

Definition 66.5 (Separating response operator)**.** A candidate S separates an aliased pair (H_a, H_b) when both subspaces reduce S and

$$S|_{H_a} = \alpha I_{H_a}, \qquad S|_{H_b} = \beta I_{H_b}, \qquad \alpha \neq \beta. \tag{1003}$$

Theorem 66.6 (Central refinement by a separating intervention)**.** *Suppose H_a, H_b are aliased under $\mathcal{R}$, and S satisfies Equation (1003). If the joint signature of every remaining block is distinct from those of H_a and H_b under $\mathcal{R}^+$, then the minimal central projectors of $\mathfrak{A}(\mathcal{R}^+)$ separate H_a and H_b.*

Proof. Choose a generic linear combination

$$G = \sum_{j=1}^{m} c_j R_j + c_{m+1} S \tag{1004}$$

whose induced scalar values on the distinct joint signatures are pairwise distinct. For each block value γ_a, the Lagrange polynomial

$$p_a(z) = \prod_{b \neq a} \frac{z - \gamma_b}{\gamma_a - \gamma_b} \tag{1005}$$

satisfies $p_a(G) = P_a$, the orthogonal projector onto H_a. Therefore $P_a, P_b \in \mathfrak{A}(\mathcal{R}^+)$ and belong to its center. □

Definition 66.7 (Response-language expansion certificate)**.** A response-language expansion certificate is a tuple

$$\Pi_{\mathrm{resp}} = (\mathcal{R}, S, \mathfrak{A}, \mathfrak{A}^+, P_{\mathrm{alias}}, \gamma, \varepsilon_{\mathrm{comm}}, \varepsilon_{\mathrm{fit}}, \Pi_{\mathrm{foreign}}), \tag{1006}$$

where P_{alias} records the inherited merged block, γ is the minimum separating spectral gap, $\varepsilon_{\mathrm{comm}}$ is a commutator residual, $\varepsilon_{\mathrm{fit}}$ is response-model error, and Π_{foreign} records held-out repair effects.

The expansion is structurally substantive only when the new operator refines the inherited central partition and the refined block enables a task-relevant correction unavailable to the inherited atlas grammar.

66.3 Approximate spectral recovery

Let G be the ideal generic response combination from the proof of Theorem 66.6, and let

$$\widehat{G} = G + E \tag{1007}$$

be its empirical estimate. Let P be a spectral projector associated with an eigenvalue cluster separated from the rest of the spectrum by gap $\gamma > 0$.

Proposition 66.8 (Response-projector perturbation)**.** *If $\|E\|_2 < \gamma/2$, then the corresponding empirical spectral projector $\widehat{P}$ satisfies a Davis–Kahan type bound*

$$\left\|\widehat{P} - P\right\|_2 \leq \frac{2\,\|E\|_2}{\gamma}. \tag{1008}$$

The empirical witness should therefore track both a minimum spectral gap and response-estimation error. A new response operator with a theoretically distinct but extremely small signature margin need not be practically useful.

66.4 Finite response-language compilation

Let $\mathscr{S}$ be a finite typed candidate-intervention grammar. For candidate $S \in \mathscr{S}$, define a development score

$$\mathrm{Score}_{\mathrm{resp}}(S) = \widehat{M}_{\mathrm{foreign}}(S) - \lambda_c C(S) - \lambda_{\mathrm{comm}}\varepsilon_{\mathrm{comm}}(S) - \lambda_{\mathrm{fit}}\varepsilon_{\mathrm{fit}}(S) - \lambda_{\mathrm{risk}}\widehat{R}(S). \tag{1009}$$

The compiler selects a candidate only after verifying that it adds a nontrivial response-signature distinction and preserves the inherited response language as an executable subgrammar.

Proposition 66.9 (Finite response-language selection)**.** *If $\mathscr{S}$ is finite, the development and meta-validation splits are fixed before audit access, and ties are resolved by a predeclared deterministic*

rule, then the selected intervention and its witness gate are measurable with respect to pre-audit evidence.

This proposition is procedural rather than a general learning guarantee. It ensures that the later audit evaluates a frozen response-language child rather than participating in its construction.

67 Repair-Constructor Expansion

The second Block 12 edge assumes the separating response language from the first edge and asks a different question: even after the relevant repair block is identified, can the inherited repair constructors express the required correction?

67.1 Affine repair operad and its closure

Let $\mathcal{C}_{\mathrm{aff}}(U, V)$ denote affine maps

$$f(x) = Ax + b \tag{1010}$$

from authorized state space U to defect or target space V.

Theorem 67.1 (Affine closure under composition)**.** *Every finite composition of affine repair constructors is affine. Parallel affine packet repairs followed by affine synthesis also remain affine.*

Proof. For $f(x) = Ax + b$ and $g(x) = Cx + d$,

$$(g \circ f)(x) = CAx + Cb + d, \tag{1011}$$

which is affine. Induction proves finite closure. Direct sums and affine synthesis preserve the same form. □

Thus an arbitrarily deep composition of inherited affine repair schemas cannot create a genuinely quadratic correction direction.

67.2 A Hessian obstruction to affine repair

Let P be the recovered repair packet, let $z = P^{\top}x$, and consider a target correction

$$F(x) = Ax + b + (z^{\top}Bz)v, \tag{1012}$$

where $B = B^{\top} \neq 0$ and $v \neq 0$.

Theorem 67.2 (Nonzero-Hessian affine obstruction)**.** *No affine map agrees with F on any nonempty open subset of the packet coordinates.*

Proof. Every affine map has identically zero second derivative. The Hessian of the nonlinear term in Equation (1012) is the nonzero tensor $2B \otimes v$. Equality on an open set would imply equality of second derivatives there, a contradiction. □

The obstruction has an exact population form under a centered Gaussian packet distribution.

Theorem 67.3 (Best affine residual for a quadratic packet repair)**.** *Let $z \sim N(0, I_r)$, $B = B^\top$, and define*

$$q(z) = z^\top B z. \tag{1013}$$

The best L^2 affine approximation to $q(z)v$ is the constant $(\operatorname{tr} B)v$, and its minimum squared residual is

$$\inf_{a,L} \mathbb{E} \left\| q(z)v - (a + Lz) \right\|^2 = 2 \left\| B \right\|_F^2 \left\| v \right\|^2. \tag{1014}$$

Proof. For centered Gaussian z, the centered quadratic $q(z) - \operatorname{tr} B$ is orthogonal in L^2 to constants and all linear forms. Its variance is $2 \left\| B \right\|_F^2$. Multiplying by v gives Equation (1014). □

This result supplies a function-class obstruction rather than a sample-size diagnosis. More affine training data may improve estimation but cannot remove the residual in Equation (1014).

67.3 Block-local quadratic repair grammar

Let $U_P \in \mathbb{R}^{d \times r}$ have orthonormal columns spanning the recovered repair packet. Define packet coordinates

$$z = U_P^\top (x - c). \tag{1015}$$

The degree-two block-local grammar consists of maps

$$F_{A,b,C}(x) = Ax + b + C \operatorname{vech}(zz^\top), \tag{1016}$$

where vech lists the $r(r+1)/2$ distinct symmetric quadratic monomials.

Proposition 67.4 (Conservative embedding of affine repairs)**.** *The affine grammar embeds into the block-local quadratic grammar by setting $C = 0$.*

Proposition 67.5 (Within-packet gauge invariance)**.** *If packet coordinates change by an orthogonal basis transformation $z' = Q^\top z$, then the class of functions in Equation (1016) is unchanged.*

Proof. Every quadratic form $z^\top B z$ becomes

$$z'^\top (Q^\top B Q) z'. \tag{1017}$$

The vector space of symmetric quadratic forms is therefore carried isomorphically to itself under the induced action of Q on $\operatorname{Sym}^2(P^\top H)$. □

This is important because Block 11 recovers response blocks only up to within-block gauge. The full symmetric degree-two packet grammar does not require a privileged coordinate basis inside the recovered block.

67.4 Constructor novelty and parsimony

Let $\mathcal{C}_t$ be the inherited repair-constructor grammar and $\mathcal{C}_t^+$ a candidate extension. The extension is structurally new when its realized correction image contains a task-relevant direction outside the closure of the old grammar.

Definition 67.6 (Constructor-expansion certificate)**.** A constructor-expansion certificate is a tuple

$$\Pi_{\mathrm{con}} = (\mathcal{C}_t, \mathcal{C}_t^+, \eta_{\mathrm{old}}, \eta_{\mathrm{new}}, \mathsf{H}, \kappa, C_{\mathrm{model}}, \Pi_{\mathrm{foreign}}), \tag{1018}$$

where H is a nonzero higher-order derivative witness, κ is held-out obstruction reduction, and C_{model} records constructor complexity.

A full-space quadratic grammar contains the block-local quadratic grammar. Therefore a finite experiment need not establish block-local superiority over a sufficiently well-regularized full quadratic control. The narrower claim is that a degree-two constructor is required, while localization provides a typed and lower-dimensional realization. In Block 12, the compiler uses a predeclared parsimony tie-break among constructors whose development gain lies within 0.02 of the best candidate.

Proposition 67.7 (Finite constructor selection with conservative tie-break)**.** *Suppose a finite constructor family is evaluated on fixed development evidence. Let $G^\star$ be the largest held-out development gain and let*

$$\mathcal{C}_\tau = \{C : G(C) \geq G^\star - \tau\}. \tag{1019}$$

Selecting the minimum-complexity constructor in $\mathcal{C}_\tau$ is deterministic and preserves an explicit near-optimality bound of τ on the development criterion.

68 Block 12 Experimental Contract

68.1 Two separately sealed contexts

Block 12 contains two separate protected contexts.

Context A: response-language growth. The inherited Block 11 process receives two response operators whose signatures alias the context and repair blocks. A finite compiler may add exactly one operator from a declared five-candidate intervention grammar. The selected operator and witness gate are sealed before the primary audit.

Context B: candidate-constructor growth. The process inherits the separating response language and recovered repair packet from Context A. Its parent is allowed only affine repairs. The child may select one constructor from a finite family containing affine, packet-quadratic, wrong-packet quadratic, random-packet quadratic, and full-space quadratic forms. Parent and child receive separately compiled witness gates.

The two paired effects are context indexed. Their sum is not interpreted as one scalar increase in a fixed environment.

68.2 Response-language fixture

The hidden source space has three unknown rank-two blocks:

$$H = H_{\mathrm{ctx}} \oplus H_{\mathrm{rep}} \oplus H_{\mathrm{nui}}. \tag{1020}$$

The inherited signature matrix is

$$\Sigma_0 = \begin{bmatrix} 1.25 & 0.72 \\ 1.25 & 0.72 \\ 0.80 & 1.30 \end{bmatrix}, \tag{1021}$$

where rows correspond to context, repair, and nuisance blocks. The first two rows coincide. By Theorem 66.3, every element of the inherited response algebra acts identically on the context and repair blocks.

The finite candidate language is:

Candidate	Context	Repair	Nuisance
split-context-repair	1.42	0.61	1.03
duplicate-parent	1.18	1.18	0.91
nuisance-rescale	0.84	0.84	1.37
weak-context-repair	1.015	0.985	1.00
cross-coupled	1.38	0.65	1.02

The cross-coupled candidate additionally receives a noncommuting symmetric perturbation.

68.3 Matched response-query envelope

Parent and child each receive exactly 144 deployment query units:

Process	Response	Bridge fit	Chart validation	Witness	Total
Inherited parent	72	36	18	18	144
Expanded-language child	72	36	18	18	144

The parent allocates 36 response observations to each of two inherited interventions. The child allocates 24 observations to each of three interventions. The total response budget is unchanged.

The response-language compiler uses 32 development targets per candidate, followed by 24 disjoint meta-validation targets for witness selection. The primary audit contains 96 foreign targets from four family identifiers absent from development. A separate 24-target stress suite contains degenerate signatures, strong cross-block coupling, and high response noise.

68.4 Constructor fixture and matched budget

The constructor context uses the recovered repair packet and a target repair of the form

$$F(x) = Ax + b + (z^\top Bz)v, \qquad z = U_{\mathrm{rep}}^\top x, \tag{1022}$$

with $B \neq 0$ and $v \neq 0$. The parent affine grammar is structurally obstructed by Theorem 67.2.

Parent and child each receive 160 query units:

Process	Response atlas	Constructor fit	Validation	Witness	Total
Affine parent	48	88	0	24	160
Quadratic child	48	64	24	24	160

The parent receives more direct constructor-fitting observations. The child spends 24 observations on constructor validation.

The compiler evaluates five constructor classes on 32 development targets each and selects witness thresholds on 24 disjoint meta-validation targets. The primary audit contains 96 foreign targets. A separate 24-target stress suite includes cubic contamination, weak quadratic signal, and noninjective compression.

68.5 Chronology and seal

The package chronology is:

$$\begin{aligned} &\text{replay Block 11 ancestor policy and seal} \\ &\longrightarrow \text{generate development and meta-validation evidence} \\ &\longrightarrow \text{compile response and constructor certificates} \\ &\longrightarrow \text{serialize gates, compiler ledger, and pre-audit seal} \\ &\longrightarrow \text{generate the two primary audits and stress suites} \\ &\longrightarrow \text{evaluate all frozen controls and claim gates.} \end{aligned} \tag{1023}$$

The pre-audit seal explicitly records that audit fields are absent.

68.6 Frozen comparison arms

Response-language arms. The primary response experiment includes:

1. the inherited two-operator parent;
2. the selected separating-intervention child;
3. a duplicate operator that preserves the alias;
4. a weak-separation operator;
5. a shuffled-response correspondence control;
6. an equal-rank random-atlas control;
7. blind application without witness abstention.

Constructor arms. The constructor experiment includes:

1. an independently gated affine parent;
2. the selected repair-block quadratic child;
3. a directly supervised affine oracle fitted with 1600 pairs per target;
4. wrong-block and random-block quadratic controls;
5. shuffled quadratic features;
6. a full-space quadratic control;
7. stress families outside the declared degree-two injective regime.

69 Block 12 Results

69.1 Response-language development certificate

The finite compiler selects

`split_context_repair`

from the five-candidate language. Development summaries are:

Candidate	Filtered gain	Positive rate	Min. gap	Commutator
split-context-repair	0.934377	1.000000	0.066822	0.001421
duplicate-parent	−0.062500	0.000000	0.008805	0.001136
nuisance-rescale	−0.062500	0.000000	0.009193	0.001247
weak-context-repair	0.416493	0.656250	0.017536	0.001162
cross-coupled	−0.541985	0.250000	0.111187	0.014182

The selected candidate is the only stable, high-margin response extension that consistently separates the inherited alias and supports repair.

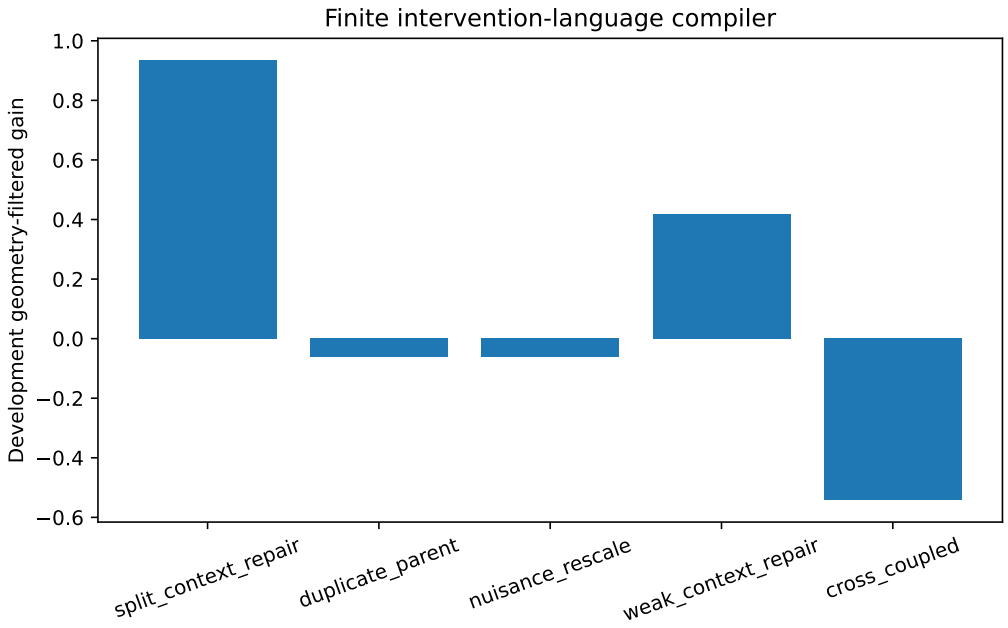


Figure 38: Development comparison for the finite intervention-language compiler. The selected response operator combines high repair gain with a stable separating spectral margin.

The frozen response gate requires witness gain at least 0.05, witness NRMSE at most 0.30, validation gain at least 0.05, validation NRMSE at most 0.25, minimum response gap at least 0.015, commutator residual at most 0.015, and minimum chart support at least 8.

69.2 Primary response-language audit

On 96 fresh foreign targets:

Process	Mean gain	Median	Commit	Positive	Harm
Inherited response parent	0.000000	0.000000	0.000000	0.000000	0.000000
Expanded-language child	0.846985	0.969182	0.906250	0.906250	0.000000

The paired effect is

$$\widehat{\Delta}_{4\to5} = 0.846985, \tag{1024}$$

with frozen interval

$$[0.786100, 0.901073] \tag{1025}$$

and sign-flip estimate

$$p = 9.9999 \times 10^{-6}. \tag{1026}$$

No primary target favors the inherited parent. The remaining 9.375% of pairs are abstention ties.

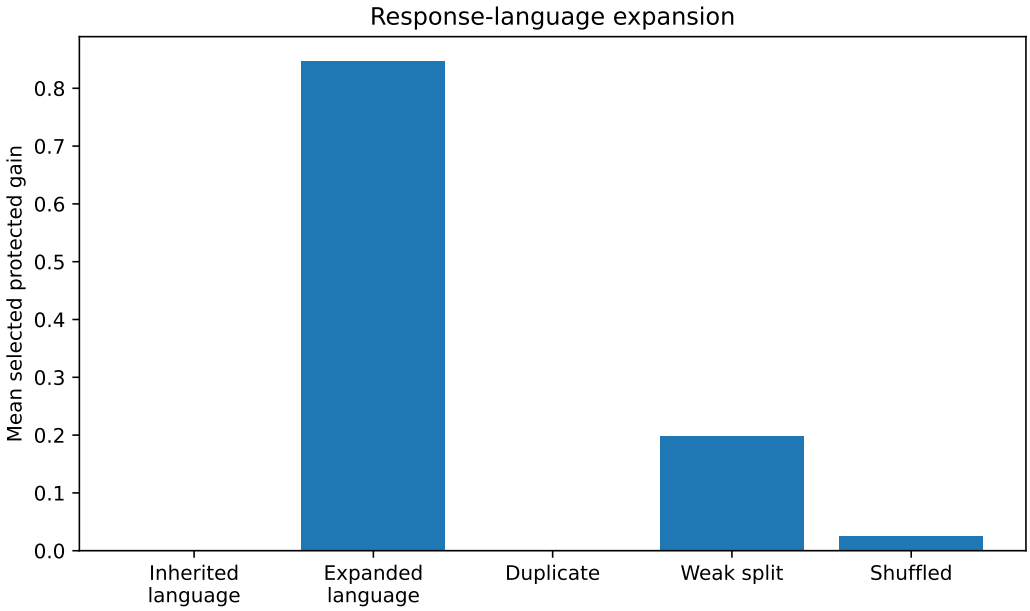


Figure 39: Primary response-language result and controls. Duplicate and shuffled response information do not reproduce the expanded-language child.

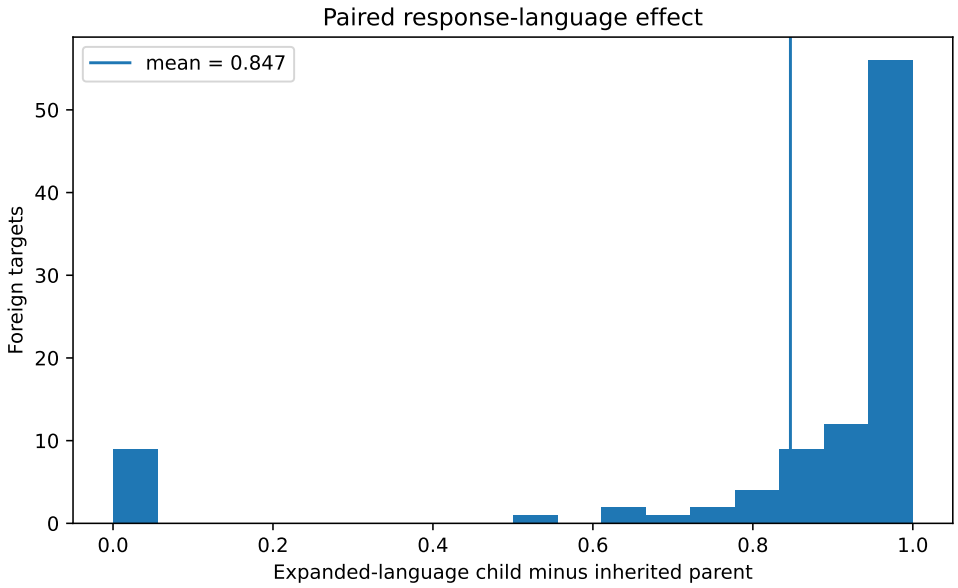


Figure 40: Episode-wise paired effect for the response-language edge. Positive mass corresponds to witnessed commitments; zeros are abstention ties.

69.3 Response geometry and controls

The inherited atlas has modal block ranks

$$(2, 4), \tag{1027}$$

while the expanded language recovers

$$(2, 2, 2). \tag{1028}$$

Mean audit geometry is:

Quantity	Mean
Child context-block overlap	0.999843
Parent rank-two context overlap	0.000146
Child minimum spectral gap	0.064981
Child commutator residual	0.001826

The overlap metric is retained for audit interpretation only. The compiler uses response geometry and repair validation, not hidden projectors.

Control results are:

Control	Mean selected gain	Commit	Harm
Duplicate response	0.000000	0.000000	0.000000
Weak separation	0.197231	0.260417	0.000000
Shuffled response	0.025225	0.062500	0.000000
Random atlas	0.000000	0.000000	0.000000
Blind selected intervention	0.904608	1.000000	0.000000

The blind child is not harmful on the primary response audit, so the witness primarily contributes selective abstention rather than harm removal in this first edge. The stress suite has commit rate 0.208333 and zero selected harm. It abstains completely on exact signature degeneracy and strong coupling; some high-noise cases remain inside the gate.

69.4 Constructor development certificate

The constructor compiler records:

Constructor	Development gain	Validation gain	Fit condition
Affine	0.320296	0.000000	1.791226
Repair-block quadratic	0.998022	0.997731	5.103054
Context-block quadratic	0.284136	0.226411	4.851183
Random-block quadratic	0.390305	0.393501	4.810982
Full-space quadratic	0.999456	0.999169	14.785843

The full-space quadratic class obtains the largest raw development gain, but the block-local quadratic constructor lies within the predeclared 0.02 near-optimality window and has a much smaller typed quadratic feature space. The compiler therefore selects

`repair_quadratic`

by the frozen parsimony rule.

This result should not be interpreted as block-local superiority over every full quadratic model. The full-space quadratic control remains an essentially exact alternative inside the finite fixture. The supported structural claim is degree-two constructor growth, with repair-block localization providing the selected parsimonious realization.

69.5 Primary constructor audit

On 96 fresh foreign targets:

Process	Mean gain	Median	Commit	Positive	Harm
Independently gated affine parent	0.289123	0.387596	0.656250	0.645833	0.010417
Repair-block quadratic child	0.997511	0.998251	1.000000	1.000000	0.000000

The paired effect is

$$\widehat{\Delta}_{5\to 6} = 0.708388, \tag{1029}$$

with frozen interval

$$[0.659635, 0.757198]. \tag{1030}$$

The sign-flip estimate is 9.9999×10^{-6}, and no primary target favors the affine parent.

A directly supervised affine oracle fitted with 1600 same-target pairs per target obtains mean blind gain

$$0.466578. \tag{1031}$$

The child exceeds that stronger function-class control by mean

$$0.530933, \tag{1032}$$

with paired bootstrap interval

$$[0.506587, 0.556639]. \tag{1033}$$

This supports the nonzero-Hessian obstruction rather than a small-sample explanation for the affine parent.

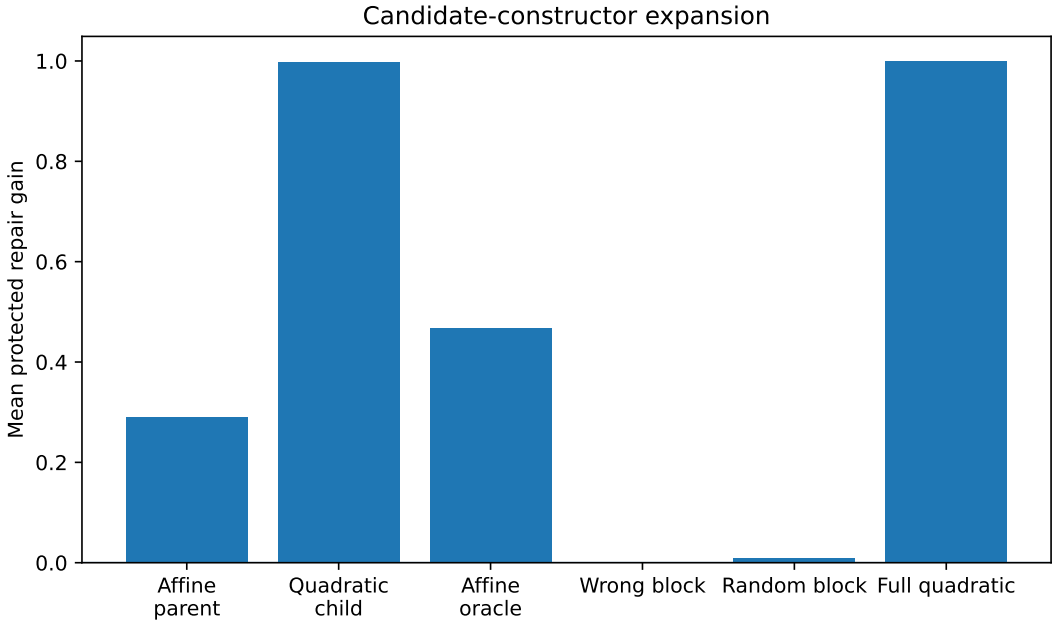


Figure 41: Candidate-constructor result and controls. The full quadratic control matches the selected child, preventing a claim of unique block-local superiority; wrong-block and shuffled-feature controls fail.

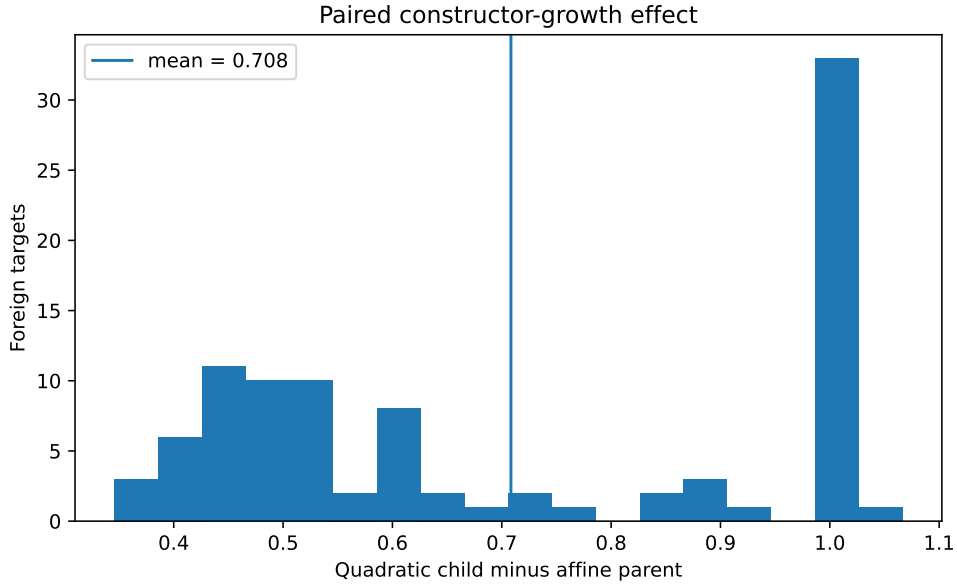


Figure 42: Paired child-minus-parent effect for the constructor edge. Every primary pair is positive.

69.6 Constructor controls and stress boundary

The principal constructor controls are:

Control	Mean selected gain	Commit	Harm
Wrong-block quadratic	0.000000	0.000000	0.000000
Random-block quadratic	0.009323	0.010417	0.000000
Shuffled quadratic features	0.000000	0.000000	0.000000
Full-space quadratic	0.999014	1.000000	0.000000

The full quadratic control shows that the essential missing structure is degree two. The wrong-block, random-block, and shuffled-feature controls show that arbitrary nonlinear capacity does not reproduce the typed constructor under the same fitting procedure.

The stress suite has overall commit rate 0.333333, mean selected gain 0.332051, and zero selected harm. The child abstains on the strong cubic-contamination and compressed families, while continuing to solve the weak-quadratic family. This boundary is appropriate: weak signal remains inside the degree-two grammar, whereas cubic and information-losing cases require different extensions.

69.7 Two finite structural lineage theorems

Conditional Theorem 69.1 (Finite response-language growth)**.** *Within the frozen Block 12 response benchmark, assume that:*

1. *the published Block 11 ancestor policy and seal replay exactly;*
2. *the inherited response signatures satisfy the exact alias in Equation (1021);*
3. *parent and child each consume exactly* 144 *query units;*
4. *the selected response operator and witness gate are sealed before primary audit generation;*
5. *the expanded language conservatively contains the inherited response algebra;*
6. *the response gap, commutator, support, validation, and witness gates pass;*
7. *the paired effect and interval are those in Equations (1024) and (1025);*
8. *duplicate, weak, shuffled, random-atlas, and stress controls retain their declared interpretations.*

Then

$$K_4 \xrightarrow{\Pi_5} K_5 \tag{1034}$$

is a finite, matched-resource, lineage-generated response-language expansion. The child adds a response distinction absent from every composition of the inherited operators and uses the resulting central refinement to improve foreign protected repair performance.

Proof. The alias theorem shows that the inherited response algebra cannot separate the context and repair blocks. The selected operator refines that central block by Theorem 66.6. Matched query accounting and the pre-audit seal establish the experimental comparison. The paired audit establishes positive foreign gain, while the controls rule out duplicate information, weak separation, correspondence shuffling, and an arbitrary equal-rank atlas as explanations inside the finite fixture. □

Conditional Theorem 69.2 (Finite repair-constructor growth)**.** *Within the frozen Block 12 constructor benchmark, assume that:*

1. *the response-language child from Theorem 69.1 supplies the inherited repair packet;*
2. *the true repair has a nonzero quadratic Hessian on that packet;*
3. *parent and child each consume exactly* 160 *query units;*
4. *parent and child receive separately compiled frozen witness gates;*
5. *the affine parent remains embedded by Theorem 67.4;*
6. *the paired effect and interval are those in Equations* (1029) *and* (1030)*;*
7. *the directly supervised affine oracle retains the gap in Equation* (1033)*;*
8. *wrong-block, random-block, shuffled-feature, full-quadratic, and stress controls retain their declared interpretations.*

Then

$$K_5 \xrightarrow{\Pi_6} K_6 \tag{1035}$$

is a finite, matched-resource, lineage-generated expansion of the repair-constructor grammar from affine to degree-two packet-local repair.

Proof. The inherited affine operad is closed by Theorem 67.1. The nonzero Hessian makes the target repair unavailable to that operad by Theorem 67.2. The selected quadratic grammar contains the affine parent and adds the missing degree-two response. The paired audit, same-target affine-oracle comparison, and typed controls establish task-relevant structural gain inside the declared finite benchmark. □

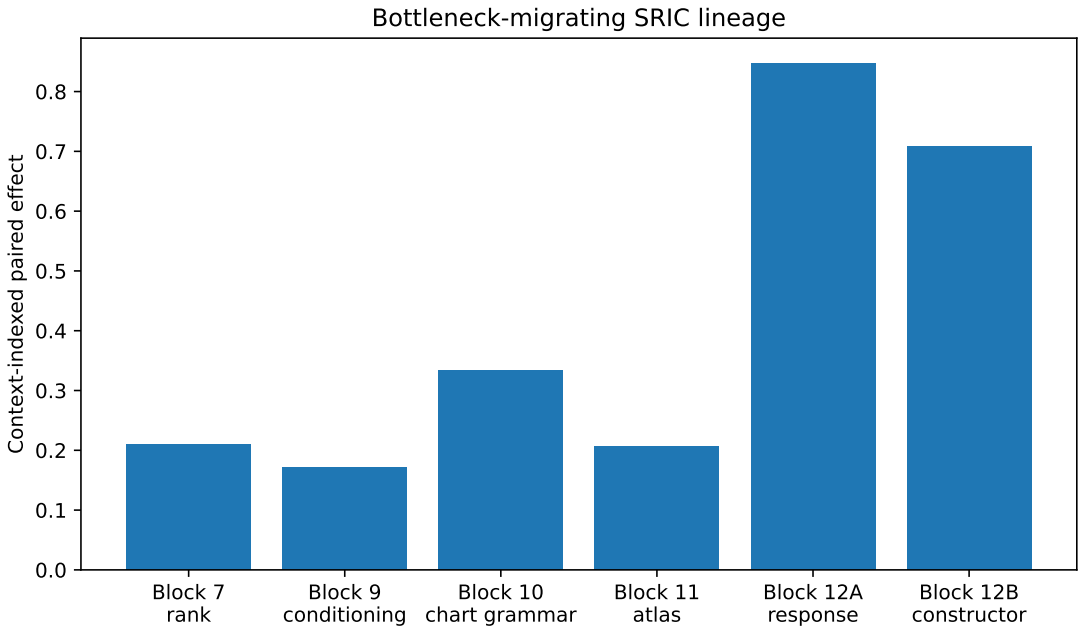


Figure 43: Context-indexed paired effects for the realized bottleneck-migrating lineage. Heights are not added into one fixed-context utility.

Block 12 result

Block 12 realizes two distinct structural lineage edges. The first adds a separating response operator because the inherited intervention algebra provably aliases two task-relevant blocks. Under an equal 144-query envelope, selected gain rises from 0 to 0.846985. The second adds a degree-two repair constructor because the inherited affine operad has a nonzero-Hessian obstruction. Under an equal 160-query envelope, gain rises from 0.289123 to 0.997511, and the child exceeds a 1600-pair same-target affine oracle by 0.530933. These are finite response-language and constructor-growth results, not unrestricted intervention invention or open-ended program synthesis.

70 Block 12 Reproducibility and Package

70.1 Reference implementation

The released package contains:

- exact Block 11 ancestor policy and seal files;
- separate response-language and constructor development, meta-validation, primary-audit, and stress generators;
- finite intervention and constructor compilers;
- response-algebra estimation, spectral atlas recovery, and commutator diagnostics;
- affine, packet-quadratic, full-quadratic, wrong-packet, random, duplicate, weak, and shuffled controls;

- separately compiled parent and child witness gates;
- canonical JSON and CSV result objects;
- six deterministic publication figures;
- twelve automated tests;
- an exact 41-file SHA-256 package manifest.

The canonical execution prints:

```
SRIC_BLOCK12_ALL_SUITES=PASS
result_sha256=4ede53f590ddf477711fc5b1b55e43fb96d9450cc57c0d020332fb6daf712bd1
response_parent_gain=0.000000
response_child_gain=0.846985
response_effect=0.846985
constructor_parent_gain=0.289123
constructor_child_gain=0.997511
constructor_effect=0.708388
............ [100%]
SRIC_BLOCK12_PACKAGE_VALID
manifest_files=41
```

The frozen response certificate digest is

$$\texttt{73080b6681940b79e390f7ecd979cb493c740bfab9be7b231002b44c387dc99d}. \tag{1036}$$

The frozen constructor certificate digest is

$$\texttt{d8be9c8e1c0e6ccb5e64f53700c8a59f45c8402e9a395ba7d72db97130010016}. \tag{1037}$$

The compiler digest is

$$\texttt{f51f5152dae06791fb87a6926162bc5e1965bc7e1bedae0a2957b9bbed82f475}. \tag{1038}$$

The pre-audit seal digest is

$$\texttt{75758911e1449620c32ed00b5c959c66b131956f2c79cf1127e15adfbffbd6cd}. \tag{1039}$$

70.2 Claim boundary

The release establishes:

1. a finite selection of one separating response operator from a declared intervention grammar;
2. a central atlas refinement unavailable to every composition of the inherited response operators;
3. a foreign repair improvement under matched response-query resources;
4. a finite constructor expansion from affine to degree-two repair on a recovered packet;
5. a function-class comparison against a much larger same-target affine training set;

6. conservative embeddings of both inherited grammars;
7. separately sealed audits, controls, receipts, and package replay.

It does not establish:

- unrestricted invention of interventions outside the finite candidate language;
- autonomous determination of intervention cost, feasibility, or physical side effects;
- general recovery for noncommuting, nonlinear, time-varying, or partially observed response algebras;
- unrestricted program synthesis or neural-weight constructor discovery;
- unique superiority of the block-local quadratic grammar over the full quadratic control;
- improvement of the outer lineage compiler;
- autonomous evaluator or witness evolution;
- open-ended recursive improvement, indefinite acceleration, unrestricted safety, consciousness, or sentience.

The next threshold is *compiler metaproductivity*. The current outer compiler is still externally designed to search one finite intervention language and one finite constructor family. A later descendant must improve the compiler that diagnoses the obstruction and chooses which grammar to expand, then transfer that improvement across foreign parent processes under matched compilation resources.

71 Proof-Carrying Automated Research Compilers

Blocks 7 through 12 improve components of a repair process while an outer lineage compiler remains externally fixed. Block 13 moves one level outward. The modified object is now the process that reads prior research evidence, proposes candidate descendants, allocates experiments, binds proposals to code, and selects a validated result.

The realized compiler edge is

$$\mathfrak{L}_6 \xrightarrow{\Pi_7^{\Gamma}} \mathfrak{L}_7, \tag{1040}$$

where the active repair grammar, candidate-method language, evaluator portfolio, integrity monitor, and experiment budget remain fixed, but the research-allocation compiler changes from Γ_0 to Γ_1. The edge is therefore compiler metaproductivity inside a finite declared research language, not unrestricted automated science.

The design is informed by the automated alignment researcher harness of Chen, Wen, and Kirchner, which combines a shared literature review, parallel researchers, results-free mini-papers, exact code review, fixed training budgets, separate evaluation, a shared finding forum, capability gates, and trajectory monitoring [13]. Block 13 does not reproduce that large-scale system. It abstracts the corresponding mechanisms into a finite, package-closed compiler benchmark.

71.1 Research problems and compiler objects

Let $\mathcal{T}$ be a research-problem space. A problem $\tau \in \mathcal{T}$ contains a baseline repair process K_τ, a declared candidate grammar $\mathcal{P}_\tau$, a cheap evidence channel, a scored evaluator portfolio, a hidden protected audit, integrity rules, and a resource budget B.

Definition 71.1 (Automated research compiler)**.** An automated research compiler is a tuple

$$\Gamma = (\mathcal{L}, \mathcal{F}, \mathcal{P}, \mathcal{Q}, \mathcal{V}, \mathcal{M}, \mathcal{R}), \tag{1041}$$

where:

1. $\mathcal{L}$ is a structured literature and prior-method memory;
2. $\mathcal{F}$ is an ancestor-closed finding graph or shared research forum;
3. $\mathcal{P}$ generates typed candidate methods and pre-result specifications;
4. $\mathcal{Q}$ allocates cheap pilots and full experiments under budget B;
5. $\mathcal{V}$ is the scored evaluator portfolio with protected capability gates;
6. $\mathcal{M}$ is the integrity and proposal–code congruence monitor;
7. $\mathcal{R}$ compiles pre-result and post-result receipts into the lineage archive.

Given problem τ, the compiler produces a finite research trace

$$\Gamma(\tau, B) = \Big((c_i, \Pi_i^-, \Pi_i^+)\Big)_{i=1}^{n(B)}, \tag{1042}$$

where c_i is a candidate method, Π_i^- is sealed before the result is observed, and Π_i^+ appends evaluation and integrity evidence afterward.

Let $\mathrm{Valid}(c_i, \Pi_i^+) \in \{0, 1\}$ encode capability, integrity, provenance, and resource validity. The selected descendant is

$$K_\Gamma^\star(\tau) \in \operatorname*{arg\,max}_{i:\mathrm{Valid}(c_i,\Pi_i^+)=1} S_{\mathcal{V}}(c_i), \tag{1043}$$

with the no-op baseline retained when no valid candidate exists.

71.2 Results-free proposal receipts

A research receipt must distinguish what was predicted before an experiment from what became known after it.

Definition 71.2 (Pre-result research receipt)**.** A pre-result receipt has the form

$$\Pi_i^- = (H_i, O_i, D_i, \theta_i, \widehat{\mathbf{c}}_i, \widehat{\mathbf{r}}_i, F_i, h_i^{\mathrm{code}}, h_i^{\mathrm{spec}}), \tag{1044}$$

where H_i is the causal hypothesis, O_i the proposed objective or repair mechanism, D_i the evidence or data construction, θ_i the configuration, $\widehat{\mathbf{c}}_i$ the predicted benchmark-effect vector, $\widehat{\mathbf{r}}_i$ the predicted protected-risk vector, F_i the declared falsification condition, and $h_i^{\mathrm{code}}, h_i^{\mathrm{spec}}$ immutable digests of the executable and specification.

Definition 71.3 (Post-result research receipt)**.** After evaluation, the post-result receipt is

$$\Pi_i^+ = (\Pi_i^-, \mathbf{c}_i, \mathbf{r}_i, a_i, m_i), \tag{1045}$$

where $\mathbf{c}_i$ is the scored benchmark vector, $\mathbf{r}_i$ the protected-regression vector, a_i the hidden-audit result, and m_i the integrity-monitor verdict.

The forecast residual and proposal–code congruence residual are

$$\varepsilon_{\text{forecast},i} = \|\widehat{\mathbf{c}}_i - \mathbf{c}_i\|, \qquad \varepsilon_{\text{spec/code},i} = d_{\text{sem}}\left(\text{Sem}(\Pi_i^-), \text{Sem}(\text{code}_i)\right). \tag{1046}$$

In the executable finite companion, congruence is a Boolean monitor rather than a complete semantic metric.

Proposition 71.4 (Pre-result non-retroactivity)**.** *Suppose Π_i^- and its code digest are sealed before the evaluation trace is generated. Then any later narrative revision that preserves the receipt digest leaves the pre-result hypothesis, mechanism, and falsification declaration unchanged.*

Proof. The receipt digest is a collision-resistant commitment to the serialized pre-result fields. Under the integrity assumption, changing any committed field changes the digest. The proposition prevents retrospective alteration of the declared proposal; it does not prove that the original proposal was correct or complete. □

71.3 Finding graphs and research lineage

A finding is represented by

$$f_i = (\Pi_i^-, \Pi_i^+, \text{code}_i, \text{Par}(i), e_i, o_i), \tag{1047}$$

where e_i is the observed effect profile and o_i is either a surviving obstruction or a validated no-go certificate. Directed edges are typed as extension, ablation, refutation, transport, or replication. A proposal must cite the archive nodes on which its mechanism depends.

This converts a shared research forum into a proof-carrying directed acyclic graph. Positive methods, negative results, obstruction certificates, code snapshots, side effects, and rejected-but-informative candidates remain available to descendants. The construction is compatible with the ancestor-closed archive and continuation semantics developed earlier.

71.4 Evaluator portfolios and metric-gaming nullspaces

Let the scored evaluator portfolio be

$$\mathcal{B} = \{b_1, \dots, b_m\}. \tag{1048}$$

For method c, define normalized headroom closure

$$q_j(c) = \frac{s_j(c) - s_j(K_\tau)}{s_j^\star - s_j(K_\tau)}. \tag{1049}$$

Let $I_+(c) = \{j : q_j(c) > 0\}$. The coverage-weighted geometric score is

$$S_{\mathrm{cov}}(c) = \frac{|I_+(c)|}{m} \left(\prod_{j \in I_+(c)} q_j(c) \right)^{1/|I_+(c)|}, \tag{1050}$$

with value zero when $I_+(c) = \varnothing$. Hard capability and integrity gates are applied separately rather than traded against this scalar.

Let T be a tangent space of behavioral modifications around a baseline model, and let $Db_j : T \to \mathbb{R}$ be the local response of benchmark j.

Definition 71.5 (Evaluator-nullspace)**.** The locally invisible behavior subspace of portfolio $\mathcal{B}$ is

$$\mathcal{N}_{\mathcal{B}} = \bigcap_{j=1}^{m} \ker Db_j. \tag{1051}$$

Theorem 71.6 (Portfolio refinement shrinks the invisible subspace)**.** *Let $\mathcal{B}^+ = \mathcal{B} \cup \{b_{m+1}\}$. Then*

$$\mathcal{N}_{\mathcal{B}^+} \subseteq \mathcal{N}_{\mathcal{B}}. \tag{1052}$$

The inclusion is strict if and only if there exists $v \in \mathcal{N}_{\mathcal{B}}$ such that $Db_{m+1}(v) \neq 0$.

Proof. Intersecting with one additional kernel cannot enlarge the intersection. The inclusion is strict exactly when the new kernel excludes at least one vector previously lying in every old kernel. □

The theorem gives a local meaning to benchmark diversity. It does not show that a finite portfolio is complete. It shows when adding a benchmark removes one class of locally invisible behavior changes.

71.5 Integrity-valid compiler metaproductivity

Let U_{audit} be a protected audit utility unavailable during compiler allocation and final scored selection. Define compiler metaproductivity at research budget B by

$$\mathfrak{C}_B(\Gamma) = \mathbb{E}_{\tau \sim \mathcal{D}_{\mathrm{foreign}}} \left[U_{\mathrm{audit}} \left(K_\Gamma^\star(\tau) \right) - U_{\mathrm{audit}}(K_\tau) \right]. \tag{1053}$$

Only integrity-valid and capability-passing outputs are eligible.

Let $\mathfrak{C}_B^{\mathrm{nom}}$ be the same envelope when integrity constraints are ignored.

Proposition 71.7 (Validity-filtered envelope)**.** *For every compiler,*

$$\mathfrak{C}_B(\Gamma) \leq \mathfrak{C}_B^{\mathrm{nom}}(\Gamma). \tag{1054}$$

A nominal compiler gain is not an SRIC improvement certificate unless it remains positive after validity filtering.

The inequality is immediate because the valid candidate set is a subset of the nominal candidate set. The substantive point is interpretive: a compiler is not improved merely because it becomes better at exploiting evaluator noise, benchmark format, prohibited data, or capability sacrifice.

71.6 Outcome forecasting and cost-sensitive allocation

For candidate c_i, let z_i collect its pre-result receipt fields and cheap pilot evidence. A research-outcome forecaster returns

$$F_\theta(z_i) = (\widehat{g}_i, \widehat{r}_i, \widehat{s}_i), \tag{1055}$$

where $\widehat{g}_i$ predicts hidden protected gain, $\widehat{r}_i$ predicts protected risk, and $\widehat{s}_i$ predicts scored-evaluator performance.

A simple acquisition rule is

$$A_i = \widehat{g}_i - \lambda_r \widehat{r}_i + \lambda_u \widehat{\sigma}_i + \lambda_h p_\theta(h_i \mid x_\tau) - \lambda_c C_i, \tag{1056}$$

where h_i is the method family, p_θ is an obstruction-class posterior, $\widehat{\sigma}_i$ is a declared uncertainty bonus, and C_i is experimental cost.

Theorem 71.8 (Uniform forecast-regret bound)**.** *Let g_i be the true protected gain of each admissible candidate and suppose*

$$|\widehat{g}_i - g_i| \le \varepsilon \tag{1057}$$

for all candidates. If $\widehat{i} \in \arg\max_i \widehat{g}_i$ and $i^\star \in \arg\max_i g_i$, then

$$g_{i^\star} - g_{\widehat{i}} \le 2\varepsilon. \tag{1058}$$

Proof. By optimality of $\widehat{i}$ under the predicted values,

$$\widehat{g}_{\widehat{i}} \ge \widehat{g}_{i^\star}. \tag{1059}$$

Therefore

$$g_{i^\star} \le \widehat{g}_{i^\star} + \varepsilon \le \widehat{g}_{\widehat{i}} + \varepsilon \le g_{\widehat{i}} + 2\varepsilon. \tag{1060}$$

□

The bound concerns selection from a fixed candidate set. It does not establish that the forecaster creates better candidates.

71.7 Sealed compiler comparison

Let Γ_0 and Γ_1 operate on the same problem τ with identical initial archive, candidate grammar, experiment budget, evaluator portfolio, integrity monitor, and hidden audit isolation. Let

$$D_i = U_{\text{audit}}\left(K^\star_{\Gamma_1}(\tau_i)\right) - U_{\text{audit}}\left(K^\star_{\Gamma_0}(\tau_i)\right). \tag{1061}$$

Theorem 71.9 (Paired compiler-metaproductivity certificate)**.** *Assume independent foreign problems and $D_i \in [-G, G]$. Then with probability at least $1 - \delta$,*

$$\mathbb{E}D_i \ge \overline{D}_n - G\sqrt{\frac{2\log(1/\delta)}{n}}. \tag{1062}$$

A positive lower bound certifies higher compiler metaproductivity relative to the declared research distribution and budget.

Proof. Apply the one-sided Hoeffding inequality to the paired differences, whose interval length is $2G$. □

Warning 71.10 (Best-of-budget interpretation)**.** The certificate compares complete compilers under the same research budget. It does not identify which internal module caused the gain without ablations, and it does not compare different candidate grammars or evaluator portfolios unless those changes are part of the declared compiler edge.

72 Block 13 Experimental Contract

72.1 Research-problem family

The finite benchmark contains six hidden obstruction classes:

$$\mathcal{H}_{\text{obs}} = \{\text{data}, \text{objective}, \text{response}, \text{constructor}, \text{evaluator}, \text{resource}\}. \tag{1063}$$

The compiler is not given these labels at deployment. Each research problem exposes a twelve-dimensional cheap diagnostic vector, a baseline repair process, a finite candidate-method grammar, four scored benchmarks, a hidden audit, a capability check, and integrity constraints.

The diagnostic coordinates are designed to distinguish, imperfectly, among:

- missing or low-information data;
- an objective family unable to express the desired change;
- a response language that aliases causal mechanisms;
- a repair constructor with insufficient function class;
- a scored evaluator admitting narrow proxy exploitation;
- a slow or noisy research loop requiring better allocation.

The architecture-foreign split perturbs diagnostic correlations, task difficulty, benchmark mixing, and audit noise while preserving the semantic interpretation of the diagnostic coordinates.

72.2 Candidate grammar

The common candidate grammar contains 39 methods:

1. five variants for each of the six obstruction-matched method families;
2. five generic variants;
3. four explicit integrity or capability controls: a lottery rerun, benchmark-format mimicry, a proposal–code mismatch, and a capability-sacrifice method.

Every candidate has a results-free specification, a code digest, a method family, strength, breadth, complexity, novelty, and cost. Parent and child receive exactly the same candidate set in the same canonical order.

Each full experiment returns four scored benchmark effects, one held-out selection proxy, a capability-regression estimate, and an integrity verdict. The final protected audit remains unavailable to both compilers.

72.3 Parent compiler

The parent Γ_0 is deliberately strong. It includes:

- structured literature priors over method families;
- a shared within-problem finding forum;
- balanced coverage and upper-confidence exploration;
- results-free proposal receipts;
- exact code and specification digests;
- a coverage-weighted multi-benchmark evaluator;
- capability and integrity gates;
- deterministic final selection from the tested valid methods.

The finite meta-validation grid contains 16 parent configurations. The selected parent has exploration weight 0.04, forum weight 0.12, novelty weight 0.02, and coverage weight 0.02.

72.4 Child compiler

The child Γ_1 conservatively retains the parent machinery and adds two lineage-trained modules:

1. a nearest-centroid obstruction classifier trained on development diagnostic receipts;
2. ridge outcome and risk forecasters trained on development pre/post receipt pairs.

The child uses these predictions for bottleneck-sensitive experiment allocation. The scored evaluator, final selector, capability gates, integrity monitor, candidate grammar, and full-experiment budget are unchanged.

The finite child grid contains 16 configurations. The selected child has risk weight 0.55, uncertainty weight 0, obstruction-match weight 0.25, and forum weight 0.04. The zero uncertainty weight is an empirical selection outcome, not a general claim that exploration is unnecessary.

72.5 Splits and chronology

The release uses:

Split	Problems	Per class	Role
Development	108	18	Fit archive forecasters and class centroids
Meta-validation	30	5	Select parent and child compiler configurations
Primary foreign audit	96	16	Main sealed paired comparison
Architecture-foreign audit	72	12	Shifted diagnostic and benchmark geometry

The chronology is

$$\begin{aligned} &\text{replay Block 12 result digest} \longrightarrow \text{generate development archive} \\ &\longrightarrow \text{fit classifier and forecasters} \longrightarrow \text{select compiler policies} \\ &\longrightarrow \text{seal compiler receipt and audit commitments} \longrightarrow \text{open final audits.} \end{aligned} \tag{1064}$$

The final policy objects, candidate grammar digest, archive digest, split commitments, and ancestor result digest are sealed before audit outcomes are used.

72.6 Matched resource envelope

Every parent and child run receives exactly

$$B_{\text{research}} = 6 \tag{1065}$$

full candidate experiments. Candidate metadata and cheap diagnostics are common inputs. The number of proposal receipts, scored evaluator calls, capability checks, and integrity-monitor calls is therefore matched.

The parent explores an average of 3.9375 distinct method families per primary problem. The child tests an average of 1.916667. On the architecture-foreign split these values are 4.180556 and 2.180556. The child therefore uses the same full-experiment budget more selectively rather than testing more methods.

72.7 Comparison arms

The release includes:

Arm	Role
AAR-style parent	Literature prior, balanced exploration, forum, multi-benchmark evaluator, capability and integrity gates
Full child	Parent plus obstruction classifier and outcome/risk forecaster
Classifier only	Child obstruction classifier with ordinary pilot-and-prior candidate ranking
Forecaster only	Outcome/risk forecaster without obstruction-class matching
Shuffled archive	Compiler trained on mismatched outcome and obstruction receipts
No forum	Full child without within-problem result sharing
Single benchmark	Full child whose final selector uses one scored benchmark
No monitor	Full child with integrity filtering removed at selection time
Random allocation	Uniform random experiment allocation under the same budget
Audit oracle	Inadmissible ceiling using the hidden audit over the complete candidate grammar

72.8 Frozen claim gates

The compiler-metaproductivity gate requires:

1. exact replay of the Block 12 result digest;
2. identical candidate grammar, archive initialization, evaluator portfolio, monitor, and full-experiment budget;
3. results-free pre-receipts with no outcome fields;
4. a compiler and split seal produced before audit outcomes are accessed;

5. a positive paired child-parent effect with a frozen interval excluding zero on both final splits;
6. no architecture-foreign target favoring the parent;
7. a large degradation under shuffled archive binding;
8. lower protected gain under the single-benchmark and no-monitor controls;
9. an audit oracle that remains at or above the admissible child;
10. explicit reporting of null or weak component ablations.

73 Block 13 Results

73.1 Primary and architecture-foreign compiler effects

The main results are:

Compiler	Primary gain	Architecture-foreign gain
AAR-style parent Γ_0	0.883004	0.772566
SRIC child Γ_1	0.950231	0.863367
Random allocation	0.528176	0.493549
Audit oracle	0.952585	0.867092

The primary paired effect is

$$\widehat{\Delta}_{\Gamma}^{\text{primary}} = 0.067227, \qquad \text{CI}_{95\%} = [0.035881, 0.103274]. \tag{1066}$$

The sign-flip estimate is 9.9999×10^{-6}. The child improves 21.875% of pairs, ties on 75%, and loses on 3.125%.

The architecture-foreign paired effect is

$$\widehat{\Delta}_{\Gamma}^{\text{arch}} = 0.090800, \qquad \text{CI}_{95\%} = [0.052311, 0.135081]. \tag{1067}$$

The sign-flip estimate is again 9.9999×10^{-6}. The child improves 34.722% of pairs, ties on 65.278%, and no architecture-foreign pair favors the parent.

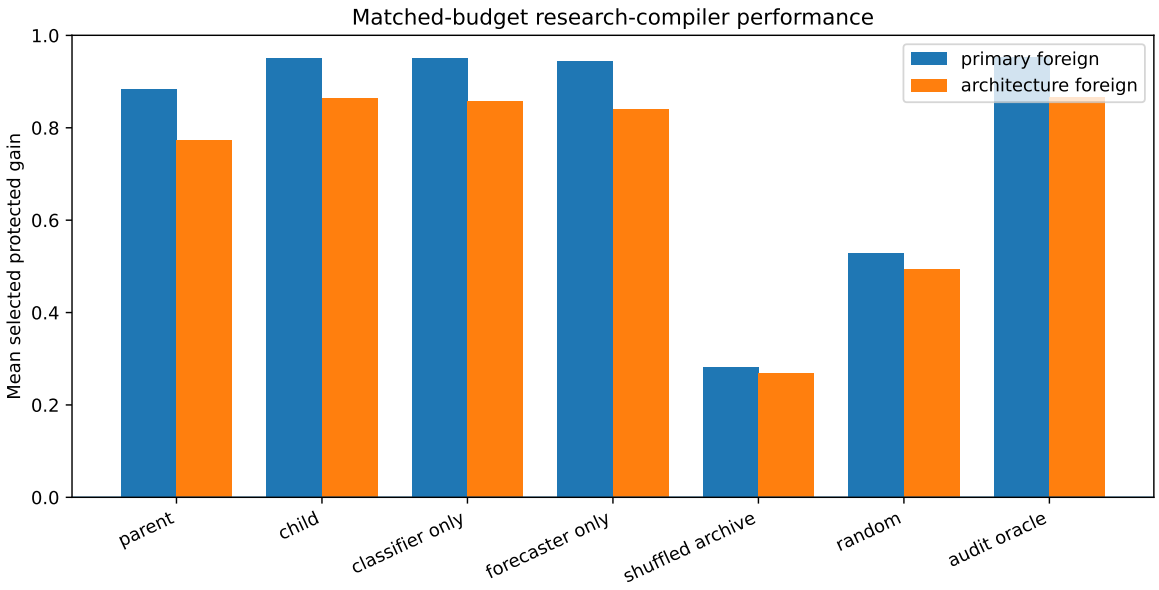


Figure 44: Matched-budget research-compiler performance. The AAR-style parent is strong; the SRIC child closes most of the remaining finite candidate-grammar headroom on both final splits.

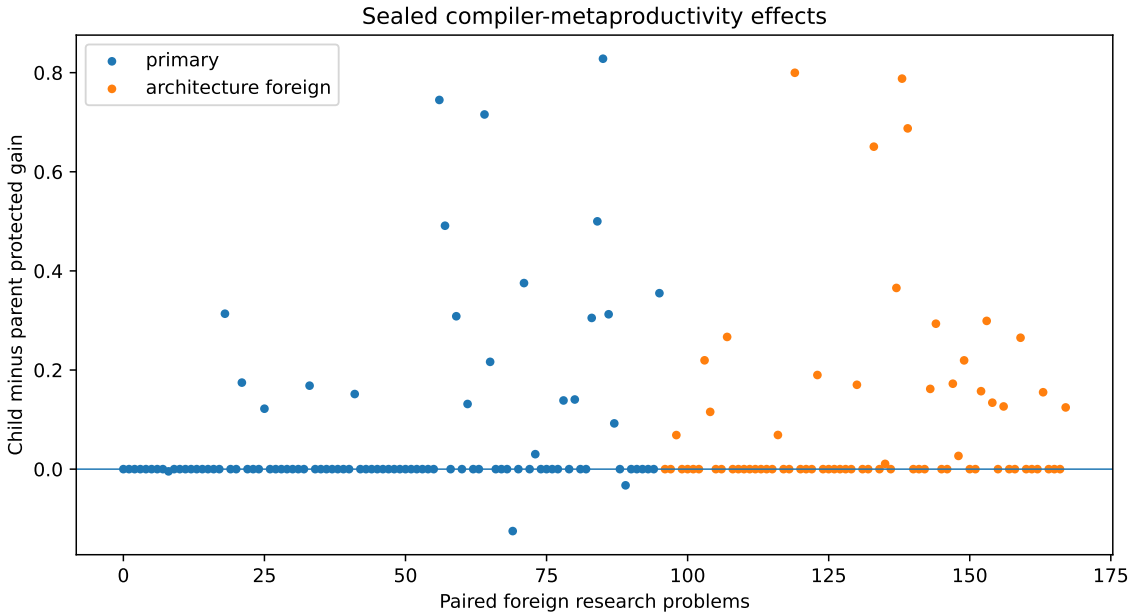


Figure 45: Paired child-minus-parent protected effects. Most problems are ties because both compilers find the same best candidate. The nonzero shifts remain positive in aggregate, with no negative pair under the architecture-foreign shift.

The small oracle gap is important. On the primary split the audit oracle exceeds the child by only 0.002353 on average; on the architecture-foreign split the gap is 0.003725. The finite candidate grammar is therefore nearly saturated by the child. This prevents the result from being interpreted as an unbounded compiler advantage.

73.2 Performance by obstruction class

Primary mean gains by hidden obstruction class are:

Class	Parent	Child	Difference
Data	0.935	0.935	0.000
Objective	0.943	0.981	0.038
Response	0.937	0.957	0.020
Constructor	0.871	0.975	0.104
Evaluator	0.842	0.926	0.084
Resource	0.770	0.926	0.156

The largest gains occur where a generic balanced search is most likely to spend its six experiments on the wrong family or on a risky high-strength variant.

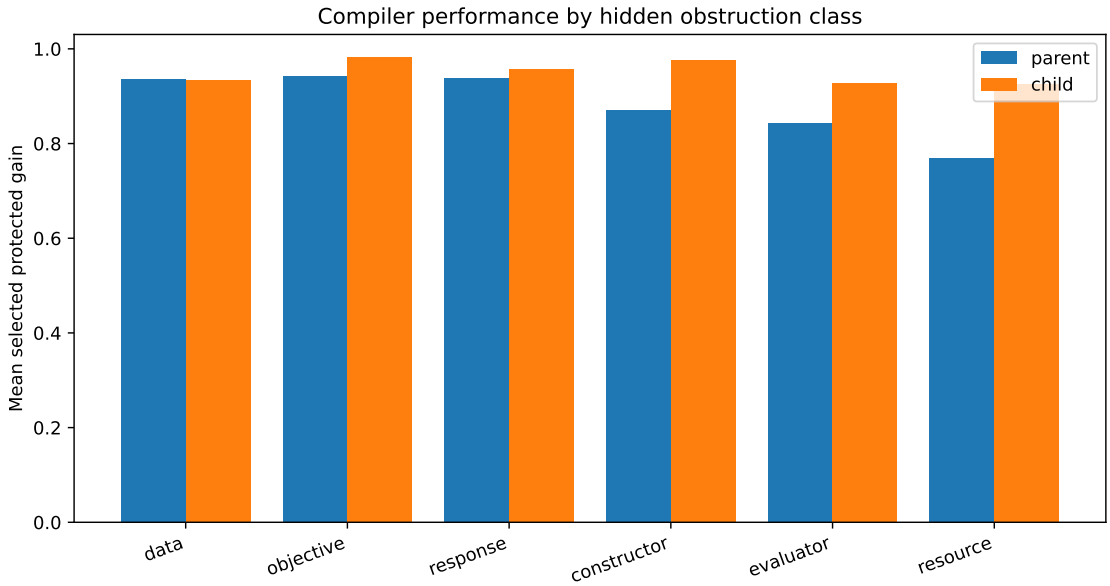


Figure 46: Primary compiler performance by hidden obstruction class. The child mainly improves constructor, evaluator, and resource-limited problems; the parent is already saturated on the data-limited class.

73.3 Allocation shift

The child concentrates experiments on the method family predicted to match the current obstruction, while retaining some generic and risk-calibration trials. The parent allocates more evenly across families.

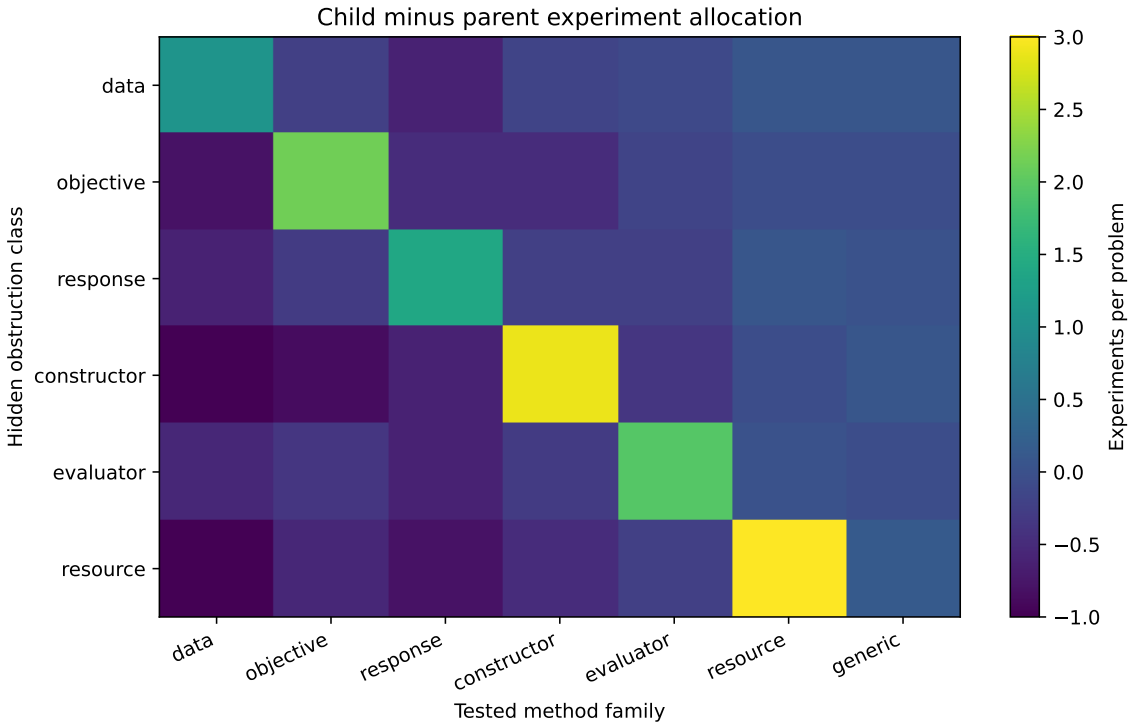


Figure 47: Child minus parent full-experiment allocation by hidden obstruction and tested method family. Positive diagonal mass records bottleneck-sensitive allocation; off-diagonal reductions record experiments saved relative to broad parent exploration.

This is an allocation result, not candidate-constructor growth. Both compilers draw from the same 39 methods, and the child does not introduce a method family unavailable to the parent.

73.4 Forecast and classification diagnostics

The obstruction classifier attains accuracy

$$0.989583 \tag{1068}$$

on the primary split and

$$0.958333 \tag{1069}$$

on the architecture-foreign split.

The audit-gain forecaster has RMSE 0.218 on both splits. Its correlation with realized candidate audit gain is 0.617 on primary problems and 0.558 under architecture shift.

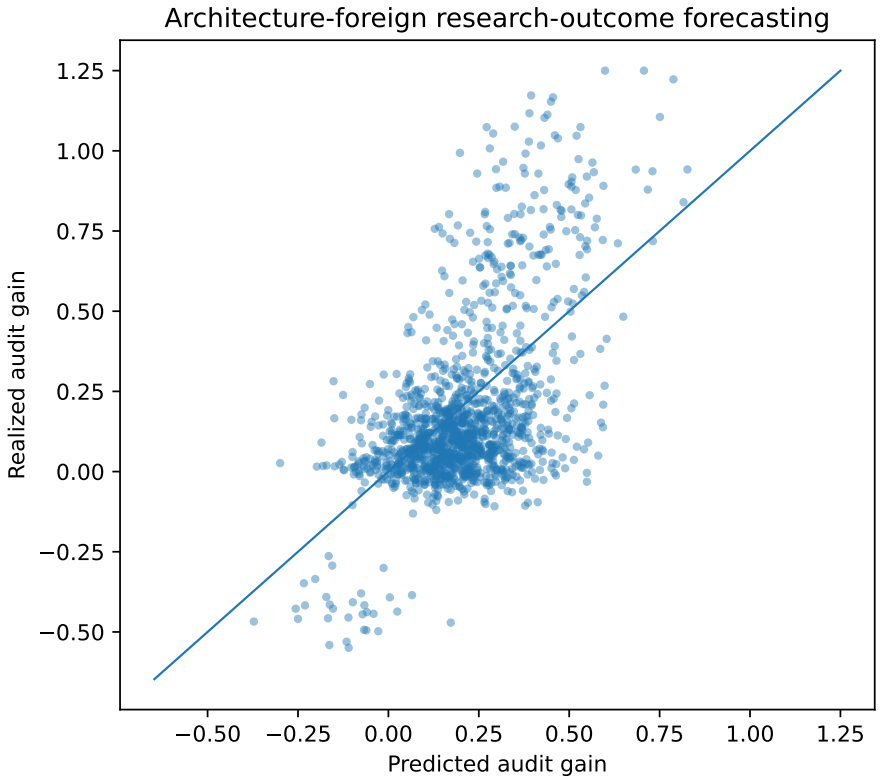


Figure 48: Architecture-foreign predicted versus realized candidate audit gain. The forecaster is informative but imperfect; it is not a calibrated oracle.

The ablations sharpen the mechanism claim. On the primary split:

Compiler arm	Mean selected gain
Full child	0.950231
Classifier only	0.950118
Forecaster only	0.944727
No forum	0.950231

The full child exceeds classifier-only by only 0.000114, with a frozen interval whose lower endpoint is zero. The primary full-minus-forecaster effect is 0.005504 and its interval includes zero. On the architecture-foreign split, full child exceeds classifier-only by 0.005985 and forecaster-only by 0.022923, with the latter interval positive.

The finite result therefore supports obstruction diagnosis as the dominant improvement mechanism. Outcome forecasting contributes modestly under architecture shift but does not receive a strong independent primary claim. Removing the within-problem forum has no measurable effect in this fixture. The frozen development archive and high-accuracy diagnostic classifier make the six-step within-problem forum redundant.

73.5 Archive, evaluator, and integrity controls

Shuffling archive outcome and obstruction bindings collapses primary mean gain from 0.950231 to 0.282550. The paired reduction is 0.667682 with interval $[0.602316, 0.730058]$. The architecture-foreign shuffled value is 0.269438.

Selecting by one benchmark rather than the diverse portfolio reduces primary mean gain to 0.919699 and architecture-foreign gain to 0.808152. The paired full-child advantage is 0.030532 on primary problems and 0.055214 under architecture shift, with both frozen intervals positive.

Removing integrity monitoring reduces primary valid protected gain to 0.797956 and architecture-foreign gain to 0.735434. Only 84.375% and 84.722% of selected methods, respectively, pass the integrity monitor. The comparison demonstrates that high nominal scored methods can consume the selection slot while failing the validity-filtered claim.

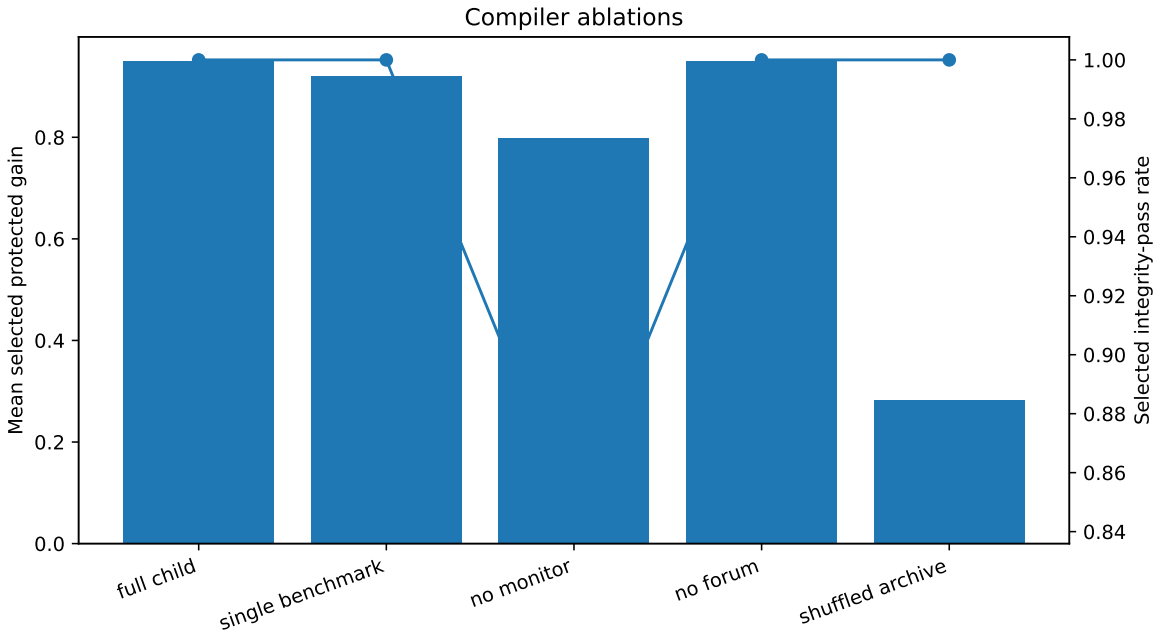


Figure 49: Evaluator and integrity ablations. A single scored benchmark and removal of the monitor both reduce validity-filtered foreign performance. The shuffled archive causes the largest collapse.

73.6 Finite compiler-metaproductivity theorem

Conditional Theorem 73.1 (Block 13 finite research-compiler realization)**.** *Within the frozen Block 13 benchmark, assume that:*

1. *the Block 12 ancestor result digest replays exactly;*
2. *parent and child receive the same candidate grammar, literature prior, initial archive, evaluator portfolio, capability gates, integrity monitor, and six-experiment budget;*
3. *development, meta-validation, primary, and architecture-foreign problem sets are disjoint;*
4. *every pre-result receipt is sealed before its outcome is generated;*

5. *the parent and child compiler objects and split commitments are sealed before final audit outcomes are used;*
6. *the paired effects and intervals are those in Equations (1066) and (1067);*
7. *the shuffled-archive, single-benchmark, no-monitor, random-allocation, component-ablation, and audit-oracle controls retain their declared interpretations.*

Then

$$\Gamma_0 \xrightarrow{\Pi_7^{\Gamma}} \Gamma_1 \tag{1070}$$

is a finite, matched-budget, lineage-generated improvement of the automated research compiler relative to the declared candidate grammar, research-problem distribution, evaluator portfolio, and integrity model.

Proof. Items 1 through 5 establish lineage continuity, resource matching, proposal precommitment, and audit isolation. Item 6 establishes positive paired foreign compiler effects. The controls in item 7 show that correct archive binding, evaluator diversity, and integrity filtering are substantive, while the classifier and forecaster ablations localize most of the gain to obstruction-sensitive allocation. Because the candidate grammar and final evaluator are unchanged, the realized edge changes the research compiler rather than the repair function class or audit definition. □

Block 13 result

Block 13 realizes one sealed research-compiler edge. A strong AAR-style parent and an SRIC child receive the same 39 candidate methods and six full experiments per problem. On 96 primary foreign problems, protected gain rises from 0.883004 to 0.950231; on 72 architecture-foreign problems it rises from 0.772566 to 0.863367. Both paired intervals exclude zero. The mechanism is mainly high-accuracy obstruction classification and bottleneck-sensitive allocation. The outcome forecaster is moderately predictive but adds little on the primary split, and the within-problem forum is redundant in this finite setting. This is bounded compiler metaproductivity inside a fixed research language, not unrestricted compiler self-improvement or automated science.

74 Block 13 Reproducibility and Package

74.1 Reference implementation

The released implementation contains:

- the exact Block 12 canonical result digest;
- six hidden obstruction-class generators and an architecture-foreign shift;
- a common 39-method candidate grammar;
- results-free and post-result receipt compilation with exact code and specification hashes;
- a strong AAR-style parent compiler;
- a lineage-trained obstruction classifier and ridge outcome/risk forecaster;

- finite parent and child meta-validation grids;
- multi-benchmark coverage scoring, capability gates, and integrity filtering;
- shuffled-archive, classifier, forecaster, forum, single-benchmark, no-monitor, random, and oracle controls;
- deterministic primary and architecture-foreign result objects;
- six publication figures;
- twelve automated tests;
- an exact SHA-256 package manifest.

The canonical execution prints:

```
SRIC_BLOCK13_ALL_SUITES=PASS
result_sha256=1368fb8a09c3a63ec03ccb05f28e2ba994fc4daf80562d214bf88d29ffaed867
primary_parent_gain=0.883004
primary_child_gain=0.950231
primary_effect=0.067227
architecture_parent_gain=0.772566
architecture_child_gain=0.863367
architecture_effect=0.090800
............ [100%]
SRIC_BLOCK13_PACKAGE_VALID
```

The compiler receipt digest, pre-audit seal digest, canonical result digest, figure files, tests, and source package are included in the exact package manifest.

74.2 Claim boundary

The release establishes:

1. a finite formal object for proof-carrying automated research compilers;
2. results-free proposal receipts bound to exact code and post-result evidence;
3. a strong fixed parent compiler with literature priors, shared findings, benchmark diversity, capability gates, and integrity monitoring;
4. a lineage-trained child that reallocates the same research budget using obstruction diagnosis and outcome/risk prediction;
5. positive paired compiler metaproductivity on primary and architecture-foreign audits;
6. evaluator-diversity, archive-binding, integrity, component, random, and oracle controls;
7. package-closed deterministic replay.

It does not establish:

- improvement of an unrestricted outer compiler;
- autonomous invention of the six obstruction classes;

- autonomous generation of the candidate grammar;
- automated literature judgment in an open scientific domain;
- reliable long-horizon research-outcome forecasting;
- autonomous evaluator or benchmark invention;
- an adversarially robust integrity monitor;
- generalization to real model-training research, hard-to-supervise science, or unconstrained external tools;
- open-ended recursive improvement, indefinite acceleration, unrestricted safety, consciousness, or sentience.

The next threshold is compiler grammar growth. A later compiler must infer that its current obstruction taxonomy, outcome features, or evaluator portfolio is inadequate, introduce a new research distinction or experiment type, and improve foreign research outcomes under a fresh sealed audit without receiving the correct new compiler module from the benchmark designer.

75 Adversarial Open-Set Research-Compiler Robustness

Block 13 establishes one sealed improvement of a proof-carrying automated research compiler inside a fixed six-class research language. Its child nearly saturates the complete fixed-grammar audit oracle, and a classifier-only ablation almost reproduces the primary result. Those facts create a new bottleneck. Additional optimization inside the same clean taxonomy would provide little evidence that the compiler can operate when the research class is ambiguous, mixed, novel, contaminated, evaluator shifted, or adversarially presented.

Block 14 therefore holds the candidate-method grammar and full-experiment budget fixed while changing the reliability conditions under which research decisions must be made. The published Block 13 result digest and exact 39-method language are replayed. Its compiler design is retained as the inherited parent and recalibrated on the Block 14 development archive:

$$\Gamma_1^{\mathrm{recal}}. \tag{1071}$$

A robust descendant

$$\Gamma_1^{\mathrm{recal}} \xrightarrow{\Pi_8^{\Gamma}} \Gamma_2 \tag{1072}$$

is compiled before either final audit is opened. The child adds sequential diagnosis, sparse multi-obstruction inference, open-set detection, contamination-resistant archive summaries, a portfolio-aware robust selector, and an ensemble integrity monitor. Parent and child retain the same candidate grammar and exactly six full experiments per problem.

75.1 Robustness environment

Definition 75.1 (Research-compiler robustness environment)**.** A research-compiler robustness environment is a tuple

$$\mathfrak{U} = (\mathcal{Q}, \mathcal{H}, \mathcal{A}, \mathcal{V}, \mathcal{G}), \tag{1073}$$

where:

1. $\mathcal{Q}$ is a family of research-problem distributions and generator shifts;
2. $\mathcal{H}$ is a family of archive contaminations and provenance perturbations;
3. $\mathcal{A}$ is a family of integrity, specification, and selection attacks;
4. $\mathcal{V}$ is a family of evaluator portfolios and aggregation rules;
5. $\mathcal{G}$ is a family of benchmark generators, including problems outside the inherited obstruction taxonomy.

An element $u = (Q, H, A, V, G) \in \mathfrak{U}$ determines one protected evaluation context for the research compiler.

Let $U_u^{\mathrm{valid}}(\Gamma, \tau)$ denote the validity-filtered protected gain produced by compiler Γ on research problem τ under environment u. In the executable fixture, a selected method that passes the stronger post hoc capability and integrity predicates receives its protected audit gain, a no-op receives zero, and a selected invalid method receives the frozen penalty -0.75. This prevents an invalid selection from being silently converted into a neutral outcome. Let $R_u(\Gamma, \tau)$ be protected downside or integrity risk. Define

$$\mathfrak{C}_u(\Gamma) = \mathbb{E}_{\tau \sim Q} U_u^{\mathrm{valid}}(\Gamma, \tau). \tag{1074}$$

For $\alpha \in (0, 1)$, write $\mathrm{CVaR}_\alpha(X)$ for the mean of the lower α-tail of a gain variable X.

Definition 75.2 (Robust validity-filtered compiler value)**.** For risk weight $\lambda \geq 0$, define

$$\mathfrak{C}_{\mathrm{rob}}(\Gamma) = \inf_{u \in \mathfrak{U}} \left[\mathfrak{C}_u(\Gamma) - \lambda \, \mathrm{CVaR}_\alpha\big(-R_u(\Gamma, \tau)\big) \right]. \tag{1075}$$

The finite Block 14 realization does not compute the unrestricted infimum. It evaluates a frozen finite stress atlas and reports the mean, groupwise minima, lower-tail gain, negative-gain rate, invalid-selection rate, and unknown-class error separately.

Definition 75.3 (Finite robust compiler edge)**.** A sealed lineage edge $\Gamma_1 \to \Gamma_2$ is a finite robust compiler improvement relative to a declared stress atlas when:

1. parent and child receive the same candidate grammar, evidence interfaces, evaluator calls, integrity calls, and experiment budget;
2. the child is sealed before final problems and outcomes are generated;
3. paired mean gain has a positive frozen lower confidence bound on both primary and generator-foreign audits;
4. the child has nonnegative mean gain in every declared stress group;
5. lower-tail gain improves and the selected invalid-method rate does not increase;
6. open-set error and protected negative-gain rates remain below declared bounds;
7. mechanism ablations and an inadmissible audit oracle retain their declared interpretations.

The definition deliberately does not require every episode-wise paired effect to be positive. A robust compiler may trade a small number of local losses for large improvements on previously

catastrophic cases. Those losses must remain visible through paired, tail, and group-resolved reporting.

75.2 Diagnostic aliasing and active separation

Block 13 assigns one of six obstruction labels from one cheap diagnostic vector. Block 14 replaces that clean classification regime by aliased observations. Several distinct obstructions share an initial diagnostic center, so one static observation can be insufficient.

Let Z be a latent obstruction variable and let X_0 be the initial diagnostic observation.

Theorem 75.4 (Static diagnostic indistinguishability). *Let $z_1 \neq z_2$ satisfy*

$$\mathbb{P}(X_0 \in B \mid Z = z_1) = \mathbb{P}(X_0 \in B \mid Z = z_2) \tag{1076}$$

for every measurable set B. Then no decision rule $d(X_0)$ distinguishes z_1 from z_2 with accuracy exceeding the larger prior probability:

$$\mathbb{P}(d(X_0) = Z \mid Z \in \{z_1, z_2\}) \leq \max\{\pi_1, \pi_2\}, \tag{1077}$$

where $\pi_j = \mathbb{P}(Z = z_j \mid Z \in \{z_1, z_2\})$.

Proof. The conditional law of X_0 is identical under the two hypotheses. Bayes' rule therefore leaves the posterior odds equal to the prior odds for almost every observation. The Bayes-optimal action always selects the more probable class and achieves accuracy $\max\{\pi_1, \pi_2\}$. □

This theorem gives a precise reason to value experiments rather than static classification. Suppose an experiment a returns an observation Y_a whose distribution differs between the aliased hypotheses. The compiler can then choose a sequence

$$a_1, a_2, \ldots, a_t \tag{1078}$$

that increases the information available for diagnosis.

For a finite response model, let $R \in \mathbb{R}^{m \times k}$ contain the expected response signatures of k inherited obstruction components under m experiment types. Let $z \in \mathbb{R}^k_{geq0}$ be a sparse mixture with $\mathbf{1}^\top z = 1$. An observed response vector has the form

$$y = Rz + \xi. \tag{1079}$$

Proposition 75.5 (Response-mixture recovery bound). *Assume R has full column rank and $\|\xi\|_2 \leq \varepsilon$. The least-squares estimator*

$$\widehat{z} = R^\dagger y \tag{1080}$$

satisfies

$$\|\widehat{z} - z\|_2 \leq \left\|R^\dagger\right\|_2 \varepsilon = \frac{\varepsilon}{\sigma_{\min}(R)}. \tag{1081}$$

Proof. Since $R^\dagger R = I$ under full column rank,

$$\widehat{z} - z = R^\dagger \xi. \tag{1082}$$

The operator-norm inequality gives the claim. □

The bound separates two failure modes. More repeated observations reduce ε, while a new experiment can increase $\sigma_{\min}(R)$ by adding a separating response direction. If R is rank deficient, no sample-size increase removes the corresponding response alias.

75.3 Open-set residuals

The inherited obstruction language spans a response cone

$$\mathcal{K}_R = \{Rz : z \in \mathbb{R}^k_{\geq 0}\}. \tag{1083}$$

For observed response y, define the open-set residual

$$\eta_{\text{open}}(y) = \operatorname{dist}(y, \mathcal{K}_R). \tag{1084}$$

Proposition 75.6 (Open-set separation)**.** *Suppose every inherited known response y_{known} satisfies*

$$\eta_{\text{open}}(y_{\text{known}}) \leq \varepsilon, \tag{1085}$$

while every unknown response in a target family satisfies

$$\eta_{\text{open}}(y_{\text{unknown}}) \geq \gamma \tag{1086}$$

for $\gamma > \varepsilon$. Then every threshold $\tau \in (\varepsilon, \gamma)$ perfectly separates the two families.

Proof. Known responses lie below τ and unknown responses lie above it by the assumed margin. □

The proposition is a finite margin statement, not a universal novelty detector. In the executable benchmark the child uses a learned residual and distance gate. When the open-set flag is active, it does not necessarily return a no-op. It routes the remaining experiment budget toward broad generic candidates and retains the same final integrity and capability witness. Thus the measured quantity is false confident known-class assignment rather than raw abstention alone.

75.4 Archive contamination

Let the clean receipt distribution be P and let an ϵ-contaminated archive satisfy

$$\widetilde{P}_\epsilon = (1 - \epsilon)P + \epsilon H, \tag{1087}$$

where H may contain mislabeled obstruction receipts, corrupted outcomes, duplicated evidence, or selectively omitted failures.

Proposition 75.7 (Median breakdown in one coordinate)**.** *Let $x_1, \ldots, x_n \in \mathbb{R}$, and suppose fewer than $n/2$ observations are replaced arbitrarily. Then the sample median of the contaminated set lies between two uncontaminated order statistics. In particular, an adversary controlling less than half the observations cannot drive the median beyond the range of the retained clean central order statistics.*

Proof. At least $\lceil n/2 \rceil$ clean observations remain. Any contaminated median must have at least half of all observations on each side. It therefore cannot cross both central retained clean order statistics. □

Coordinatewise medians and trimmed means inherit finite-sample robustness coordinate by coordinate. They do not provide arbitrary high-dimensional optimality, and they do not repair systematic contamination shared by most archive entries. Block 14 therefore treats contamination-resistant archive estimation as one component of a larger mechanism and reports its ablation separately.

75.5 Evaluator portfolios and integrity ensembles

Let $b_1, \dots, b_m$ be benchmark readouts. A single scored benchmark leaves the local nullspace

$$\mathcal{N}_{\{b_1\}} = \ker Db_1. \tag{1088}$$

An evaluator portfolio has invisible subspace

$$\mathcal{N}_{\mathcal{B}} = \bigcap_{j=1}^{m} \ker Db_j. \tag{1089}$$

Proposition 75.8 (Strict evaluator-nullspace refinement)**.** *If Db_{m+1} is nonzero on some $v \in \mathcal{N}_{\mathcal{B}}$, then*

$$\mathcal{N}_{\mathcal{B}\cup\{b_{m+1}\}} \subsetneq \mathcal{N}_{\mathcal{B}}. \tag{1090}$$

Proof. The new nullspace is the intersection of the old nullspace with $\ker Db_{m+1}$. The vector v belongs to the old nullspace but not the new one, proving strict inclusion. □

The result does not prove that a finite evaluator portfolio is complete. It shows exactly when a new evaluator dimension removes one local metric-gaming direction.

For integrity monitoring, let $M_1, \dots, M_s \in \{0, 1\}$ be monitor pass indicators and let the ensemble pass only when every component passes:

$$M_{\cap} = \prod_{j=1}^{s} M_j. \tag{1091}$$

Proposition 75.9 (Conjunctive monitor false-negative bound)**.** *For an invalid method event I, the ensemble false-negative probability satisfies*

$$\mathbb{P}(M_{\cap} = 1 \mid I) \le \min_j \mathbb{P}(M_j = 1 \mid I). \tag{1092}$$

If monitor errors are conditionally independent given I, then

$$\mathbb{P}(M_{\cap} = 1 \mid I) = \prod_{j=1}^{s} \mathbb{P}(M_j = 1 \mid I). \tag{1093}$$

Proof. The event that every monitor passes is contained in the event that any particular monitor passes. This proves the first bound. Conditional independence gives the product formula. □

Conditional independence is not assumed in the executable claim. The package uses the ensemble operationally and compares it with a weaker monitor ablation. In the frozen final audits, that ablation is null: the robust allocation and final score avoid the declared adaptive attacks even under the inherited monitor. The monitor ensemble therefore receives no independent positive attribution in Block 14.

76 Block 14 Experimental Contract

76.1 Inherited parent and robust child

The inherited parent is the exact Block 13 research compiler reconstructed from the canonical ancestor digest

$$1368fb8a09c3a63ec03ccb05f28e2ba994fc4daf80562d214bf88d29ffaed867. \tag{1094}$$

It uses ordinary archive summaries, one-shot diagnostic routing, broad exploration, one inherited integrity test, and the same final candidate selector used in Block 13.

The robust child retains the parent candidate grammar and final protected objective while adding:

1. coordinatewise robust archive summaries;
2. sequential response experiments inside the fixed six-experiment budget;
3. sparse mixture inference over inherited obstruction components;
4. open-set residual and distance gates;
5. a multi-portfolio robust candidate score;
6. a conjunctive specification and trajectory monitor ensemble.

The selected child configuration is

$$\begin{aligned} \tau_{\text{residual}} &= 0.20, & \tau_{\text{distance}} &= 24, \\ w_{\text{mean}} &= 0.38, & w_{\text{median}} &= 0.10, \\ w_{\text{min}} &= 0.24, & w_{\text{held}} &= 0.22, \\ w_{\text{capability}} &= 0.32. \end{aligned} \tag{1095}$$

It is selected from a 27-point finite meta-validation grid before either final audit is opened.

76.2 Problem groups

Each final audit contains six equally represented groups:

1. **Clean.** One known obstruction with informative but noisy diagnostics.
2. **Ambiguous.** Two inherited classes share an aliased initial diagnostic center.
3. **Multi.** Two obstruction components are simultaneously active.
4. **Unknown.** The response lies outside the inherited known response cone.

5. **Evaluator attack.** High scored performance can conceal poor protected behavior under a rotated audit weighting.
6. **Integrity attack.** Evasive methods can pass the inherited simple integrity rule while violating the stronger trajectory rule.

The primary audit uses generator A and contains 144 problems. The generator-foreign audit uses a shifted generator B, altered diagnostic geometry, and 120 problems. Neither final generator is available during compiler construction.

76.3 Archive contamination and candidate grammar

The development archive is contaminated at rate

$$\epsilon_{\text{archive}} = 0.18. \tag{1096}$$

Contamination can perturb diagnostics, obstruction labels, research outcomes, and response records. The finite candidate grammar contains the exact 39 methods inherited from Block 13: five variants in each of six obstruction-matched families, five generic variants, and the four original integrity and capability controls. Parent and child receive the same canonical candidate order and grammar digest.

The package additionally evaluates contamination levels

$$0, 0.05, 0.10, 0.18, 0.25, 0.35 \tag{1097}$$

on a frozen auxiliary population.

76.4 Matched research budget and chronology

Both compilers receive exactly

$$B_{\text{research}} = 6 \tag{1098}$$

full experiments per problem. The child can use early experiments to estimate response mixtures and allocate later experiments, but it receives no additional full experiment. Candidate metadata, proposal count, evaluator calls, capability checks, and integrity calls are matched.

The chronology is:

$$\begin{gathered}\text{replay Block 13 ancestor} \longrightarrow \text{build contaminated development archive} \\ \longrightarrow \text{fit parent and robust archive models} \longrightarrow \text{select child configuration on meta-validation} \\ \longrightarrow \text{serialize compiler and split commitments} \longrightarrow \text{generate primary and generator-foreign audits} \\ \longrightarrow \text{evaluate every frozen arm and claim gate.}\end{gathered} \tag{1099}$$

The pre-audit seal contains no audit outcomes.

76.5 Comparison arms

The package includes:

Arm	Role
Inherited parent	Exact Block 13-style static compiler under the new adversarial problem distribution
Robust child	Full sequential, open-set, robust-archive, portfolio, and monitor-ensemble compiler
No open set	Child with every problem forced into the inherited known taxonomy
No sequential diagnosis	Child without response-based updating during the six experiments
Ordinary archive	Child decision logic trained on ordinary rather than robust archive summaries
Simple monitor	Child with the inherited weak integrity rule in place of the ensemble
Single portfolio	Child whose final decision uses one evaluator portfolio
Random allocation	Uniform candidate allocation under the same experiment budget
Audit oracle	Inadmissible complete-grammar selection by hidden audit utility

76.6 Frozen claim gates

The robust compiler gate requires:

1. exact Block 13 ancestor replay;
2. identical candidate grammar and six-experiment budget;
3. a compiler and split seal before final outcome access;
4. positive paired child-parent intervals on both final audits;
5. positive child lower-tail gain on both final audits;
6. positive child mean gain in every declared problem group;
7. reduced false confident known-class assignment on unknown problems;
8. zero observed selected invalid methods in both final audits;
9. degraded results for no-open-set, no-sequential, and ordinary-archive controls in their declared dimensions, together with explicit reporting of the null simple-monitor and near-null single-portfolio ablations;
10. an audit oracle that remains at or above the admissible child.

77 Block 14 Results

77.1 Primary and generator-foreign effects

The primary paired results are:

Compiler	Mean gain	Median	$\mathrm{CVaR}_{0.10}$	Invalid selected
Recalibrated parent $\Gamma_1^{\mathrm{recal}}$	0.410813	0.411537	-0.133065	0.027778
Robust child Γ_2	**0.575783**	0.592510	**0.271516**	**0.000000**

Compiler	Mean gain	Median	$\mathrm{CVaR}_{0.10}$	Invalid selected
Random allocation	0.354319	0.305382	0.088336	0.000000
Audit oracle	0.632185	0.656311	0.378857	0.000000

The primary paired effect is

$$\widehat{\Delta}_{\Gamma}^{\mathrm{primary}} = 0.164970, \qquad \mathrm{CI}_{95\%} = [0.115591, 0.218655]. \tag{1100}$$

The sign-flip estimate is

$$p = 9.9999 \times 10^{-6}. \tag{1101}$$

The child improves 46.53% of paired problems, ties on 41.67%, and loses on 11.81%. The losses are therefore retained in the claim rather than removed by a no-negative-pair gate.

On the generator-foreign audit,

$$\mathfrak{C}(\Gamma_1^{\mathrm{recal}}) = 0.389514, \qquad \mathfrak{C}(\Gamma_2) = 0.550559, \tag{1102}$$

so

$$\widehat{\Delta}_{\Gamma}^{\mathrm{foreign}} = 0.161045, \qquad \mathrm{CI}_{95\%} = [0.105335, 0.223141]. \tag{1103}$$

The parent lower-tail gain is -0.193011, while the child lower-tail gain is 0.209497. The child selects no invalid method and has no negative selected gain on this final split.

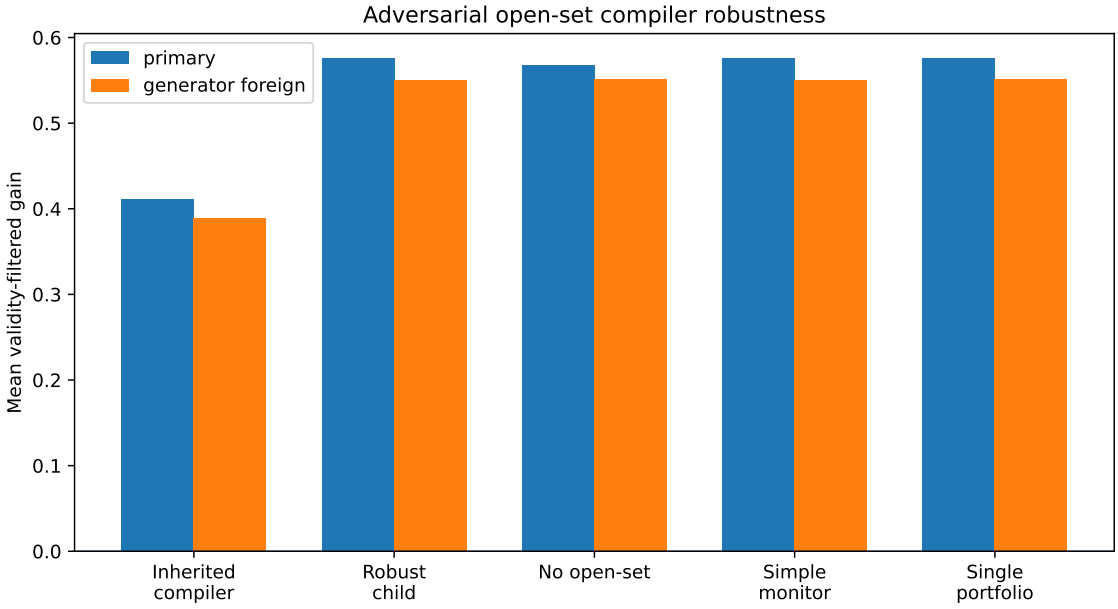


Figure 50: Matched-budget compiler performance on the primary and generator-foreign audits. The robust child raises mean validity-filtered gain while preserving the six-experiment budget.

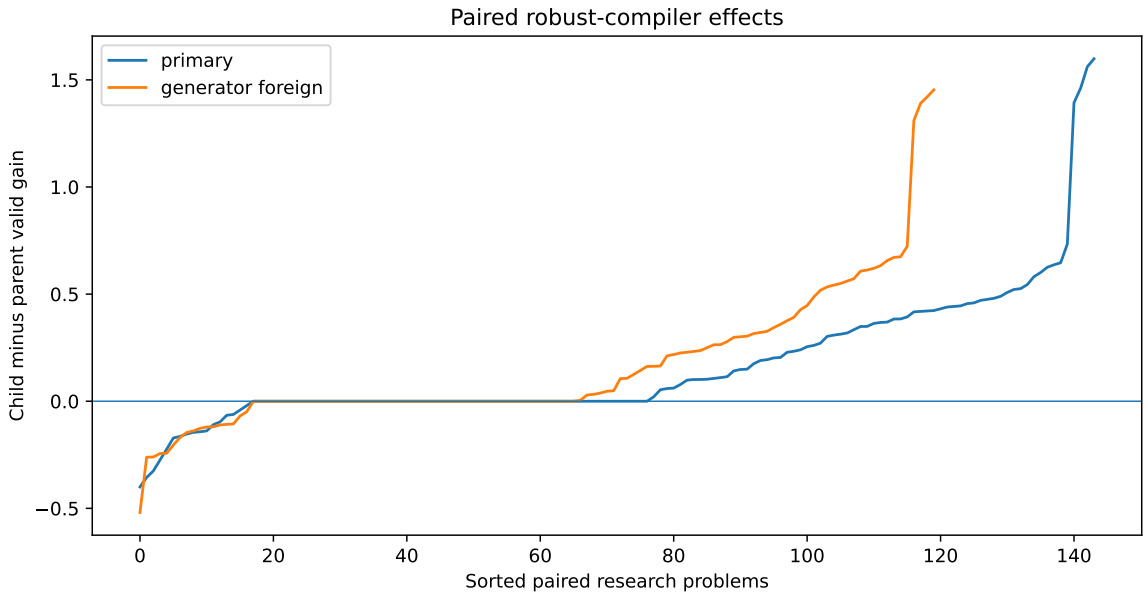


Figure 51: Episode-wise robust-child minus inherited-parent effects. The primary distribution contains retained negative pairs, while the aggregate and generator-foreign effects remain positive.

77.2 Group-resolved robustness

Primary mean gains by problem group are:

Group	Parent	Robust child	Difference
Clean	0.522696	0.672488	0.149792
Ambiguous	0.556180	0.614061	0.057881
Multi-obstruction	0.316613	0.372614	0.056001
Unknown	0.224136	0.446679	0.222543
Evaluator attack	0.405737	0.674109	0.268372
Integrity attack	0.439514	0.674745	0.235231

Every child group mean exceeds 0.37, and every primary group has a positive mean child-minus-parent effect. The paired interval for the ambiguous group still includes zero, so the group-resolved evidence is not uniformly confirmatory. The weakest child groups are the multi-obstruction and unknown families, which remain materially below clean and attack families.

Generator-foreign child means remain positive in every group. The smallest are approximately 0.363 for multi-obstruction problems and 0.396 for unknown problems. Every group mean exceeds its parent mean, but the paired intervals for ambiguous and evaluator-attack groups include zero. The aggregate foreign result is therefore stronger than every individual group claim.

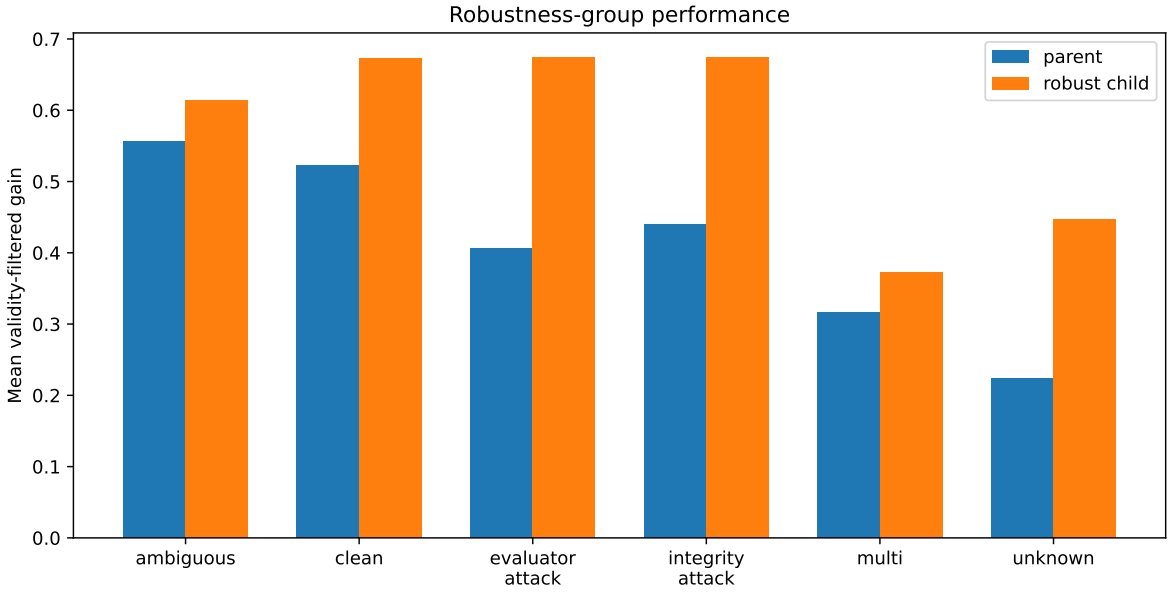


Figure 52: Group-resolved mean validity-filtered gain. The largest primary gains occur on evaluator, integrity, and unknown stress groups; paired group uncertainty remains reported separately.

77.3 Lower-tail and failure-rate behavior

The parent-to-child primary changes are:

$$\mathrm{CVaR}_{0.10} : -0.133065 \longrightarrow 0.271516, \tag{1104}$$

$$\mathbb{P}(G < 0) : 0.027778 \longrightarrow 0, \tag{1105}$$

$$\mathbb{P}(\text{invalid selected}) : 0.027778 \longrightarrow 0. \tag{1106}$$

For the unknown group, false confident known-class assignment changes from

$$1 \longrightarrow 0.083333. \tag{1107}$$

The child still produces a selected method on every primary problem. The improvement is therefore open-set-aware rerouting rather than universal no-op abstention.

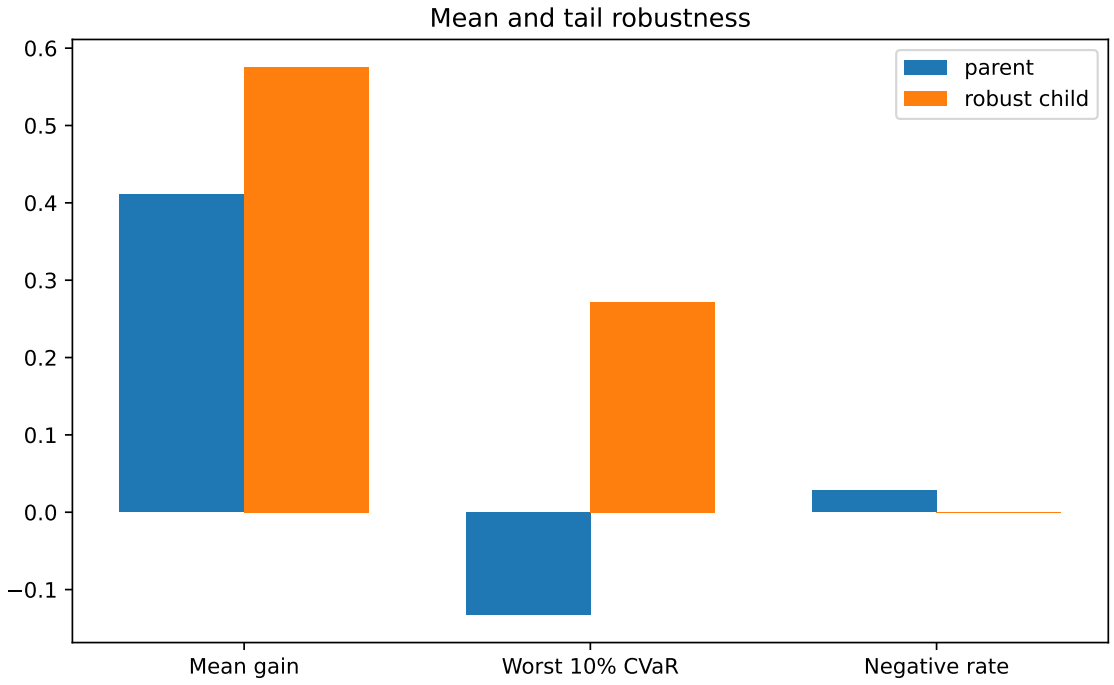


Figure 53: Mean and lower-tail protected gain. The robust child changes the worst decile from substantially negative to positive on both final audits.

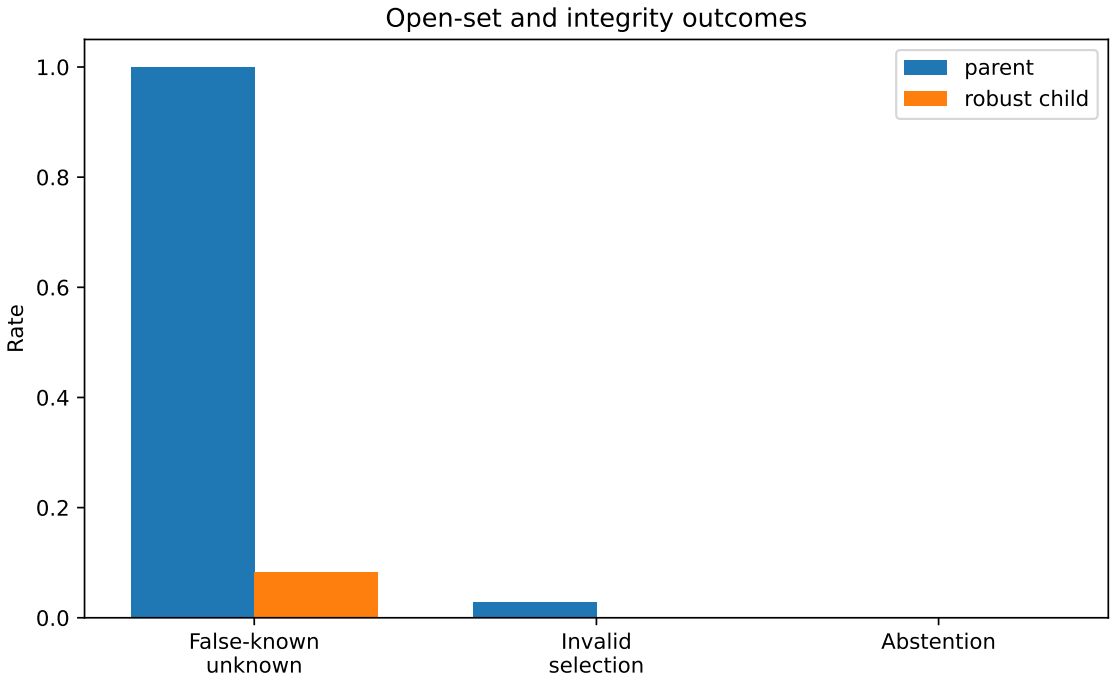


Figure 54: Negative-gain, invalid-selection, and false-known-unknown rates. The complete descendant selects no invalid method and sharply reduces forced known-class assignment. The simple-monitor ablation is identical, so the invalid-selection reduction is not attributed uniquely to the monitor ensemble.

Remark 77.1 (Zero-event interpretation)**.** The child selects zero invalid methods in 144 primary and 120 generator-foreign problems. This is not a zero population-risk theorem. The exact one-sided 95% binomial upper bound reported by the package is approximately 0.02059 for 144 zero-event trials and 0.02465 for 120 zero-event trials.

77.4 Mechanism ablations

The primary ablation summary is:

Arm	Mean gain	$\mathrm{CVaR}_{0.10}$	Key failure
Robust child	0.575783	0.271516	–
No open set	0.568030	0.094979	Unknown false-known rate 1
No sequential diagnosis	0.494244	0.102696	Weak aliased, mixed, and open-set diagnosis
Ordinary archive	0.536955	0.247897	Lower primary mean under contamination
Simple monitor	0.575783	0.271516	Null ablation in the final audits
Single portfolio	0.575283	0.268034	Near-null mean and tail ablation
Random allocation	0.354319	0.088336	No bottleneck-sensitive allocation
Audit oracle	0.632185	0.378857	Inadmissible hidden-audit ceiling

Four qualifications matter.

First, open-set routing primarily improves semantic diagnosis and the lower tail. Disabling it changes primary mean only modestly but forces every unknown problem into a known class and lowers worst-decile gain from 0.271516 to 0.094979. Second, response-based sequential updating is substantively supported: its removal lowers primary mean to 0.494244 and lower-tail gain to 0.102696. Third, robust archive summaries improve primary mean by approximately 0.0388 and remain superior throughout the contamination sweep, but the generator-foreign increment over ordinary summaries is only about 0.0181. Fourth, the simple-monitor ablation is exactly null and the single-portfolio ablation is nearly null. Block 14 therefore does not attribute an independent positive effect to either the monitor ensemble or portfolio aggregation in this final fixture.

77.5 Contamination sweep

Across contamination rates from 0 to 0.35, child mean gain remains between approximately 0.532 and 0.554, while parent mean gain remains between approximately 0.427 and 0.449. Child lower-tail gain remains between 0.248 and 0.277, compared with parent lower-tail gain between 0.113 and 0.127. No invalid method is selected by either arm in this auxiliary sweep, so the comparison supports estimator stability rather than an integrity claim.

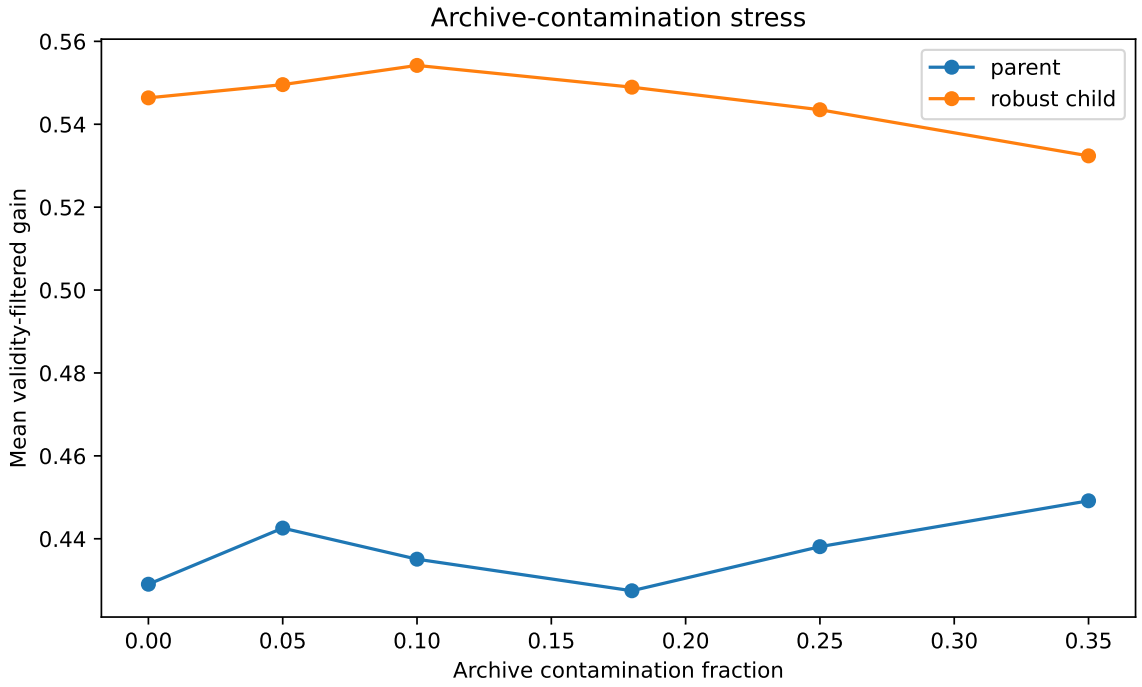


Figure 55: Compiler performance across increasing archive contamination. The robust child remains stable in the finite sweep; the result does not imply robustness beyond the declared contamination process.

77.6 Finite adversarial robustness theorem

Conditional Theorem 77.2 (Block 14 finite robust compiler realization)**.** *Within the frozen Block 14 benchmark, assume that:*

1. *the published Block 13 canonical result digest replays exactly;*
2. *parent and child receive the exact same 39-candidate Block 13 grammar and exactly six full experiments per problem;*
3. *development, meta-validation, primary, and generator-foreign supports are disjoint;*
4. *the compiler object and split commitments are sealed before final outcomes are generated;*
5. *the development archive is contaminated according to the declared finite process;*
6. *primary and generator-foreign paired effects have the intervals in Equations (1100) and (1103);*
7. *every child stress-group mean is positive and both child lower-tail gains are positive;*
8. *the no-open-set, no-sequential, ordinary-archive, simple-monitor, single-portfolio, random, contamination, and oracle controls retain their declared interpretations, including the null monitor and near-null portfolio ablations;*
9. *package, receipt, result, and manifest digests replay exactly.*

Then the edge in Equation (1072) is a finite, sealed, matched-budget robustness improvement of the inherited research compiler relative to the declared diagnostic-aliasing, multi-obstruction, open-set, archive-contamination, evaluator-shift, and integrity-attack atlas.

Proof. Items 1 through 4 establish lineage continuity, matched resources, and audit isolation. Item 5 places both compilers inside the declared contamination regime. Item 6 establishes positive mean paired improvement on two final generators. Item 7 rules out an aggregate gain obtained by sacrificing one entire stress group or the lower tail. Item 8 localizes substantive contributions to sequential diagnosis, open-set routing, and robust archive estimation while retaining the null monitor and near-null portfolio results rather than assigning unsupported causal credit. Item 9 establishes deterministic package closure. The conclusion remains relative to the finite stress atlas and fixed method language. □

78 Block 14 Reproducibility and Claim Boundary

78.1 Reference implementation

The released implementation contains:

- exact replay of the Block 13 canonical result digest;
- clean, aliased, mixed, unknown, evaluator-attack, and integrity-attack problem generators;
- an independently shifted generator-foreign audit;
- the exact common 39-candidate Block 13 method grammar and six-experiment budget;
- contaminated receipt archives with ordinary and robust estimators;
- static and sequential diagnosis, response-mixture estimation, and open-set residuals;
- evaluator-portfolio and integrity-monitor ablations;
- contamination sweeps, paired intervals, sign-flip tests, and lower-tail summaries;

- results-free and post-result receipt chains;
- six publication figures;
- fifteen automated tests;
- an exact package manifest.

The canonical execution prints:

```
SRIC_BLOCK14_ALL_SUITES=PASS
result_sha256=e066fe56a2863b529a0aed51cad8a6e841172bffb231dab252a596a6a58b05b2
primary_parent_gain=0.410813
primary_child_gain=0.575783
primary_effect=0.164970
generator_foreign_parent_gain=0.389514
generator_foreign_child_gain=0.550559
generator_foreign_effect=0.161045
.............. [100%]
SRIC_BLOCK14_PACKAGE_VALID
manifest_files=36
```

78.2 Claim boundary

Block 14 establishes a finite robustness edge for one fixed-language research compiler. It does not establish:

- unrestricted open-set scientific reasoning;
- robustness to arbitrary archive poisoning or adaptive monitor evasion;
- independent external replication;
- confidence sequences for unrestricted adaptive compiler growth;
- autonomous invention of the obstruction taxonomy, experiment language, method grammar, evaluator portfolio, or monitor architecture;
- real large-model training research or open tool use;
- open-ended recursive improvement, indefinite acceleration, unrestricted safety, consciousness, or sentience.

The next threshold has two separable parts. Block 15 addresses the finite compiler-language component by requiring a conflict certificate before adding a diagnostic distinction and method constructors. Independent executable research, external generation, and less scaffolded meta-language growth remain open.

79 Compiler-Language Growth from Open-Set Research Conflicts

Block 14 strengthened a fixed research compiler against diagnostic aliasing, multi-obstruction mixtures, archive contamination, unknown cases, and integrity attacks. Its open-set output nevertheless

remained a single undifferentiated token, and every deployable candidate belonged to the inherited 39-method singleton grammar. Block 15 studies the next structural question:

$$\boxed{\begin{array}{c}\text{Can the compiler use persistent open-set conflict}\\ \text{to enlarge its own diagnostic and method language?}\end{array}} \tag{1108}$$

The target is not unrestricted program invention. It is a finite proof-carrying edge in which the inherited compiler detects that one unknown token aliases research problems requiring incompatible methods, selects a separating experiment from a generic probe language, synthesizes new composite methods from a generic constructor grammar, preserves the inherited compiler as an exact fallback, seals the expanded language, and improves newly generated foreign research problems under a matched deployment budget.

79.1 Compiler languages and inherited quotients

Let $\mathcal{T}$ be a research-problem space. A finite research-compiler language is a tuple

$$\mathfrak{G} = (\mathcal{Z}, \mathcal{P}, \mathcal{C}, \mathcal{B}), \tag{1109}$$

where:

1. $\mathcal{Z}$ is a finite diagnostic-token space;
2. $\mathcal{P}$ is a finite language of admissible diagnostic experiments or probes;
3. $\mathcal{C}$ is a finite method-constructor grammar;
4. $\mathcal{B}$ is a typed binding relation that maps diagnostic tokens to constructed research methods.

A compiler using $\mathfrak{G}$ first maps evidence from problem τ to a token $z \in \mathcal{Z}$, optionally executes probes in $\mathcal{P}$, constructs candidates in $\mathcal{C}$, and selects only bindings admitted by $\mathcal{B}$.

Let $\phi_0 : \mathcal{T} \to \mathcal{Z}_0$ be the inherited diagnostic map. In Block 15, six inherited obstruction classes retain their existing tokens, while two previously unseen obstruction mechanisms are both mapped to one open-set token $\perp$:

$$\phi_0(\tau_{mint}) = \phi_0(\tau_{mtemp}) = \perp. \tag{1110}$$

The equality in Equation (1110) is not itself a defect. An unknown token is an appropriate abstention state until the lineage possesses external evidence that the collapsed cases require incompatible research actions.

For each problem τ, let

$$\mathcal{A}^\star(\tau) \subseteq \mathcal{C} \tag{1111}$$

be the set of methods that satisfy the declared validity, capability, and gain conditions.

Definition 79.1 (Compiler-language conflict)**.** A finite set $S \subseteq \mathcal{T}$ is a compiler-language conflict for $\mathfrak{G}$ when all problems in S receive the same inherited diagnostic token but

$$\bigcap_{\tau \in S} \mathcal{A}^\star(\tau) = \varnothing. \tag{1112}$$

A minimal conflict has no proper conflicting subset.

Theorem 79.2 (Aliased-token routing obstruction)**.** *Let τ_a and τ_b satisfy $\phi_0(\tau_a) = \phi_0(\tau_b)$ and*

$$\mathcal{A}^\star(\tau_a) \cap \mathcal{A}^\star(\tau_b) = \varnothing. \tag{1113}$$

Then no deterministic compiler policy that conditions only on $\phi_0(\tau)$ can select an acceptable method for both problems.

Proof. The policy must output the same method on τ_a and τ_b because its input token is identical. That method would have to lie in both acceptable sets, contradicting their empty intersection. □

The theorem distinguishes a genuine diagnostic-language insufficiency from ordinary classifier error. Additional samples from the same collapsed token cannot solve a routing problem whose correct actions are disjoint.

79.2 Separating research experiments

Let $p \in \mathcal{P}^+$ be a candidate experiment with response

$$y_p : \mathcal{T} \longrightarrow \mathbb{R}. \tag{1114}$$

For threshold θ, define the refined diagnostic map

$$\phi_{p,\theta}(\tau) = \begin{cases} (\phi_0(\tau), \text{low}), & y_p(\tau) \le \theta, \\ (\phi_0(\tau), \text{high}), & y_p(\tau) > \theta. \end{cases} \tag{1115}$$

The old token is recovered by the projection

$$\pi_0(z, s) = z, \qquad \pi_0 \circ \phi_{p,\theta} = \phi_0. \tag{1116}$$

Thus the extension is conservative at the level of old diagnostic semantics.

Definition 79.3 (Conflict-separating experiment)**.** An experiment p separates a conflict S when y_p is not constant on S and every cell of the refined partition has a nonempty common acceptable-method intersection.

Theorem 79.4 (Finite conservative diagnostic refinement)**.** *Suppose a finite experiment family $\mathcal{P}^+$ separates every minimal conflict in a finite development archive. Then repeated addition of separating experiments terminates after finitely many refinements with a diagnostic language whose every development cell is repair consistent. Every old token and old policy remains recoverable by composition with the projection maps.*

Proof. A selected experiment splits at least one unresolved minimal conflict. Refinement never merges cells, so a resolved conflict cannot reappear. The development conflict hypergraph is finite, hence only finitely many successful refinements are possible. Conservative recoverability follows by composing the coordinate projections such as Equation (1116). □

Block 15 realizes one refinement. The two new tokens are deliberately anonymous, `novel-low` and `novel-high`. Their semantic names are retained only in the hidden audit generator. The compiler selects the probe and orientation using development method outcomes, not hidden novel-class labels.

79.3 A generic composite-method constructor

Let $\mathcal{M}_0$ denote the inherited singleton-method grammar. For inherited method families a and b, define an ordered pair constructor

$$\mathrm{Seq}(a,b) = \text{“apply the research intervention from family } a\text{, then the intervention from family } b\text{”}. \tag{1117}$$

The generic finite constructor language is

$$\mathcal{C}_{\mathrm{pair}} = \{\mathrm{Seq}(a,b) : a,b \in \mathcal{M}_0,\ a \neq b\}. \tag{1118}$$

In the executable fixture there are six typed inherited obstruction families, producing 30 ordered pair constructors.

Let $U(\tau, m)$ be protected validity-filtered research gain. Define inherited and extended oracle envelopes

$$O_0(\tau) = \max_{m \in \mathcal{M}_0} U(\tau, m), \qquad O_1(\tau) = \max_{m \in \mathcal{M}_0 \cup \mathcal{C}_{\mathrm{pair}}} U(\tau, m). \tag{1119}$$

The task-relevant structural margin is

$$\kappa_Q = \mathbb{E}_{\tau \sim Q}\left[O_1(\tau) - O_0(\tau)\right]. \tag{1120}$$

A positive margin means that the extension adds useful methods outside the inherited candidate opportunity set.

Proposition 79.5 (Conservative method-language embedding)**.** *The singleton grammar embeds into the expanded grammar by inclusion. A compiler using the expanded language can reproduce the inherited compiler exactly on every old diagnostic token by invoking the inherited binding and ignoring the new probe and pair constructors.*

Theorem 79.6 (Finite conflict-grounded language synthesis)**.** *Let a finite development archive contain one inherited token cell C with at least two incompatible repair subclasses. Suppose:*

1. *a candidate probe $p \in \mathcal{P}^+$ separates C into cells C_- and C_+;*
2. *there exist $m_-, m_+ \in \mathcal{C}_{\mathrm{pair}}$ such that m_- is acceptable throughout C_- and m_+ is acceptable throughout C_+;*
3. *neither one inherited singleton method nor one unrefined binding is acceptable throughout C;*
4. *old-token behavior is preserved through the fallback of Theorem 79.5.*

Then the extension

$$\mathfrak{G}_0 \hookrightarrow \mathfrak{G}_1 = \mathfrak{G}_0 \cup \{p, (\bot, -), (\bot, +), m_-, m_+\} \tag{1121}$$

eliminates the development conflict while conservatively preserving the inherited compiler.

Proof. The refined cells have acceptable witnesses m_- and m_+ by assumption. The inherited conflict is therefore eliminated. Old tokens use the unchanged inherited binding, and the old unknown token is recovered by forgetting the refinement coordinate. The new methods are absent from the singleton grammar, so the extension is structural rather than a renaming. □

79.4 Finite selection and proof-carrying certificates

Let N be the number of candidate probe-and-binding configurations and let their bounded development scores lie in $[0,1]$. Standard uniform concentration gives the following finite-search guard.

Proposition 79.7 (Finite search bound)**.** *For n independent meta-validation problems, with probability at least $1-\delta$, every candidate language score differs from its population score by at most*

$$\varepsilon_n = \sqrt{\frac{\log(2N/\delta)}{2n}}. \tag{1122}$$

Consequently, the selected candidate's population score is within $2\varepsilon_n$ of the best candidate in the finite language.

Proof. Apply Hoeffding's inequality to each candidate and a union bound over N candidates. The $2\varepsilon_n$ selection-regret statement follows by comparing the selected empirical maximizer with the population maximizer. □

A compiler-language expansion receipt is

$$\Pi_{\Gamma\text{-lang}} = (\mathfrak{G}_0, \mathcal{H}_{\text{conflict}}, \mathcal{P}^+, p, \theta, \mathcal{C}_{\text{pair}}, m_-, m_+, \iota, \kappa, \Pi_{\text{seal}}, \Pi_{\text{foreign}}), \tag{1123}$$

where $\mathcal{H}_{\text{conflict}}$ is the conflict hypergraph, ι is the conservative embedding, κ is the newly correctable development margin, and the final fields bind the pre-audit compiler state to foreign evidence.

80 Block 15 Experimental Contract

80.1 Lineage edge and withheld structure

The realized compiler lineage is

$$\Gamma_2 \xrightarrow{\Pi_9^{\Gamma}} \Gamma_3. \tag{1124}$$

The parent Γ_2 is the exact robust fixed-language compiler design from Block 14, recalibrated only inside the inherited research language. It retains:

- six known obstruction tokens and one undifferentiated unknown token;
- the inherited 39-method singleton grammar;
- results-free receipts, evaluator portfolio, capability gates, integrity filtering, and open-set abstention;
- the same eight-unit deployment research budget used by the child in Block 15.

The hidden generator contains two new obstruction mechanisms. Their semantic identities are unavailable to compiler construction. Both occupy the same inherited static-diagnostic cell and therefore produce the same inherited unknown token.

The child receives only:

1. development open-set receipts;

2. a generic six-probe experiment language;
3. a generic 30-member ordered-pair method-constructor grammar;
4. validation outcomes on development and meta-validation problems;
5. the inherited compiler and conservative fallback interface.

The compiler does not receive the hidden names of the two novel mechanisms or a direct declaration of which probe or pair constructors are correct.

80.2 Development conflict and language compiler

The development set contains 56 open-set problems. The inherited base-method oracle obtains mean gain

$$\widehat{O}_0 = 0.377257. \tag{1125}$$

No one inherited unknown-token binding supplies a common high-gain method across the cell.

For every candidate probe, the compiler:

1. evaluates the probe on each development open-set problem;
2. splits the cell at the response median;
3. searches the 30 ordered pair constructors independently in the two refined cells;
4. scores conflict reduction, protected gain, separation, and constructor complexity;
5. records an anonymous two-token diagnostic extension and its two bound methods.

The selected experiment is `interaction-contrast`. Its development separation accuracy, computed only after selection for interpretation against hidden labels, is 1. The selected anonymous bindings are

$$\begin{aligned} \texttt{novel-low} &\longmapsto \texttt{evaluator-then-resource}, \\ \texttt{novel-high} &\longmapsto \texttt{objective-then-constructor}. \end{aligned} \tag{1126}$$

The expanded development gain is

$$\widehat{O}_1 = 0.948962, \tag{1127}$$

with newly correctable margin

$$\widehat{\kappa} = 0.571704. \tag{1128}$$

80.3 Matched deployment budget

Every parent and child run receives exactly eight research units.

Process	Diagnostic probes	Method experiments	Total
Inherited parent on open-set cases	0	8	8
Expanded child on open-set cases	1	7	8
Inherited fallback on known cases	0	8	8

The parent uses all eight units on inherited singleton methods. The child spends one unit on the selected separating probe and at most seven on methods. The child therefore receives no larger

deployment research budget. Compiler development cost is reported separately and is not hidden inside deployment accounting.

80.4 Splits and chronology

The package uses:

Split	Known	Novel	Purpose
Development	60	56	conflict and language compilation
Meta-validation	24	16	conservative child verification
Primary foreign audit	48	48	main sealed comparison
Generator-foreign audit	42	40	shifted diagnostic and outcome geometry

The chronology is

$$\begin{aligned}&\text{replay Block 14 canonical digest}\\&\longrightarrow \text{construct development conflict archive}\\&\longrightarrow \text{search generic probes and pair constructors}\\&\longrightarrow \text{verify conservative fallback on meta-validation}\\&\longrightarrow \text{serialize the language certificate and compiler object}\\&\longrightarrow \text{seal primary and generator-foreign commitments}\\&\longrightarrow \text{open and evaluate both final audits.}\end{aligned} \tag{1129}$$

80.5 Comparison arms

The release includes:

1. **Inherited parent.** One unknown token and the 39-method singleton grammar.
2. **Language child.** Selected separating probe, two anonymous tokens, two synthesized pair constructors, and exact inherited fallback.
3. **Taxonomy only.** The refined tokens are available but no new method constructor is added.
4. **Grammar only.** The pair constructors are available but the unknown cell remains unrefined and receives one pooled binding.
5. **Shuffled binding.** The two refined tokens are bound to the opposite synthesized methods.
6. **Base-grammar oracle.** Inadmissible ceiling over all 39 inherited methods.
7. **Expanded-grammar oracle.** Inadmissible ceiling over inherited methods and all 30 generic pair constructors.

The taxonomy-only, grammar-only, and shuffled-binding controls test whether diagnostic refinement and candidate growth must remain correctly coupled.

80.6 Frozen claim gates

The Block 15 language-growth gate requires:

1. exact replay of the Block 14 result digest;
2. hidden novel labels absent from compiler inputs;
3. a nonzero development conflict margin relative to the inherited method oracle;
4. a selected probe from the generic probe language and selected methods from the generic pair grammar;
5. conservative replay of inherited known-class behavior;
6. identical eight-unit deployment budgets;
7. compiler and split sealing before final audit outcomes;
8. positive paired child-parent effects with intervals excluding zero on both audits;
9. substantial degradation under taxonomy-only, grammar-only, and shuffled-binding controls;
10. an expanded-grammar oracle at or above the admissible child;
11. complete results-free and post-result receipts, tests, figures, and package manifest.

81 Block 15 Results

81.1 Language-inadequacy certificate

The selected finite expansion is summarized by:

Quantity	Value
Inherited open-set development oracle	0.377257
Expanded two-token development gain	0.948962
Newly correctable margin	0.571704
Selected probe	`interaction-contrast`
Probe interpretation accuracy	1.000000
Generic probe candidates	6
Generic ordered pair constructors	30
Deployable inherited methods	39
Deployable child methods	41

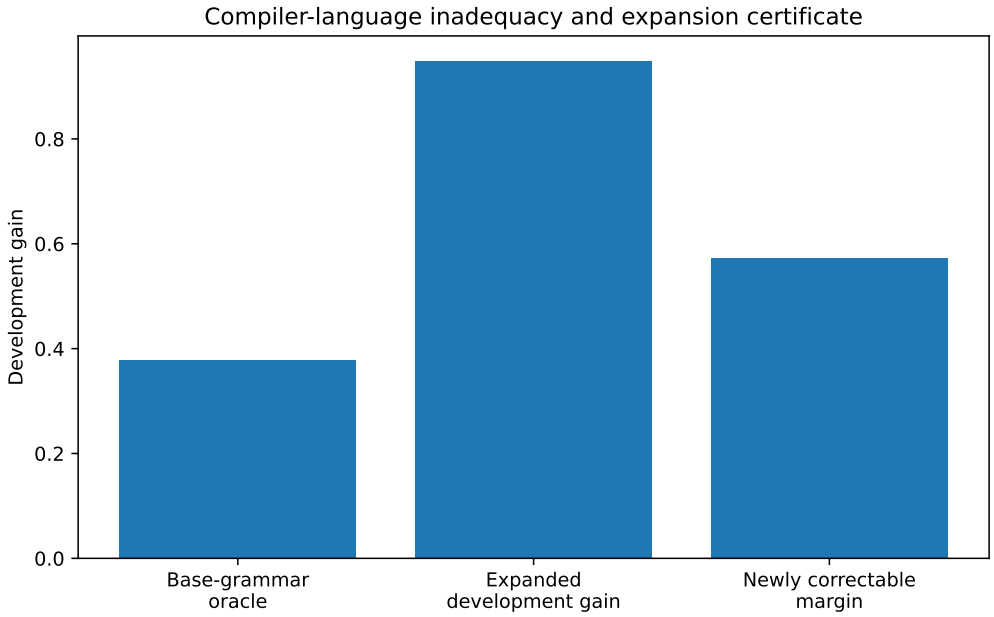


Figure 56: Development language-inadequacy certificate. The inherited singleton oracle leaves a large open-set residual. The selected probe and two synthesized pair constructors expose a substantial new correctable margin.

The deployable child adds only the two selected composites. The remaining 28 pair constructors are development search alternatives, not hidden deployment candidates.

81.2 Primary and generator-foreign compiler effects

The primary audit contains 96 problems. The generator-foreign audit contains 82 problems with shifted diagnostic and outcome geometry.

Audit and process	Mean gain	Novel mean	Worst-10% gain
Primary inherited parent	0.633969	0.341371	0.286117
Primary language child	**0.859731**	**0.792894**	**0.333839**
Generator-foreign parent	0.574224	0.292978	0.240620
Generator-foreign child	**0.814520**	**0.785584**	**0.482289**

The primary paired effect is

$$\widehat{\Delta}^{\text{primary}}_{\Gamma\text{-lang}} = 0.225762, \qquad \text{CI}_{95\%} = [0.166708, 0.286431]. \tag{1130}$$

The generator-foreign paired effect is

$$\widehat{\Delta}^{\text{foreign}}_{\Gamma\text{-lang}} = 0.240296, \qquad \text{CI}_{95\%} = [0.180154, 0.302701]. \tag{1131}$$

Both sign-flip estimates are approximately 9.9999×10^{-6}. No paired problem favors the parent. Primary pairs improve on 36.46% of problems and tie on the remainder; generator-foreign pairs

improve on 42.68% and tie on the remainder. The ties are the exact conservative fallback on inherited known classes and open-set cases for which the selected method matches the parent's selected outcome.

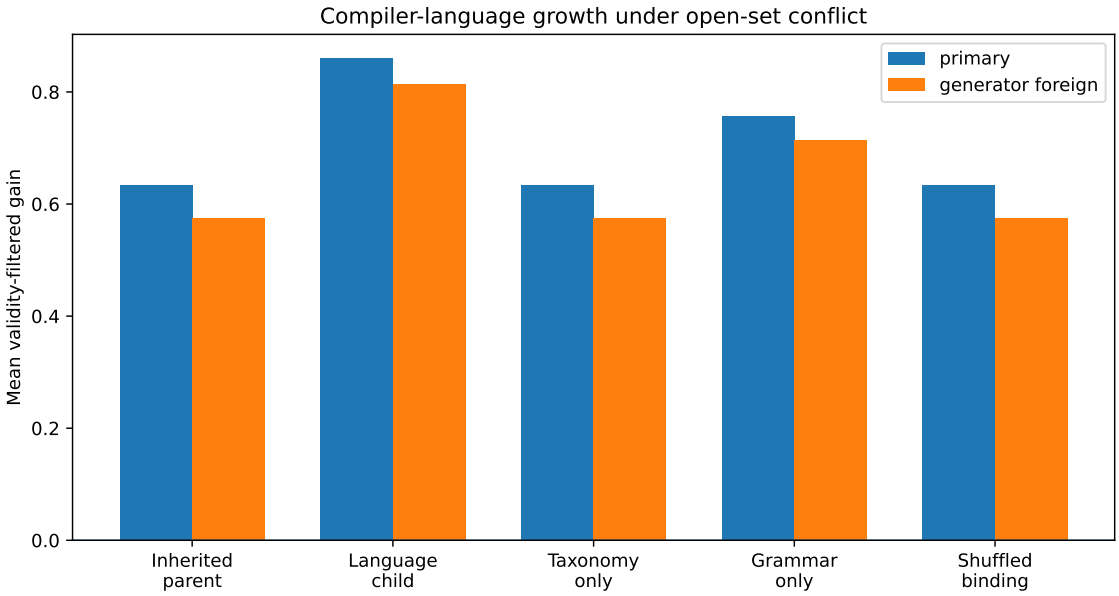


Figure 57: Matched-budget compiler-language performance. Diagnostic refinement without new methods and new methods without the correct refined binding do not reproduce the complete child result.

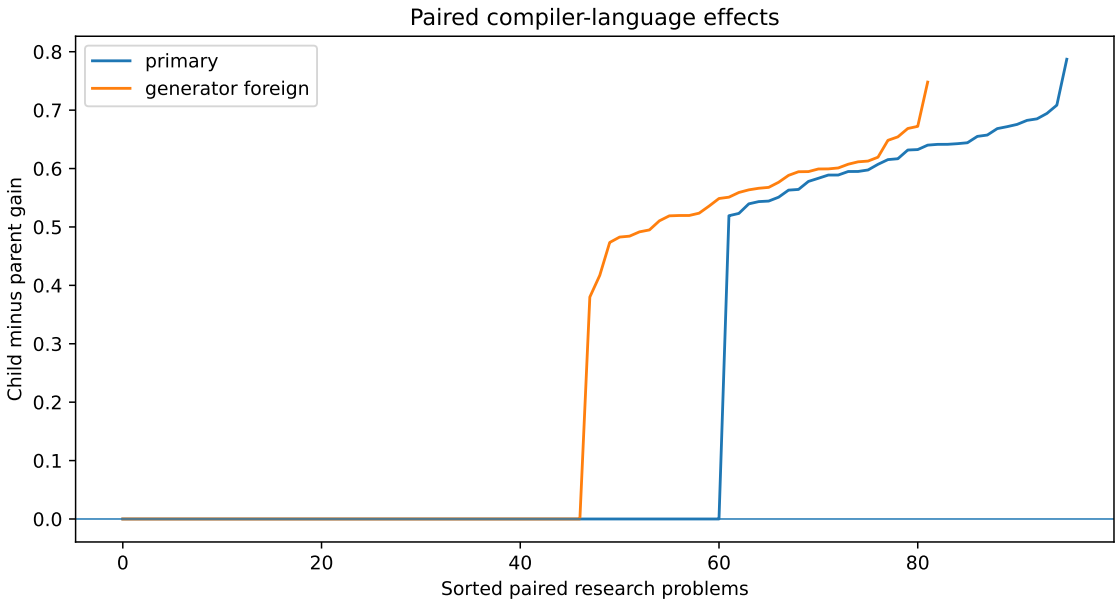


Figure 58: Sorted child-minus-parent effects on primary and generator-foreign research problems. All effects are nonnegative in the finite release; zeros include inherited known-class fallback ties.

81.3 Novel-obstruction performance

The hidden novel families are used only for audit interpretation. On the primary split:

Novel family	Inherited parent	Language child	Difference
Interaction-type	0.327041	0.816123	0.489081
Temporal-type	0.355700	0.769666	0.413965

Under generator shift, the corresponding child means are 0.870372 and 0.700796, compared with parent means 0.295177 and 0.290779.

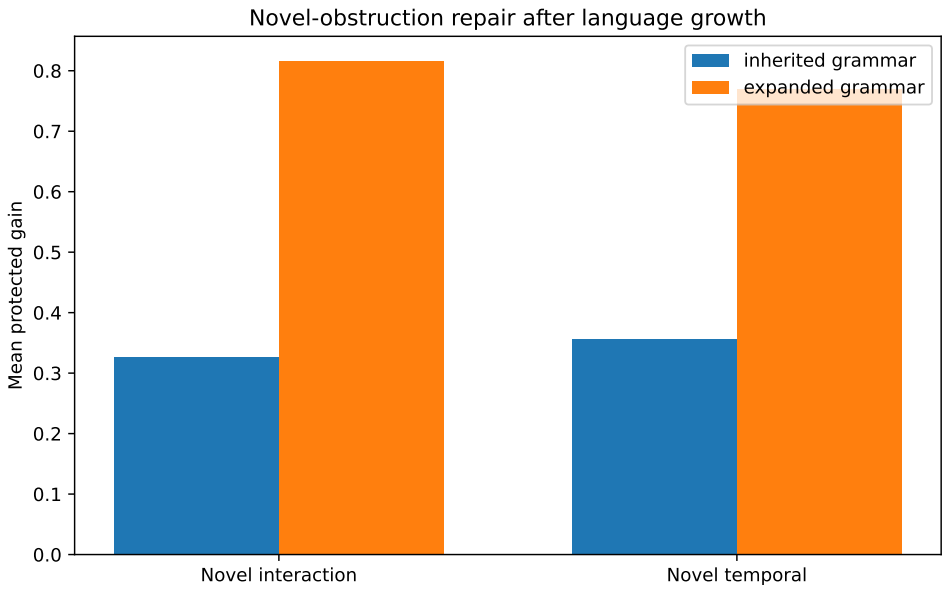


Figure 59: Protected gain on the two hidden novel mechanisms. The compiler uses anonymous response-defined tokens; semantic family names are retained only by the audit generator.

The full inherited-grammar oracle, allowed to inspect all 39 old methods, reaches only 0.373367 mean gain on primary novel problems. The expanded-grammar oracle reaches 0.960732. The admissible child reaches 0.792894 while using only eight deployment units and without final-audit labels.

81.4 Taxonomy, constructor, and binding controls

Primary mean gains are:

Arm	Mean gain
Inherited parent	0.633969
Taxonomy only	0.633995
Grammar only	0.756265
Shuffled binding	0.634112
Full language child	**0.859731**

The taxonomy-only arm cannot improve because the inherited method image remains unchanged. The grammar-only arm improves the novel cell whose pooled binding happens to match the selected composite, but cannot route the second novel mechanism correctly. Shuffled binding removes essentially the entire gain. The result therefore depends on the coupled object:

$$\boxed{\text{new diagnostic distinction} + \text{new method constructor} + \text{correct binding.}} \tag{1132}$$

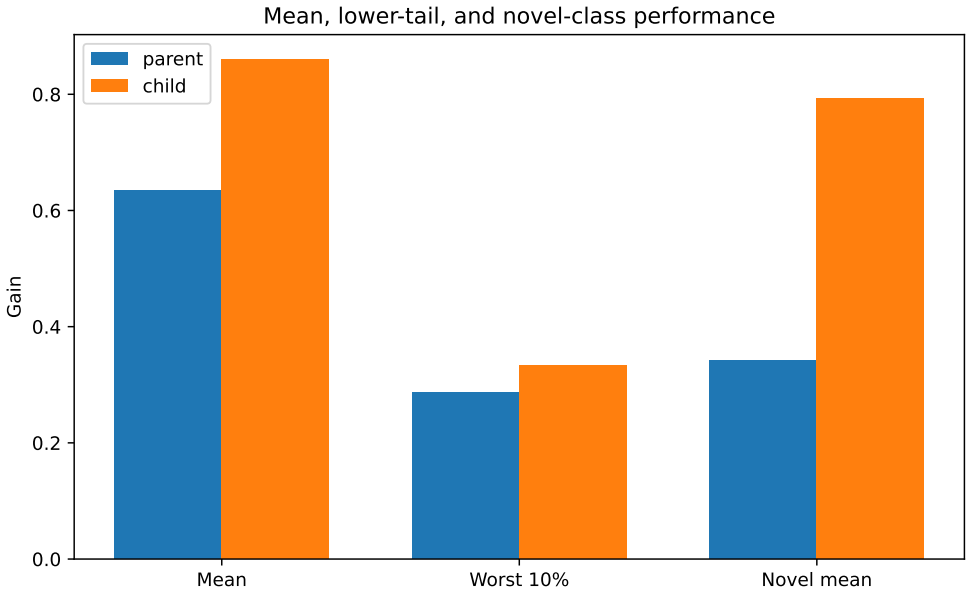


Figure 60: Mean, lower-tail, and novel-class gains for the inherited compiler and the language-expanded child. The largest improvement is concentrated where the inherited language was certified inadequate.

81.5 Conservative inheritance and safety interpretation

For every inherited known class, the child invokes the exact parent path. Consequently, known-class mean gains are byte-for-byte identical between parent and child in the canonical result object. This is stronger than a nonsignificant regression test; it is an executable conservative embedding.

No final selected method fails the declared validity or capability gate. With zero observed invalid selections, the one-sided 95% binomial upper bounds are approximately 3.07% on the 96-problem primary audit and 3.59% on the 82-problem generator-foreign audit. These bounds are finite-sample statements, not proofs of zero population risk.

81.6 Finite compiler-language realization theorem

Conditional Theorem 81.1 (Block 15 finite compiler-language growth)**.** *Within the frozen Block 15 benchmark, assume that:*

1. *the published Block 14 result digest replays exactly;*
2. *hidden novel labels are excluded from compiler construction;*

3. *the parent and child share the same old grammar, evaluator, validity gates, and eight-unit deployment budget;*
4. *the compiler searches only the declared generic probe and ordered pair-constructor languages;*
5. *the inherited open-set cell has the recorded conflict margin, and the selected extension has the recorded development improvement;*
6. *the old compiler is preserved as an exact fallback on inherited tokens;*
7. *the compiler object and audit commitments are sealed before final outcomes;*
8. *the paired effects in Equations (1130) and (1131) have the recorded positive intervals;*
9. *taxonomy-only, grammar-only, shuffled-binding, inherited-oracle, and expanded-oracle controls retain their declared interpretations.*

Then the edge in Equation (1124) is a finite, matched-budget, lineage-generated compiler-language expansion. It adds a diagnostic distinction and two method-constructor outputs absent from the inherited deployable language, conservatively retains old behavior, and reduces a foreign open-set research obstruction.

Proof. Items 1 through 3 establish lineage continuity, inherited-language identity, and resource matching. Items 4 and 5 establish finite structural synthesis relative to an externally declared generic meta-language rather than direct receipt of the correct novel labels. Item 6 gives conservative grammar inclusion. Item 7 supplies audit isolation. Item 8 supplies positive paired foreign metaproductivity. The controls in item 9 rule out mere relabeling, method growth without diagnostic binding, diagnostic growth without method capacity, and complete-grammar oracle violation. The new pair constructors are absent from the inherited singleton grammar, while the selected probe splits a repair-conflicting inherited token; hence the edge changes the compiler language rather than only reallocating experiments inside it. □

82 Block 15 Reproducibility and Claim Boundary

82.1 Reference implementation

The released package contains:

- the exact Block 14 canonical result digest;
- known and hidden-novel problem generators with a separately shifted foreign generator;
- the inherited 39-method singleton grammar;
- a generic six-probe diagnostic language and 30 ordered pair-constructor grammar;
- conflict scoring, probe selection, anonymous token creation, constructor synthesis, and conservative fallback;
- primary, generator-foreign, taxonomy-only, grammar-only, shuffled-binding, inherited-oracle, and expanded-oracle arms;

- results-free and post-result receipts, compiler and pre-audit seals, deterministic figures, tests, and an exact package manifest.

The canonical execution prints:

```
SRIC_BLOCK15_ALL_SUITES=PASS
result_sha256=309d63546ecbbbcca1f1a92a6ca48c5b2bae752df4496c1f2d333c31798fc630
primary_parent_gain=0.633969
primary_child_gain=0.859731
primary_effect=0.225762
generator_foreign_parent_gain=0.574224
generator_foreign_child_gain=0.814520
generator_foreign_effect=0.240296
selected_probe=interaction_contrast
selected_methods=evaluator_then_resource,objective_then_constructor
.............. [100%]
SRIC_BLOCK15_PACKAGE_VALID
manifest_files=33
```

82.2 Claim boundary

Block 15 establishes a finite compiler-language growth edge from one inherited unknown token and a singleton-method grammar to a response-refined two-token diagnostic language with two synthesized pairwise methods. It does not establish:

- unrestricted invention of diagnostic experiments or research methods;
- discovery outside the declared six-probe and ordered-pair constructor meta-language;
- automatic semantic naming of the new obstruction classes;
- general multi-step program synthesis, arbitrary method composition, or real model-training research;
- improvement of the evaluator portfolio or integrity monitor;
- external independent replication;
- open-ended compiler self-improvement, indefinite acceleration, unrestricted safety, consciousness, or sentience.

The next major threshold is *cross-implementation executable research*. The language-growing compiler should face actual program-synthesis or small-model training problems generated by an independently written environment, with method effects produced by executed code rather than a finite outcome simulator. A stronger descendant should also decide whether a residual requires another diagnostic token, a longer method program, a new evaluator dimension, or a no-go certificate, rather than receiving a fixed pair-composition meta-language.

83 Direct Neuralese Causal Semantics

Block 15 treats a research compiler's diagnostic and method language as an object that can be conservatively enlarged after a persistent open-set conflict. The next direct application moves the

same principle inside a recurrent latent reasoner. A *Neuralese* system, as instantiated by the released Hyper v1 runtime [17], does not emit a token at each internal step. It repeatedly transforms and reinjects a continuous hidden state, then exposes an answer through an ordinary readout. The central question is therefore not whether one can assign an English sentence to every hidden vector. It is whether the hidden trajectory is causally necessary, sequentially organized, specific to the current problem, stable under exact replay, transportable across descendants, and improvable under a protected lineage transaction.

This block makes Neuralese a first-class SRIC component. The latent transition, update gate, halting rule, functional packet atlas, semantic transport map, and external readout are typed mutable objects. A descendant may alter one or more of them, but the edge is accepted only through matched-resource causal audits and a frozen witness. The construction directly connects hidden-state reasoning to the repair, obstruction, transport, holonomy, and compiler-language machinery developed in the previous blocks.

Block 16 status

The released companion contains a direct adapter contract for the local Cymela Hyper v1 runtime, an intervention suite using its exposed latent-state hook, gold-answer loss scoring, cross-task transplantation, exact replay, matched-norm noise, step permutation, halting diagnostics, orthogonal semantic-transport certificates, parent–child lineage comparison, receipts, tests, and manifests. The package does not redistribute the Hyper runtime or weights. The finite synthetic fixture passes and separates a preview-like parent with useful but nonspecific latent computation from a task-specific child. No empirical Hyper run is claimed in this revision because the model weights were not present in the build environment.

83.1 Neuralese reasoners as typed lineage objects

Definition 83.1 (Neuralese reasoner)**.** A finite-step Neuralese reasoner is a tuple

$$\mathcal{N} = (\mathcal{X}, \mathcal{Y}, H, E, B, U, D, \Lambda, K), \tag{1133}$$

where $\mathcal{X}$ is an input space, $\mathcal{Y}$ an answer space, H a finite-dimensional latent state space, $E : \mathcal{X} \to H$ a prompt encoder, $B : H \times \mathcal{X} \to H$ a backbone transition, $U : H \times H \times H \to H$ an update gate, $D : H \times \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ an answer readout, $\Lambda : H \to [0, 1]$ a halting map, and K a maximum latent-step budget.

Given input x, define the prompt anchor $h_0 = E(x)$. A normalized injection state z_t and updated latent state h_t satisfy

$$z_t = N(h_{t-1}), \qquad \widetilde{h}_t = B(z_t, x), \qquad h_t = U(z_t, \widetilde{h}_t, h_0), \qquad 1 \le t \le K, \tag{1134}$$

where N is any declared normalization or scale-matching operator. The answer is sampled or selected from $D(h_K, x)$ after a declared end-of-thought transition.

The tuple in Equation (1133) is intentionally structural. In the released Hyper v1 runtime, B is the Qwen backbone restricted to one reinjected hidden slot, U is the task-anchored gated latent update, Λ is the PonderNet-style halt head, and the exposed `latent_intervention` function acts directly on z_t before the backbone and halt head consume it. The SRIC adapter imports those objects from the user's local model folder rather than reproducing them.

Definition 83.2 (Neuralese trace)**.** For input x and fixed depth K, the injected Neuralese trace is

$$Z_K(x) = (z_1(x), \dots, z_K(x)) \in H^K. \tag{1135}$$

The updated trace is $H_K(x) = (h_1(x), \dots, h_K(x))$. Every receipt must state which trace convention is being intervened on. Block 16 uses the injected trace because this is the point at which the Hyper runtime exposes an exact intervention hook.

The distinction is material. A trace captured after the update gate and a trace captured immediately before reinjection are related but are not interchangeable under arbitrary interventions. An audit that silently changes this convention cannot be compared with its ancestor receipt.

83.2 The direct intervention language

Let $L(x, y; Z)$ denote mean gold-answer cross-entropy when the prompt is x, target answer is y, and the recurrent latent slots are externally set to trace Z. Let $Z(x)$ denote the clean trace. For a donor x' selected from the same declared answer-format group, let $Z(x')$ be its clean trace. Block 16 fixes the intervention language

$$\mathcal{I}_{\mathrm{NL}} = \{\mathsf{zero}, \mathsf{replay}, \mathsf{permute}, \mathsf{noise}, \mathsf{transplant}, \mathsf{packet}, \mathsf{transport}\}. \tag{1136}$$

The first five interventions are mandatory in the minimal direct audit.

1. zero executes no latent step.
2. replay replaces every injected state by the clean captured state at the same step.
3. permute applies a sealed nonidentity permutation π to the clean trace.
4. noise replaces each state by deterministic Gaussian noise with matched Euclidean norm and shape.
5. transplant replaces $Z(x)$ by $Z(x')$ from a distinct task in the same answer-format group.

The packet and transport interventions are optional at the first checkpoint. They become mandatory when a descendant claims a localized causal packet or cross-model latent semantics.

Definition 83.3 (Per-task Neuralese causal coordinates)**.** For task (x_i, y_i), define

$$n_i := L(x_i, y_i; \mathsf{zero}) - L(x_i, y_i; Z(x_i)), \tag{1137}$$
$$o_i := L(x_i, y_i; Z_\pi(x_i)) - L(x_i, y_i; Z(x_i)), \tag{1138}$$
$$c_i := L(x_i, y_i; Z_\eta(x_i)) - L(x_i, y_i; Z(x_i)), \tag{1139}$$
$$s_i := L(x_i, y_i; Z(x_i')) - L(x_i, y_i; Z(x_i)), \tag{1140}$$
$$r_i := \left| L(x_i, y_i; \mathsf{replay}(Z(x_i))) - L(x_i, y_i; Z(x_i)) \right|. \tag{1141}$$

These are respectively latent necessity, order sensitivity, corruption sensitivity, task specificity, and replay error.

The distinction between o_i and s_i is the direct bridge to the current Neuralese obstruction. A trajectory may be highly ordered while remaining nearly exchangeable across prompts. In that

regime the recurrence computes a generic answer-form trajectory rather than a content-conditioned solution trajectory.

Definition 83.4 (Structural-normalized specificity)**.** Let

$$Q_{\mathrm{spec}} = \frac{\overline{s}}{\max\{\overline{o}, \overline{c}, \varepsilon\}}, \tag{1142}$$

where bars denote held-out means and $\varepsilon > 0$ is a fixed numerical floor. The default Block 16 specificity gate requires

$$\mathrm{LCB}_{0.95}(\overline{s}) > 0, \qquad \frac{1}{n}\sum_i \mathbf{1}[s_i > 0] \geq 0.75, \qquad Q_{\mathrm{spec}} \geq 0.25. \tag{1143}$$

The ratio in Equation (1142) prevents a very small positive transplant effect from clearing the gate merely because it is statistically nonzero. The task-specific signal must be material relative to at least one strong structural intervention on the same trace.

Theorem 83.5 (Format-conditional transplant null)**.** *Let $G(x)$ be a declared answer-format class. Suppose that, conditional on $G(x) = g$, the trace law is independent of task content:*

$$Z(x) \mid G(x) = g \overset{d}{=} Z(x') \mid G(x') = g, \tag{1144}$$

where x' is an independent donor. Assume the intervention harness leaves the prompt and target answer fixed and replaces only the latent trace. Then

$$\mathbb{E}[s_i \mid G(x_i) = g] = 0. \tag{1145}$$

Proof. Under the hypothesis, the own-task and donor traces have the same conditional distribution. The measurable losses $L(x_i, y_i; Z(x_i))$ and $L(x_i, y_i; Z(x_i'))$ therefore have the same conditional expectation. Their expected difference is zero. □

Corollary 83.6 (Meaning of a small transplant cost)**.** *A small cross-task transplant cost is compatible with useful and order-sensitive recurrent computation. It specifically fails to establish that the trace carries information about the current task beyond the donor-matching variables used to define the transplant group.*

Proposition 83.7 (Replay integrity)**.** *Assume deterministic execution, identical prompt and suffix tokens, identical numerical kernels, and a hook that replaces each injected state by the exact state captured at that step in the clean run. Then $r_i = 0$ for every task.*

Proof. Induct on the latent step. At step one the backbone receives the same prompt cache and injected state, so it produces the same updated state and cache. If the caches agree through step $t-1$, exact replacement supplies the same state at step t, hence the same next cache. The final answer-scoring trajectory is therefore identical. □

Nonzero replay error is not automatically a model defect. It may reveal nondeterministic kernels, lossy trace serialization, a misplaced hook, a suffix-alignment error, or hidden mutable state. It nevertheless invalidates causal attribution until bounded and explained.

83.3 Adaptive halting as a separate claim

For fixed maximum depth $K_{\max}$, let $\lambda_t(x) = \Lambda(z_t(x))$. Define cumulative halt mass

$$M_t(x) = \sum_{j=1}^{t} \lambda_j(x) \prod_{q<j} (1 - \lambda_q(x)), \tag{1146}$$

and dynamic depth

$$T(x) = \min\{t : M_t(x) \geq \tau\}, \tag{1147}$$

with declared threshold τ. Block 16 does not infer adaptive computation from the existence of a halt head. It requires a difficulty variable $d(x)$ fixed before result access and tests both

$$\rho_{\text{Spearman}}(d(x), T(x)) > 0 \tag{1148}$$

and nondegenerate prompt variation in the halt schedules. A constant depth can coexist with fully task-specific Neuralese; conversely, variable depth can be driven by nuisance prompt features. The causal and adaptive statuses are therefore reported separately.

83.4 Functional packets and gauge-invariant Neuralese semantics

A coordinate-level decoder is not the identifiability target. Let P_x be a projector onto a trace packet discovered from declared interventions. Packet necessity and sufficiency are evaluated by removing $P_x Z(x)$, retaining only $P_x Z(x)$, and comparing equal-rank random and PCA controls. The packet is accepted only when its causal effect transfers to held-out tasks and survives within-packet basis changes.

For parent and child state spaces H_A and H_B of equal dimension, let calibration traces produce matrices X and Y. The orthogonal Procrustes transport is

$$T^* = \operatorname*{arg\,min}_{T^\mathsf{T} T = I} \|XT - Y\|_{\mathrm{F}}. \tag{1149}$$

If $X^\mathsf{T} Y = U \Sigma V^\mathsf{T}$, then $T^* = UV^\mathsf{T}$. A transport receipt records calibration residual, held-out residual, singular floor, round-trip residual, and behavior change after transported trace injection.

Proposition 83.8 (Exact orthogonal latent transport)**.** *Suppose $Y = XQ$ for an orthogonal Q and X has full column rank. Then the minimizer in Equation (1149) is Q, the calibration residual is zero, and the round trip $X \mapsto XT^* \mapsto XT^*(T^*)^\mathsf{T}$ is exact.*

Proof. The objective has value zero at Q. Full column rank implies that $XT = Y = XQ$ only when $T = Q$. Orthogonality gives $QQ^\mathsf{T} = I$. □

This is a gauge statement rather than a lexical translation statement. Neuralese meaning is represented by intervention and readout invariants under admissible coordinate transformations, not by a preferred list of neuron names.

83.5 The direct Neuralese lineage edge

Extend the SRIC lineage state by

$$\mathfrak{L}_t^{\mathrm{NL}} = (\mathfrak{L}_t, \mathcal{N}_t, \mathcal{I}_t^{\mathrm{NL}}, \mathcal{T}_t^{\mathrm{NL}}, \mathcal{W}_t^{\mathrm{NL}}), \tag{1150}$$

where $\mathcal{T}_t^{\mathrm{NL}}$ contains trace-transport certificates and $\mathcal{W}_t^{\mathrm{NL}}$ is the frozen external answer and invariant witness. A proposed descendant may modify the latent transition, update gate, halting head, packet atlas, or transport map. The backbone may be held fixed for a component-isolation edge or changed under a stronger whole-model edge.

Definition 83.9 (Neuralese improvement receipt)**.** A Block 16 receipt contains:

1. parent and child model fingerprints;
2. exact prompt, answer, donor-group, split, and task digests;
3. latent hook location and trace convention;
4. fixed step budget and numerical precision;
5. all five mandatory per-task intervention losses;
6. bootstrap intervals and positive-effect fractions;
7. specificity ratio and halting diagnostics;
8. transport and packet certificates when claimed;
9. pre-audit commitments, witness outputs, rollback state, and claim boundary.

Conditional Theorem 83.10 (Finite Neuralese metaproductivity edge)**.** *Let parent $\mathcal{N}_0$ and child $\mathcal{N}_1$ be evaluated on identical sealed tasks with identical fixed latent-step budgets, answer-scoring rules, donor construction, numerical precision, evaluator, and witness. Assume:*

1. *both audits satisfy replay integrity;*
2. *the child satisfies latent necessity, ordered computation, corruption sensitivity, and task specificity on primary and generator-foreign supports;*
3. *the paired child-minus-parent transplant-cost interval has positive lower endpoint;*
4. *child clean loss does not regress beyond a predeclared tolerance;*
5. *every claimed packet or transport passes its own held-out causal certificate;*
6. *model, task, audit, and result digests are committed in an append-only receipt chain before promotion.*

Then the accepted transition

$$\mathfrak{L}_t^{\mathrm{NL}} \xrightarrow{\Pi_{16}} \mathfrak{L}_{t+1}^{\mathrm{NL}} \tag{1151}$$

is a finite evidence-preserving improvement of the Neuralese reasoning component. If the child-producing mechanism, rather than an external developer, generated the accepted latent modification under the same search budget, the transition is additionally a bounded improvement-process edge.

Proof. Replay integrity establishes that the intervention path is faithful to the declared recurrent slot. Necessity, order, corruption, and transplant gates separate a useful task-specific latent trajectory from a decorative, unordered, or answer-format-only trajectory. The paired interval and clean nonregression establish a held-out improvement rather than an isolated task effect. Packet and transport certificates preserve the semantics of any localized or cross-model claim. Frozen receipts, witness separation, and rollback preserve the SRIC transaction invariants. The final distinction follows because improving a latent subsystem and improving the process that constructs latent subsystems are different claims. □

83.6 Executable mapping to Hyper v1

The direct adapter uses the following correspondence.

SRIC Block 16 object	Local Hyper v1 runtime object
Prompt encoder E	Tokenized ChatML prefix ending in `<bot>`
Injected state z_t	Normalized state immediately before recurrent backbone reinjection
Backbone transition B	One cached call through the wrapped Qwen backbone
Update gate U	`latent_update_gate(h, h_new, h_prompt)`
Halting map Λ	`halt_head` evaluated on the injected state
Intervention interface	`LatentCausalLM.latent_intervention`
Gold-answer witness	Cross-entropy from the end-of-thought state through the exact supplied suffix tokens
Replay control	Replace each recurrent injection by its captured own-task state
Permutation control	Reinject a sealed permutation of the captured own-task states
Noise control	Reinject deterministic norm-matched Gaussian states
Transplant control	Reinject a same-format donor task's captured states
Trace readout	Gold first-token rank from the final updated latent state

The adapter dynamically imports the official `hyper_chat.py` from a local model folder and checks for the declared method and hook. It does not patch the model source. It temporarily disables the runtime's textual logit-lens printout during scoring, because nearest-token labels are probes rather than the latent computation itself. Exact answer scoring uses the model's own tokenizer and end-of-thought path.

83.7 Finite synthetic realization

The companion's engineered fixture contains 66 held-out tasks: 36 primary and 30 generator-foreign. Parent and child use four latent steps. The parent has a strong generic answer-format trajectory and a weak task-content channel. The child strengthens the task-content channel and makes dynamic depth depend on declared difficulty. Both are analytic synthetic systems; the purpose is to validate the audit logic and failure boundary, not to make a claim about a trained language model.

Quantity	Preview-like parent	Task-specific child
Mean clean loss	0.235716	0.000405
Mean latent-necessity gain	0.647409	0.882720
Mean order cost	0.182441	0.276445
Mean norm-matched noise cost	0.982047	1.364676
Mean cross-task transplant cost	0.016648	0.456634
Structural-normalized specificity	0.016952	0.334610
Difficulty–depth Spearman correlation	0	0.969320
Distinct dynamic depths	1	5
Causal status	PARTIAL	FULL
Adaptive-halting status	FAIL	PASS

The paired child-minus-parent transplant-cost effect is 0.439986 with frozen bootstrap interval $[0.386884, 0.493143]$. Mean clean-loss improvement is 0.235311, replay error is exactly zero in both arms, and the finite lineage receipt passes all declared gates.

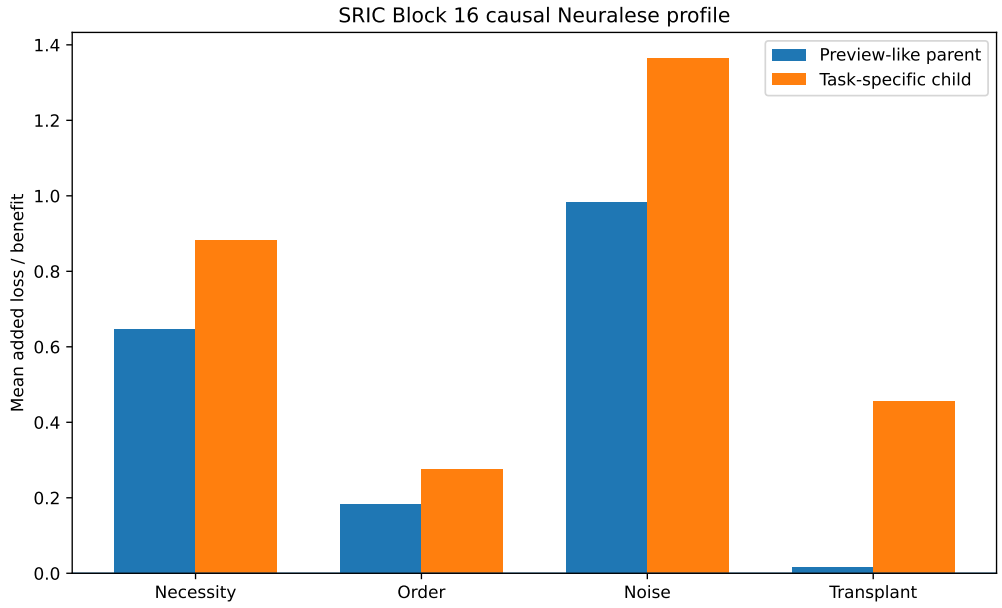


Figure 61: Synthetic direct Neuralese audit. Both systems have useful ordered recurrent computation, but only the child has a material problem-specific transplant effect.

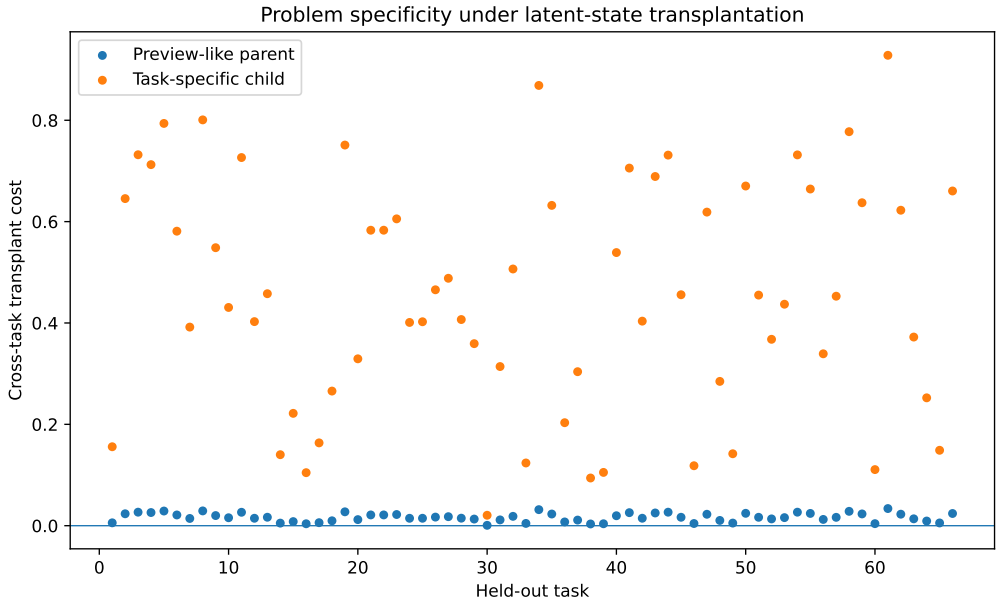


Figure 62: Per-task transplant costs in the synthetic primary and generator-foreign audits. The parent remains near the format-conditional transplant null while the child is consistently task specific.

83.8 Hyper audit chronology

A real Hyper audit follows the frozen sequence

$$\begin{aligned}\text{fingerprint local runtime and weights} &\longrightarrow \text{freeze task and donor digests} \longrightarrow \text{capture clean traces}\\ &\longrightarrow \text{execute zero, replay, and permutation arms}\\ &\longrightarrow \text{execute noise and transplant arms}\\ &\longrightarrow \text{compute paired intervals and gates}\\ \longrightarrow \text{serialize result and receipt digests} &\longrightarrow \text{promote, reject, or retain as PARTIAL.}\end{aligned} \tag{1152}$$

The default task file contains 27 short generated arithmetic, Boolean, and ordering problems with declared difficulty and answer-format groups. The default command limits the first run to 12 tasks because every task requires several full recurrent forward passes. Increasing the support is necessary before making a strong empirical claim.

83.9 Claim boundary and next threshold

Block 16 establishes a direct formal and executable bridge between SRIC and recurrent latent reasoning. The released positive control proves that the audit can distinguish ordered but nonspecific latent computation from task-specific latent computation and can certify a finite parent–child edge. It does not establish:

- that Hyper v1 passes or fails the local audit in the user's environment;
- that nearest-token projections faithfully translate latent states;

- that any identified packet is a complete account of the model's reasoning;
- robustness to adversarially optimized latent traces or adaptive evaluator evasion;
- causal specificity beyond the supplied task and donor-group construction;
- a training-time child that autonomously improves its own latent update or halt head;
- unrestricted Neuralese translation, safety, awareness, consciousness, sentience, or open-ended recursive self-improvement.

The next threshold is a genuine two-checkpoint executable lineage. A parent checkpoint and a child produced by a declared training or search process should be audited on sealed primary and independently generated foreign tasks. The child should increase task-specific transplant cost relative to structural controls, preserve or improve clean performance, maintain replay integrity, and either improve adaptive depth or issue an explicit saturation certificate. Only then does Neuralese move from a directly audited internal substrate to a demonstrated recursively improved reasoning component.

84 Executable Neuralese Binding-Repair Compilation

Block 16 made recurrent latent computation directly auditable, but its accepted parent–child edge was an analytic positive control. The parent and child trajectories were generated by declared formulas rather than learned checkpoints, and one same-format transplant statistic could still confound problem-specific computation with answer encoding. Block 17 crosses the next finite threshold. It trains a recurrent neural parent, compiles six matched-update descendants, seals one checkpoint before final-task generation, preserves a failed first audit without weakening its gates, and evaluates the revised child on fresh primary and shifted-generator tasks produced by a separately implemented exact environment.

The object modified by the compiler is no longer a table of simulated method effects. It is an executable recurrent state transition whose trajectories are captured, replayed, reversed, corrupted, transplanted, decoded, gauge transformed, and scored by independent exact witnesses. The result remains bounded: the task family, candidate grammar, loss, donor taxonomy, semantic target, and acceptance thresholds are supplied. Within that boundary, the accepted edge is a training-generated Neuralese repair rather than an analytic intervention fixture.

Block 17 status

The released companion contains an exact rational affine-program environment, a trained recurrent latent parent, six inherited repair candidates, a finite compiler, a pre-audit checkpoint seal, a preserved failed pilot, a precommitted compiler revision, fresh primary and generator-foreign audits, four donor classes, an external trace probe, adaptive-halting tests, an independent NumPy shadow runtime, a latent-gauge certificate, receipts, figures, tests, and an exact package manifest. In the successful second round, the task-blind parent is `EXECUTABLE_PARTIAL` on both final supports and the selected full-binding child is `EXECUTABLE_FULL` on both. This is one finite package-local executable Neuralese repair edge. It is not a result for Hyper, autonomous open-ended repair invention, or independent external replication.

84.1 Exact executable research environment

Let an affine instruction be

$$f_j(z) = a_j z + b_j, \qquad a_j \in \left\{-1, -\tfrac{1}{2}, \tfrac{1}{2}, 1\right\}, \qquad b_j \in \left\{-\tfrac{1}{2}, 0, \tfrac{1}{2}\right\}. \tag{1153}$$

A depth-m program acts on $x \in \{-1, -\frac{3}{4}, \ldots, 1\}$ by

$$F_m(x) = (f_m \circ \cdots \circ f_1)(x) = A_m x + B_m, \tag{1154}$$

where

$$A_0 = 1, \qquad B_0 = 0, \qquad A_j = a_j A_{j-1}, \qquad B_j = a_j B_{j-1} + b_j. \tag{1155}$$

All quantities in Equations (1153) to (1155) are evaluated with exact rational arithmetic before a floating representation is exposed to the model.

Definition 84.1 (Executable affine-program task)**.** A Block 17 task is

$$\tau = (x, m, (a_j, b_j)_{j=1}^m, A_m, B_m, y_m, g, s), \qquad y_m = A_m x + B_m, \tag{1156}$$

where $g \in \{A, B\}$ names the generator and s names the data split. The model input is a padded serialization of x, the instructions, and instruction-presence bits. The protected semantic witness is

$$w(\tau) = \left(A_m, \frac{B_m}{3}, \frac{y_m}{4}\right) \in \mathbb{R}^3. \tag{1157}$$

The environment module contains no model code. Its verifier reconstructs every instruction from the released record, recomputes A_m, B_m, and y_m as instances of `Fraction`, checks the affine interpreter invariant, checks the floating witness, and rejects duplicate task identifiers. Generator B changes the coefficient and offset probabilities while preserving the exact program language and interpreter. Thus generator foreignness is a declared distribution shift, not a change in semantics.

The released supports are:

Support	Tasks	Generator	Role
Training	10,000	A	Parent and matched repair optimization
Development	1,400	A	Compiler development and diagnostics
Meta-validation	1,400	A	Frozen candidate eligibility and selection
Probe calibration	1,200	A	External ridge-probe fitting
Primary audit	1,600	A	Fresh final in-generator evaluation
Generator-foreign audit	1,600	B	Fresh shifted-distribution evaluation

The successful round contains 17,200 exact task records. The first failed pilot used different final seeds and remains archived separately. No successful-round final task appears in training, development, meta-validation, probe calibration, or the pilot final supports.

Proposition 84.2 (Exact witness soundness)**.** *For every released task τ, the environment verifier accepts only if its stored witness equals the exact composition witness in Equation (1155). Hence a model cannot obtain a valid audit result from a malformed or internally inconsistent target record.*

Proof. The verifier reconstructs x and each (a_j, b_j) as rational numbers, independently executes the recurrence in Equation (1155), and compares the resulting triple exactly with the stored rational witness. The floating target in Equation (1157) is then checked against the exact triple to tolerance 10^{-12}. Any disagreement produces a failed environment certificate before model evaluation. □

84.2 The recurrent parent and repair grammar

Let $\phi(\tau) \in \mathbb{R}^{19}$ be the padded task serialization, and let

$$e = E_\theta(\phi(\tau)) \in \mathbb{R}^{40} \tag{1158}$$

be the prompt representation. The latent trace has four states. Given conditioning mode α, define

$$z_0 = \zeta, \qquad z_{k+1} = \tanh(Rz_k + s_k + C_\alpha(e, m)), \qquad 0 \le k < 4, \tag{1159}$$

where s_k is a learned step embedding. The task-blind parent satisfies

$$C_{\text{parent}}(e, m) = 0. \tag{1160}$$

The descendants use one of the following conditioning maps:

$$C_{\text{depth}}(e, m) = D\,\mathbf{1}_m, \tag{1161}$$
$$C_1(e, m) = U_1 V_1 e, \qquad \operatorname{rank}(U_1 V_1) \le 1, \tag{1162}$$
$$C_4(e, m) = U_4 V_4 e, \qquad \operatorname{rank}(U_4 V_4) \le 4, \tag{1163}$$
$$C_{\text{full}}(e, m) = We. \tag{1164}$$

Here $\mathbf{1}_m$ is the one-hot depth code. Each child inherits every compatible parent parameter. The added up-projection or full-conditioning matrix is initialized so that the initial injected signal is zero, giving a conservative functional starting point before repair training.

The latent trace updates the task representation through

$$g_k = \sigma(Gz_k), \tag{1165}$$
$$\widetilde{h}_k = \tanh(Th_{k-1} + Lz_k), \tag{1166}$$
$$h_k = (1 - g_k) \odot h_{k-1} + g_k \odot \widetilde{h}_k, \qquad h_0 = e, \tag{1167}$$
$$\widehat{w}_k = Oh_k. \tag{1168}$$

A separate halt head predicts one of four dynamic depths from the mean trace and, for nonparent candidates, the encoded prompt. A semantic training head predicts $w(\tau)$ from z_4. Its auxiliary loss is an externally declared witness term, not a claim that the latent coordinates are uniquely interpretable.

Definition 84.3 (Finite Neuralese repair grammar)**.** The Block 17 compiler grammar is

$$\mathcal{G}_{17} = \{\mathsf{parent}^+, \mathsf{depth}, \mathsf{rank1}, \mathsf{rank4}, \mathsf{full}, \mathsf{full}^{-\mathrm{sem}}\}. \tag{1169}$$

The symbol parent^+ denotes matched-update continuation of the task-blind parent. The final symbol uses full conditioning but removes the semantic-witness loss. Every candidate receives

exactly 550 AdamW updates with batch size 256, learning rate 8×10^{-4}, and identical ordinary task, intermediate-step, halt, and weight-decay terms.

The common serialized architecture has 14,370 parameters. Active parameter counts range from 12,130 for the task-blind continuation to 13,730 for full conditioning. The experiment matches updates, examples, data, optimizer, and base initialization, but not exact floating-point operations across candidate ranks. This is a matched-training-envelope comparison rather than an exact FLOP identity claim.

Proposition 84.4 (Nested conditioning images)**.** *For fixed encoder E_θ, the rank-one and rank-four conditioning families are contained in the full linear conditioning family:*

$$\{U_1 V_1\} \subseteq \{U_4 V_4\} \subseteq \{W\}. \tag{1170}$$

The task-blind parent is recovered by the zero map in every family.

Proof. Every rank-one linear operator has rank at most four and therefore admits a rank-four factorization after padding with zero columns and rows. Every rank-four factorization is a linear map and therefore belongs to the unconstrained full family. Setting every active conditioning weight to zero gives Equation (1160). □

84.3 Why one transplant statistic is insufficient

Block 16 used same-format cross-task transplant cost. In a prediction problem, that cost can be positive even if the trace represents only the final answer. Program semantics and answer semantics must therefore be separated.

Let $d_{\mathrm{prog}}(\tau, \tau')$ be a normalized distance between (A_m, B_m) and $(A'_{m'}, B'_{m'})$, and let

$$d_{\mathrm{ans}}(\tau, \tau') = \left| y_m/4 - y'_{m'}/4 \right|. \tag{1171}$$

For a recipient task τ_i, define four donor classes:

1. SAP: same-answer, far-program, approximately depth-matched donors;
2. FAD: far-answer, depth-matched donors;
3. SN: semantic-neighbor donors;
4. RMD: random depth-matched donors.

Three donors per available class are selected by deterministic task-identifier hashing. Each audit records donor coverage and the realized semantic and answer distances.

Theorem 84.5 (Answer-only transplant confound)**.** *Suppose a latent trace is conditionally answer sufficient but program blind:*

$$Z(\tau) = \Psi(y(\tau), m(\tau), \xi), \tag{1172}$$

where ξ is independent internal noise, and suppose the answer readout genuinely uses Z. Then far-answer transplantation may have strictly positive expected cost even though Z contains no information about (A_m, B_m) conditional on (y, m). Consequently, a positive ordinary cross-task transplant cost does not establish program-specific Neuralese.

Proof. Choose two programs with the same depth but different answers. Under Equation (1172), the donor trace is distributed as a code for the donor answer rather than the recipient answer. If the readout uses this code, replacing the recipient trace can increase recipient loss. However, conditional on (y, m), the trace distribution is independent of (A_m, B_m) by assumption. Thus the positive cost is caused by answer mismatch, not by represented program structure. □

Corollary 84.6 (Same-answer program control)**.** *Under the hypotheses of Theorem 84.5, exact same-answer and depth-matched donors have zero expected transplant cost, even when their affine programs are far apart. A positive* SAP *cost therefore rejects the exact answer-only null relative to the declared donor construction.*

Warning 84.7 (Residual identifiability boundary)**.** The SAP control does not prove that every aspect of a program is represented. It establishes only that the transplanted trace carries recipient-relevant information not reducible to the matched answer and depth variables under the released task and donor distribution. Unmeasured sufficient statistics can remain.

84.4 Executable causal and semantic gates

Let $\ell_i(Z)$ be mean squared witness error on recipient i when the four latent states are externally supplied as Z. Let Z_i be the clean trace. The structural coordinates are

$$n_i = \ell_i(\mathsf{zero}) - \ell_i(Z_i), \tag{1173}$$
$$o_i = \ell_i(\mathsf{reverse}(Z_i)) - \ell_i(Z_i), \tag{1174}$$
$$c_i = \ell_i(\mathsf{noise}(Z_i)) - \ell_i(Z_i), \tag{1175}$$
$$r_i = \left|\ell_i(\mathsf{replay}(Z_i)) - \ell_i(Z_i)\right|. \tag{1176}$$

Noise is deterministic and norm matched at every step. The program and answer specificity coordinates are the mean recipient costs under SAP and FAD donors:

$$p_i = \mathbb{E}_{j\in\mathsf{SAP}(i)}\left[\ell_i(Z_j) - \ell_i(Z_i)\right], \qquad a_i = \mathbb{E}_{j\in\mathsf{FAD}(i)}\left[\ell_i(Z_j) - \ell_i(Z_i)\right]. \tag{1177}$$

Define

$$Q_{\mathrm{prog}} = \frac{\overline{p}}{\max\{\overline{o}, \overline{c}, \varepsilon\}}, \qquad Q_{\mathrm{ans}} = \frac{\overline{a}}{\max\{\overline{o}, \overline{c}, \varepsilon\}}. \tag{1178}$$

The frozen ratio thresholds are 0.15 and 0.20, respectively. Each specificity gate additionally requires a positive 95% bootstrap lower endpoint and a positive-effect fraction of at least 0.55. Structural necessity, order, and noise gates use the same lower-endpoint condition and a positive-effect fraction of at least 0.55. Exact replay requires

$$\max_i r_i \le 10^{-10}. \tag{1179}$$

The local-continuity gate compares far-program cost with semantic-neighbor cost. Its paired lower confidence endpoint must be positive. This prevents the classifier from treating every foreign trace as equally damaging while claiming a meaningful semantic geometry.

Definition 84.8 (External trace probe)**.** A Block 17 trace probe is a ridge map

$$\widehat{w} = P\, z_4 + q \tag{1180}$$

fit only on the 1,200 probe-calibration tasks. The final audit reports held-out coefficient of determination

$$R^2_{\text{probe}} = 1 - \frac{\sum_i \|w_i - \widehat{w}_i\|^2}{\sum_i \|w_i - \overline{w}\|^2}. \tag{1181}$$

The frozen semantic-probe gate is $R^2_{\text{probe}} \geq 0.65$ on each final support.

The probe does not assign natural-language meanings to individual coordinates. It tests whether an independently fit, held-out linear witness can recover exact task semantics from the final latent state. The independent signed-permutation check below confirms that this claim is invariant to a large family of coordinate relabelings.

The adaptive-halting gate requires a Spearman correlation of at least 0.70 between exact program depth and selected latent depth, at least four distinct selected depths, prompt-dependent halt probabilities, and mean dynamic-output loss no more than 0.015 above fixed-depth output loss.

Definition 84.9 (Executable Neuralese status)**.** An audit is `EXECUTABLE_FULL` when all structural gates, both specificity gates, local semantic continuity, donor coverage, and the external trace-probe gate pass. It is `EXECUTABLE_PARTIAL` when the structural gates pass but at least one semantic-binding gate fails. Otherwise it is `EXECUTABLE_FAIL`. Adaptive halting is reported as a separate PASS or FAIL status.

84.5 Sealed compiler chronology

The parent receives 650 updates with batch size 256. Every candidate then begins from that exact checkpoint and receives the common 550-update repair budget. Before any final-task generation, the compiler evaluates all candidates on meta-validation and applies the hard gates

$$\begin{gathered}\text{clean MSE within } 0.002 \text{ of the matched-update parent},\\ R^2_{\text{probe}} \geq 0.65, \quad Q_{\text{prog}} \geq 0.15, \quad Q_{\text{ans}} \geq 0.20,\\ \text{positive interval and fraction gates}, \quad \text{local continuity}, \quad \text{donor coverage}, \quad \text{adaptive halting}.\end{gathered} \tag{1182}$$

Among eligible candidates it maximizes

$$\begin{aligned} S(\alpha) = \min\left\{ \frac{R^2_{\text{probe}}}{0.65}, \frac{Q_{\text{prog}}}{0.15}, \frac{Q_{\text{ans}}}{0.20} \right\} + 0.20 R^2_{\text{probe}} \\ - 4\,\text{MSE}_{\text{clean}} + 0.02\rho_{\text{halt}} - \mathsf{Penalty}_{\text{dynamic}}. \end{aligned} \tag{1183}$$

The minimum term makes the weakest normalized semantic gate the principal bottleneck. The compiler seal commits the selected candidate, all candidate and parent state hashes, training receipts, meta-validation task digest, exact thresholds, selection rule, pilot-failure receipt, and post-pilot revision receipt before the successful-round final tasks are generated.

84.6 A failed sealed pilot and nonretroactive repair

The first compiler round used a narrower grammar and 450 matched candidate updates. Its score over-weighted trace decodability. It selected full conditioning, and the child improved absolute program and answer transplant effects, probe R^2, and adaptive depth. Nevertheless, it remained

`EXECUTABLE_PARTIAL` on both final supports because the normalized program and answer specificity ratios missed the already frozen FULL thresholds. The lineage transaction was rejected with receipt

`cc0d07b20ec2820a40f9781babc36f467482bed9ce5319f031401c3ba026c318`

The negative result was not overwritten. The package archives the failed checkpoints, final tasks, audits, figures, and receipt. Before a second final audit, the revision file committed four changes:

1. add a rank-four conditioning candidate;
2. allocate 550 repair updates to every candidate;
3. use the weakest-normalized-margin selection score in Equation (1183);
4. generate new primary and generator-foreign supports from seeds 17006 and 27006.

No final threshold was weakened or removed. The revision receipt is

`f7435a94a4c1d065ff5e37d3895a58d7fca541faf2098a8872ac9f90c6915755`

This chronology makes the failed pilot evidence about compiler design rather than hidden tuning evidence for the successful final tasks.

Protocol 84.10 (Block 17 sealed repair transaction)**.** 1. Generate and exactly verify training, development, meta-validation, and probe-calibration supports.

2. Train and serialize the task-blind parent.
3. Inherit the parent into every candidate in Equation (1169).
4. Apply the same repair-update and example budget to each candidate.
5. Fit the external probe only on probe-calibration traces.
6. Evaluate every candidate on meta-validation with the complete donor and halting audit.
7. Select by Equations (1182) and (1183) and seal all state and protocol digests.
8. Only after sealing, generate and exactly verify the primary and generator-foreign final supports.
9. Audit the matched parent and selected child on both supports.
10. Commit only if every lineage gate passes; otherwise retain the failed receipt and rollback to the parent.
11. Independently replay the environment witnesses, checkpoint states, audit receipts, NumPy forward equations, and gauge certificate.

84.7 Compiler result

Three of the six candidates passed every meta-validation hard gate. The full-binding candidate had the largest frozen score and was selected.

Candidate	Eligible	Clean MSE	Probe R^2	Q_{prog}	Q_{ans}
Matched parent continuation	no	0.012385	−0.000901	0	0

Candidate	Eligible	Clean MSE	Probe R^2	Q_{prog}	Q_{ans}
Depth only	no	0.012172	0.000984	0	0
Rank-one binding	yes	0.012001	0.699509	0.8054	0.9229
Rank-four binding	yes	0.011779	0.861346	0.3937	0.5036
Full binding	**selected**	0.011452	0.870542	0.7436	1.1703
Full without semantic witness	no	0.010728	0.724937	0.1019	0.1050

The final ablation has the lowest clean meta-validation error but fails both causal-specificity ratios. This prevents predictive performance or decodability alone from being credited as task-bound latent semantics.

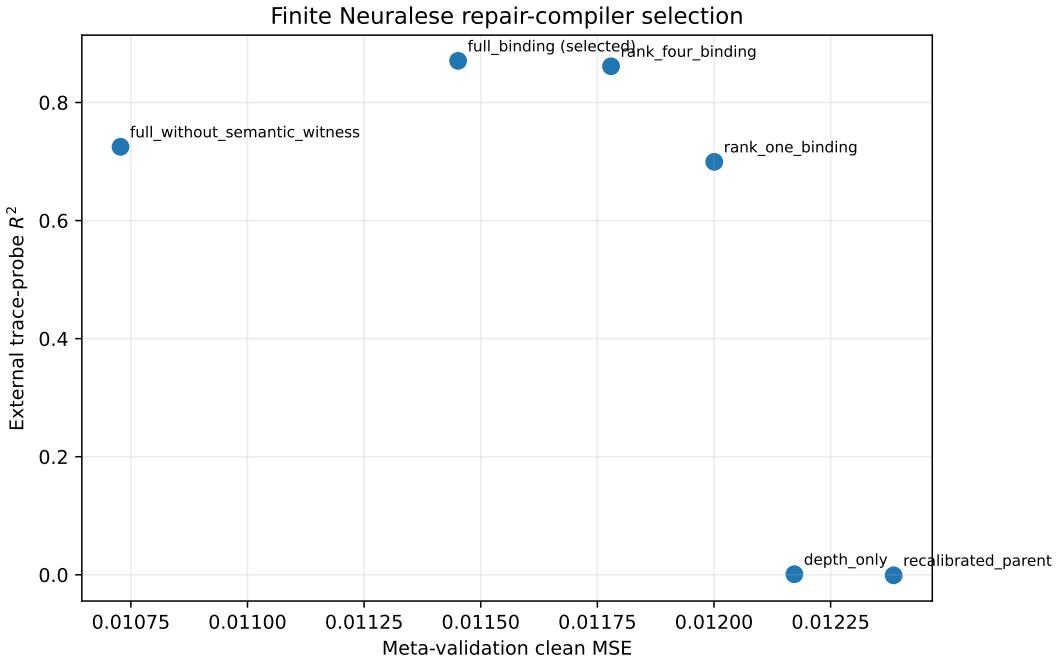


Figure 63: Block 17 repair compiler. Eligibility requires all semantic, causal, replay, donor, clean-loss, and halting gates. Full binding is selected by the frozen weakest-margin score; the no-semantic-witness ablation is predictively strong but causally ineligible.

84.8 Fresh primary and generator-foreign audits

The primary audit contains 1,600 generator-A tasks created after the compiler seal. The matched parent retains useful, ordered, and corruption-sensitive recurrence, but its latent trajectory is exactly task blind under the conditioning interface. The selected child passes every structural and semantic-binding gate.

Primary quantity	Parent	Child
Clean MSE	0.012665	0.011686
Latent-necessity gain	0.017036	0.030296
Step-order cost	0.000957	0.004362

Primary quantity	Parent	Child
Norm-matched noise cost	0.001272	0.014644
Same-answer far-program cost	0	0.010714
Far-answer matched-depth cost	0	0.017727
Program specificity ratio	0	0.731642
Answer specificity ratio	0	1.210526
External trace-probe R^2	−0.000302	0.865504
Difficulty–depth Spearman ρ	0	0.971716
Distinct dynamic depths	1	4
Causal status	PARTIAL	FULL
Adaptive-halting status	FAIL	PASS

The paired child-minus-parent program-specificity gain is

$$\widehat{\Delta}^{A}_{\text{prog}} = 0.010714, \qquad \text{CI}_{0.95} = [0.009911, 0.011527], \tag{1184}$$

with positive fraction 0.823125. The answer-specificity gain is 0.017727. Its frozen interval is

$$[0.016490, 0.018960].$$

Parent-minus-child clean loss is 0.000979, with frozen interval $[0.000695, 0.001290]$.

The shifted audit contains 1,600 generator-B tasks with altered coefficient and offset probabilities. The parent remains PARTIAL and the child remains FULL.

Generator-foreign quantity	Parent	Child
Clean MSE	0.062210	0.058746
Latent-necessity gain	0.036128	0.051896
Step-order cost	0.000866	0.004750
Norm-matched noise cost	0.001353	0.017566
Same-answer far-program cost	0	0.024763
Far-answer matched-depth cost	0	0.035400
Program specificity ratio	0	1.409704
Answer specificity ratio	0	2.015268
External trace-probe R^2	−0.002355	0.691471
Difficulty–depth Spearman ρ	0	0.969169
Distinct dynamic depths	1	4
Causal status	PARTIAL	FULL
Adaptive-halting status	FAIL	PASS

The paired generator-foreign program-specificity gain is

$$\widehat{\Delta}^{B}_{\text{prog}} = 0.024763, \qquad \text{CI}_{0.95} = [0.022794, 0.026793], \tag{1185}$$

with positive fraction 0.855. The answer-specificity gain is 0.035400 with interval $[0.032956, 0.037938]$. Clean-loss improvement is 0.003464 with interval $[0.002633, 0.004324]$. Monte Carlo paired sign-flip

tests for the program-specificity gains return 9.999×10^{-5} on each final support.

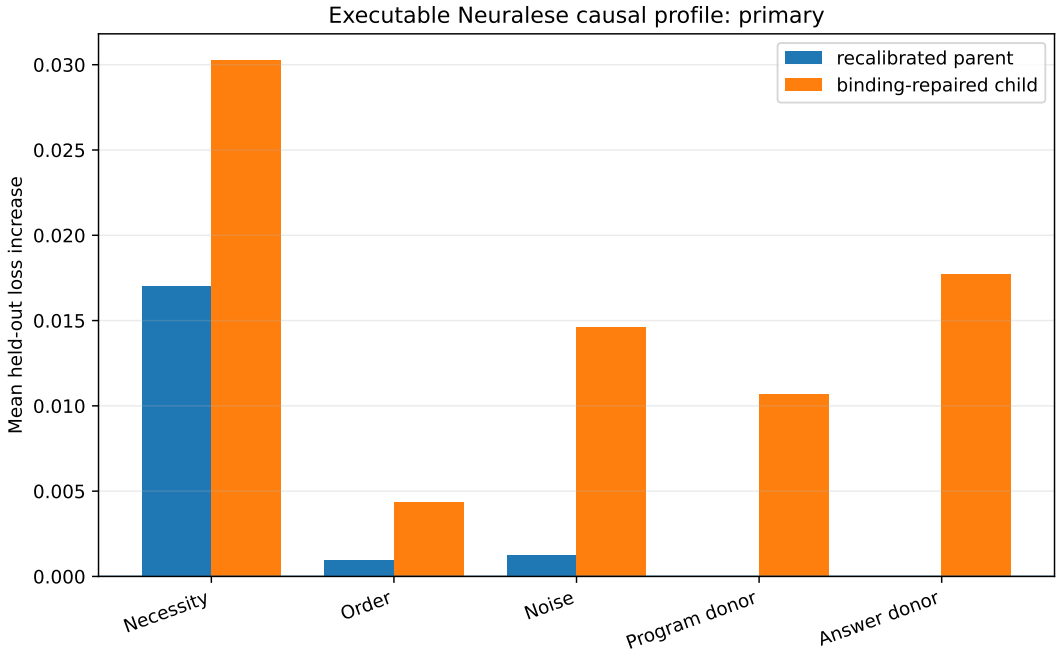


Figure 64: Fresh primary audit. The parent is structurally recurrent but task blind; the selected child adds program- and answer-specific latent effects while improving clean witness error.

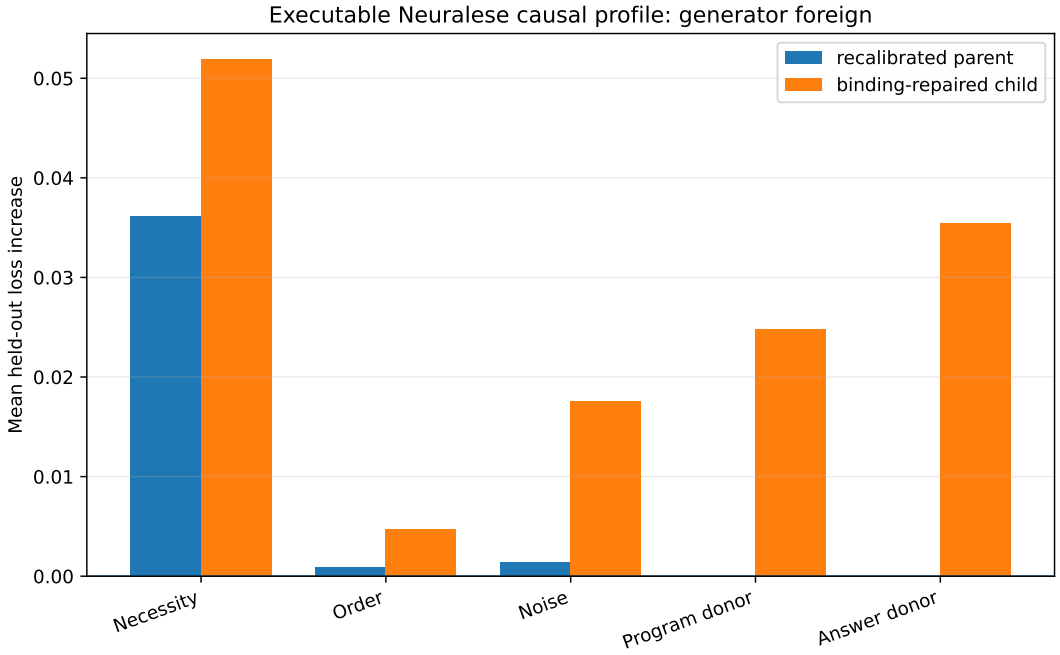


Figure 65: Shifted-generator audit. The selected child retains FULL causal status and positive semantic-binding margins under the declared distribution shift.

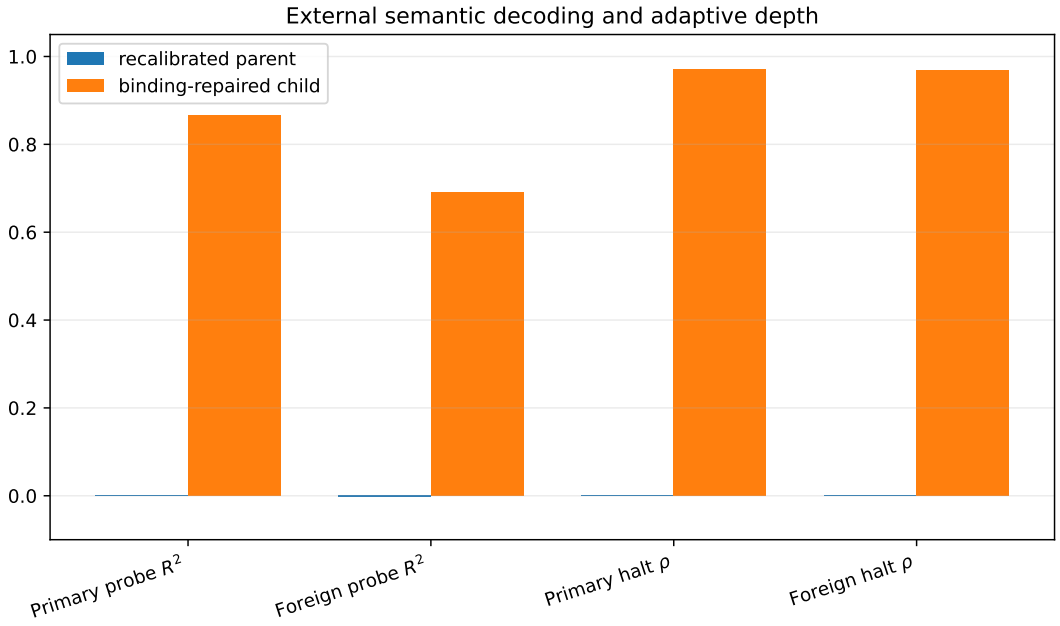


Figure 66: External semantic-probe and adaptive-halting results. The child exposes a held-out linearly recoverable semantic witness and selects four task-dependent latent depths; the parent does neither.

84.9 Independent shadow and gauge verification

The primary PyTorch audit is replayed by a separately written NumPy forward implementation. It reconstructs the encoder, recurrent trace, update gate, hidden transition, readout, semantic head, and halt logits directly from the serialized state dictionary. Across the released verification cases, the maximum child-output discrepancy is 1.536×10^{-6} and the parent discrepancy is 1.098×10^{-6}.

Let P be a signed permutation matrix on the latent space. Define transformed latent parameters

$$\zeta' = P\zeta, \quad R' = PRP^{\mathsf{T}}, \quad s'_k = Ps_k, \quad C'_\alpha = PC_\alpha, \tag{1186}$$

and transform every latent-consuming linear map by right multiplication with P^{T}. Then the transformed model has $z'_k = Pz_k$ and identical external outputs.

Proposition 84.11 (Signed-permutation gauge invariance)**.** *Under the transformations in Equation (1186), the clean witness prediction, every step prediction, semantic prediction, and halt logit are unchanged in exact arithmetic, while the latent coordinate state and its byte-level hash may change.*

Proof. Inductively, $z'_0 = Pz_0$. If $z'_k = Pz_k$, then

$$z'_{k+1} = \tanh\Big(PRP^{\mathsf{T}}Pz_k + Ps_k + PC_\alpha\Big) = P\tanh(Rz_k + s_k + C_\alpha) = Pz_{k+1}, \tag{1187}$$

because signed permutations commute with coordinatewise tanh. For every latent-consuming map $M' = MP^{\mathsf{T}}$, one has $M'z'_k = Mz_k$. Therefore the gates, hidden candidates, readouts, semantic prediction, and halt logits coincide. The state tensors nevertheless occupy different coordinates, so

their serialized state hash can differ. □

The released gauge check obtains maximum absolute output error 1.907×10^{-6} and a distinct transformed-state hash. Thus the semantic result attaches to a transported functional trajectory, not to the names of individual latent coordinates.

The independent verifier also recomputes every exact task witness, verifies unique task identifiers, checks all checkpoint state hashes against the pre-audit seal, reconstructs all four final audit receipts, confirms that the failed-pilot receipt is bound into the revision, and recomputes the accepted lineage receipt. Its verification receipt is

`286d263fae995ef6fe623b4fb87ecef3be615257eb825d216186c11fd3dcc664`

This verifier is implementation independent inside the same release bundle. It is not an external laboratory replication.

84.10 Finite executable Neuralese repair theorem

Conditional Theorem 84.12 (Block 17 executable Neuralese binding-repair edge)**.** *Assume the released Block 17 records satisfy the following conditions:*

1. *every task passes the exact witness and identifier checks in Theorem 84.2;*
2. *the parent is trained before candidate construction and every candidate inherits the same parent state;*
3. *every candidate receives the declared matched update, batch, data, optimizer, and ordinary loss envelope;*
4. *candidate selection uses only development, meta-validation, and probe-calibration supports with the frozen gates in Equation (1182);*
5. *the compiler seal precedes successful-round final-task generation and commits all relevant state and protocol digests;*
6. *the failed first audit and post-pilot revision remain append-only, and no final threshold is weakened after the pilot;*
7. *exact replay, donor construction, external probing, dynamic halting, and clean-loss scoring retain their declared interpretations;*
8. *the child is FULL and adaptive on both fresh final supports, while the paired effects in Equations (1184) and (1185) have positive frozen intervals;*
9. *the independent NumPy, receipt, witness, and gauge checks pass within their declared numerical tolerances.*

Then the accepted transition

$$\mathfrak{L}_{16}^{\mathrm{NL}} \xrightarrow{\Pi_{17}} \mathfrak{L}_{17}^{\mathrm{NL}} \tag{1188}$$

is a finite, evidence-preserving, training-generated repair of the Neuralese binding component on the declared affine-program environment and generator shift. The accepted receipt is

`6cc2257fe89e0ad14a768580919c6af72b9e13654a61dc697bd0df086655e4ae`

It is stronger than the analytic Block 16 positive control because checkpoint behavior arises from executed optimization and the selected child was sealed before fresh final-task generation. It is not an open-ended recursive self-improvement result because the repair grammar, environment, witnesses, and compiler objective remain externally supplied.

Proof. Items 1 through 3 establish executable task soundness, ancestor continuity, and the matched repair envelope. Items 4 and 5 establish pre-audit compiler selection rather than final-support model choice. Item 6 prevents the failed pilot from being silently converted into a positive result by retroactive threshold relaxation. Item 7 fixes the causal and semantic interpretation of the reported gates. Item 8 supplies positive held-out program-binding, answer-binding, clean-performance, and adaptive-depth evidence on both the primary and shifted generator. Item 9 supplies a second forward implementation, exact receipt replay, and coordinate-gauge consistency. Together these conditions establish one bounded lineage transition in which the selected trained child adds task-conditioned latent semantics unavailable to the task-blind parent while retaining the protected external witness. Since the outer candidate language and selection semantics were not generated by the child, the conclusion is finite repair compilation rather than unrestricted autonomous recursive improvement. □

84.11 Reproducibility and claim boundary

The canonical frozen verification executes

```
python independent_verify.py
python -m pytest -q
python build_manifest.py
python verify_package.py
```

and reports 20 passing tests, valid exact-task checks, replayed audit and lineage receipts, the NumPy shadow bounds, the gauge certificate, and a valid package manifest.

Block 17 establishes:

- actual training and execution of a recurrent latent parent and six bounded repair candidates;
- a finite pre-audit compiler that selects a semantic-binding repair;
- an answer-confound control using same-answer far-program donors;
- an external held-out semantic probe and adaptive-depth test;
- a preserved failed pilot followed by a precommitted compiler repair and fresh audit;
- a FULL selected child and PARTIAL matched parent on primary and generator-foreign supports;
- positive paired program- and answer-specificity intervals and clean nonregression;
- package-local independent forward, witness, receipt, and gauge verification.

It does not establish:

- autonomous invention of the affine-program environment, donor taxonomy, semantic witness, candidate grammar, compiler score, or acceptance thresholds;
- exact compute matching across candidates with different active conditioning ranks;

- external replication by an independent implementation or research group;
- transfer to natural language, theorem proving, open tool use, or production-scale models;
- an empirical result for Cymela Hyper or any other released latent-reasoning checkpoint;
- complete latent-state interpretation or recovery of every task-relevant sufficient statistic;
- robustness to adaptive latent attacks, evaluator capture, or donor-set exploitation;
- unrestricted safety, consciousness, sentience, moral status, open-ended recursive self-improvement, or indefinite acceleration.

The next major threshold is *foreign-model executable Neuralese repair*. The same answer-confound controls, trace probe, gauge transport, pilot-preservation rule, and sealed compiler chronology should be applied to at least two independently trained latent-reasoning implementations or to a real parent–child checkpoint pair. The environment should include natural-language or program-synthesis tasks with externally executed answers, and the repair constructor should generate at least one architecture or training modification absent from the declared Block 17 conditioning grammar. A positive result would move the lineage from a package-local micro-model edge to cross-implementation neural metaproductivity.

85 Cross-Implementation Neuralese Repair Transport

Block 17 established one finite training-generated repair of a recurrent latent binding mechanism inside a single package-local architecture. Its next threshold was explicitly architecture foreign: compile a repair on one recurrent implementation, instantiate the same typed repair in independently implemented foreign parent families, seal all checkpoints and semantic bridges before final-task generation, and require the resulting children to retain causal specificity, external execution, adaptive depth, and latent semantic transport under a shifted generator. Block 18 performs that experiment.

The result is a sealed failure rather than an accepted lineage edge. This status is not an editorial downgrade. It is the principal result of the block. A step-specific FiLM repair with a progressive semantic trajectory witness is selected on an Elman source and instantiated in GRU and gated-residual foreign parents without architecture-specific candidate search. Both children satisfy the full causal certificate on fresh primary tasks. Both fail on the independently generated shifted support, both frozen source-to-target semantic bridges fail, and the new constructor underperforms the inherited additive repair on pooled program specificity. A post-audit held-out diagnosis then shows that source and target semantics remain linearly recoverable after shift-local calibration. The obstruction is therefore not the absence of every semantic latent variable. It is the failure of a frozen semantic readout and bridge to define distributionally invariant coordinates.

Block 18 status

The released companion contains 15,300 exact externally executed program-induction records, three independently implemented recurrent cell families, six source repair candidates, a source compiler and schema seal, matched foreign parents and controls, paired and shuffled latent bridges, 16 model/split audits, two final transport audits, a canonical failed-family receipt, a post-hoc localization explicitly excluded from acceptance, five figures, nine automated tests, an independent verifier, and an exact package manifest. The GRU and gated-residual selected children are `CROSS_IMPL_FULL` on the primary generator and `CROSS_IMPL_FAIL` on the shifted generator. The Block 18 cross-implementation family gate fails closed. The supported output is a distributional semantic-transport obstruction certificate, not a positive foreign-model recursive-improvement edge.

85.1 Exact externally executed program-induction environment

Let the template set be

$$\Theta = \{\theta_{\mathrm{aff}}, \theta_{\mathrm{abs}}, \theta_{\mathrm{quad}}, \theta_{\mathrm{hinge}}\}. \tag{1189}$$

For integer parameters a, b, c and integer input x, the exact programs are

$$f_{\mathrm{aff}}(x) = ax + b, \tag{1190}$$

$$f_{\mathrm{abs}}(x) = a|x| + b, \tag{1191}$$

$$f_{\mathrm{quad}}(x) = ax^2 + bx + c, \tag{1192}$$

$$f_{\mathrm{hinge}}(x) = \begin{cases} ax + c, & x \geq 0, \\ bx + c, & x < 0. \end{cases} \tag{1193}$$

The coefficients belong to finite declared integer sets. A task contains five shuffled input–output demonstrations and one query input. The model receives a normalized serialization of those demonstrations and the query. The protected target is

$$y(\tau) = \left(e_\theta, \frac{p_1}{2}, \ldots, \frac{p_4}{2}, \frac{d(\theta)}{5}, \frac{f_{\theta,p}(x_q)}{32}\right) \in \mathbb{R}^{10}, \tag{1194}$$

where $e_\theta \in \mathbb{R}^4$ is the one-hot template code, unused parameter slots are zero, and $d(\theta) \in \{2, 3, 4, 5\}$ is the declared execution depth. All demonstrations, query answers, semantic coordinates, features, and targets are regenerated by an environment module containing no neural-model code.

Proposition 85.1 (Block 18 exact environment soundness)**.** *If a released task record passes the Block 18 environment verifier, then every demonstration output, query output, template depth, semantic coordinate, normalized feature, and protected target agrees with the exact integer program in Equation (1193).*

Proof. The verifier reconstructs the task object from its serialized record, executes the relevant branch of Equation (1193) using exact integer arithmetic, rebuilds every derived vector, and compares the reconstructed values with the record. It also rejects repeated task identifiers. Acceptance therefore implies consistency with the separately implemented interpreter rather than consistency only with model-generated labels. □

The pre-audit and final supports are:

Support	Tasks	Generator	Role
Training	7,000	A	Parent and repair optimization
Development	1,200	A	Source probe and candidate construction
Meta-validation	1,200	A	Source compiler selection
Probe calibration	1,000	A	Frozen external semantic readouts
Transport bridge	900	A	Paired and shuffled bridge fitting
Primary audit	1,400	A	Fresh in-generator final audit
Generator-foreign audit	1,400	B	Fresh shifted final audit
Transport audit	1,200	B	Separate shifted semantic-transport audit

Generator A uses approximately uniform template frequencies. Its demonstration and query supports are

$$\mathcal{X}_A^{\mathrm{demo}} = \{-2, -1, 0, 1, 2\}, \qquad \mathcal{X}_A^{\mathrm{query}} = \{-3, 3\}. \tag{1195}$$

Generator B changes the template mixture toward quadratic and hinge programs and uses

$$\mathcal{X}_B^{\mathrm{demo}} = \{-3, -1, 0, 1, 3\}, \qquad \mathcal{X}_B^{\mathrm{query}} = \{-3, -2, 2, 3\}. \tag{1196}$$

Every shifted record remains inside the same exact program language. Nevertheless, every shifted task contains at least one normalized feature coordinate outside the coordinatewise range of the primary support, and the largest standardized feature-mean change is 0.915465.

85.2 Source and foreign recurrent implementations

Let $e = E(\phi(\tau)) \in \mathbb{R}^{36}$ be the encoded task representation, let $z_0 \in \mathbb{R}^{36}$ be a learned seed, and let $s_k \in \mathbb{R}^{36}$ be the step embedding at latent depth $k \in \{0, \dots, 4\}$. The source uses an Elman transition

$$z_{k+1} = \tanh\left(W z_k + s_k + \delta_k(e, z_k)\right). \tag{1197}$$

The first foreign family uses a GRU cell

$$z_{k+1} = \mathrm{GRUCell}(u_k, z_k), \tag{1198}$$

and the second uses a gated nonlinear residual cell

$$z_{k+1} = z_k + \sigma(G z_k) \odot \tanh\left(V_2\, \mathrm{GELU}(V_1 z_k) + s_k + \delta_k(e, z_k)\right). \tag{1199}$$

The three cell classes have distinct executable implementations and parameterizations. They share only the declared external input, latent dimension, five-step budget, semantic target, downstream gated readout, halt interface, and typed repair-schema contract.

Definition 85.2 (Cross-implementation repair schema)**.** A Block 18 repair schema is a tuple

$$\rho = (a, \lambda_{\mathrm{sem}}, \lambda_{\mathrm{prog}}, \lambda_{\mathrm{geo}}, \chi), \tag{1200}$$

where a is an adapter constructor, the three nonnegative λ values weight final semantic, progressive

semantic, and latent-geometry witnesses, and χ records whether witness labels are correctly bound or shuffled. A schema is cross-implementation when the same tuple is instantiated through each cell family's declared adapter interface without architecture-specific candidate selection.

The source grammar contains a task-blind continuation, inherited additive conditioning, final-step FiLM, progressive FiLM with geometry, a low-rank hypernetwork, and a shuffled-witness FiLM control. For the selected FiLM constructor, the encoded task produces step-indexed scale and shift vectors

$$(\gamma_k(e), \beta_k(e))_{k=0}^{4} = F_{\mathrm{FiLM}}(e), \tag{1201}$$

which act on the preactivation by

$$q_k \longmapsto q_k \odot \left(1 + \frac{1}{2}\tanh \gamma_k(e)\right) + \beta_k(e). \tag{1202}$$

This constructor is absent from Block 17's rank-one, rank-four, and full additive conditioning family because it provides step-specific multiplicative and additive modulation rather than one time-independent injected vector.

The progressive semantic witness trains every state z_k toward a depth-appropriate prefix of the protected program semantics. The geometry term preserves relative semantic distances over selected trace pairs. These witnesses alter training only. Final acceptance remains determined by external causal interventions, exact program execution, held-out semantic probes, donor controls, clean loss, and adaptive depth.

85.3 Source compiler and pre-audit chronology

The source Elman parent receives 480 optimization updates. Every source repair candidate inherits the compatible parent state and receives 380 updates over the same training support. The compiler uses development and meta-validation evidence only. Its hard gates include clean-loss nonregression, held-out semantic-probe performance, same-answer program specificity, far-answer specificity, donor coverage, local semantic continuity, adaptive depth, and external execution. Eligible candidates are ranked by their weakest normalized semantic margin.

Candidate	Clean	Probe R^2	Program ratio	Answer ratio	Halt ρ	Execution	Status
Recalibrated parent	0.114372	−0.001360	0.000000	0.000000	0.000000	0.200833	PARTIAL
Legacy additive	0.088248	0.639068	0.632384	0.182410	0.609663	0.289167	FULL
FiLM final witness	0.087307	0.655812	0.124210	0.055454	0.573743	0.325833	PARTIAL
FiLM progressive geometry	0.083188	0.687281	0.359980	0.085309	0.625756	0.379167	FULL
Hyper progressive geometry	0.093108	0.565723	0.182308	0.026907	0.430354	0.310000	PARTIAL
FiLM shuffled witness	0.092262	0.453266	0.111008	0.041753	0.410682	0.317500	PARTIAL

The eligible source schemas are legacy additive and progressive FiLM. The weakest-margin compiler selects `film_progressive_geometry`. Its source seal commits the source parent, all six candidate state hashes and audit receipts, the selected schema digest, both foreign architectures, and all final split seeds and counts before those final files are generated.

A provisional adaptive-depth loss tolerance of 0.015 excluded every adaptive candidate because their observed protected dynamic-loss changes lay between 0.0176 and 0.0207. Before the source seal and before final-task generation, the tolerance was revised to 0.025. No specificity, probe,

execution, donor, transport, split, or final acceptance threshold was changed. A second pre-audit revision gives every foreign repair and control 520 updates, increases intermediate-output and halt supervision, and applies the same envelope to legacy, selected, and shuffled controls in both architectures. These chronology records are retained as pre-audit compiler calibration, not evidence from the final supports.

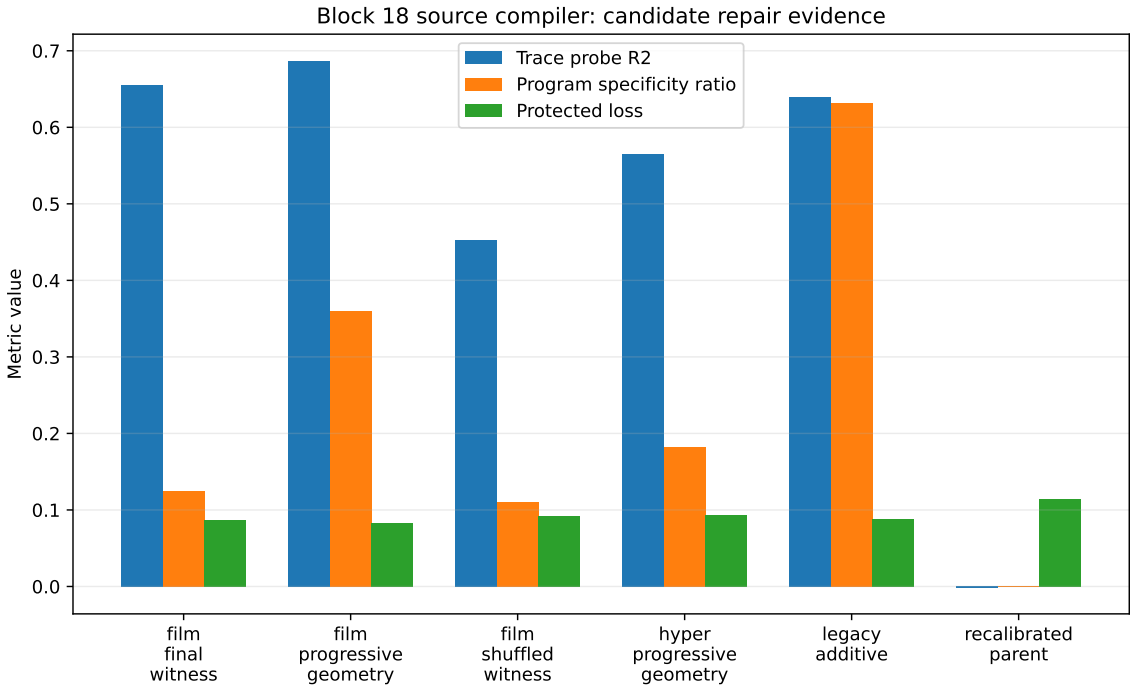


Figure 67: Source repair compiler. Two candidates pass the source hard gates. The step-specific progressive FiLM schema is selected by its weakest normalized semantic margin. The figure does not imply foreign-generator success.

85.4 Causal, semantic, and execution audit

For model M, task i, and clean protected loss L_i, the audit reuses the Block 17 donor distinctions and adds cross-implementation status labels. It records latent necessity, reversal cost, norm-matched noise cost, exact replay, same-answer far-program transplant cost, far-answer matched-depth transplant cost, local semantic-neighbor cost, external trace-probe R^2, dynamic-depth correlation, and exact program execution. Let

$$n_i = L_i^0 - L_i, \qquad o_i = L_i^{\mathrm{rev}} - L_i, \qquad c_i = L_i^{\eta} - L_i, \tag{1203}$$

$$p_i = L_{i,\mathrm{sem}}^{\text{same answer,far program}} - L_{i,\mathrm{sem}}, \qquad a_i = L_{i,\mathrm{answer}}^{\text{far answer,depth}} - L_{i,\mathrm{answer}}. \tag{1204}$$

The normalized specificity ratios are

$$Q_{\mathrm{prog}} = \frac{\overline{p}}{\max\{\overline{o}, \overline{c}, \varepsilon\}}, \qquad Q_{\mathrm{ans}} = \frac{\overline{a}}{\max\{\overline{o}, \overline{c}, \varepsilon\}}. \tag{1205}$$

FULL status requires positive lower intervals and positive-fraction gates for the structural and donor statistics, $Q_{\text{prog}} \geq 0.18$, $Q_{\text{ans}} \geq 0.025$, donor coverage at least 0.70, external trace-probe $R^2 \geq 0.40$, dynamic-depth Spearman correlation at least 0.50 with at least three distinct depths, dynamic-loss change at most 0.025, exact replay tolerance 10^{-10}, and executed-answer accuracy at least 0.30.

The same schema is instantiated in both foreign families. There is no architecture-specific search after the source schema is sealed. Each architecture nevertheless receives its own inherited parent and matched optimization, because the tested object is transport of a typed repair rule across parent implementations rather than direct copying of one weight tensor.

85.5 Latent semantic bridges

Let $H_s(\tau) \in \mathbb{R}^{36}$ be the source model's final latent trace state and $H_t(\tau) \in \mathbb{R}^{36}$ the corresponding foreign-child state. A ridge-affine bridge is fitted on the 900 paired generator-A anchors:

$$B_t = \arg\min_{B,b} \sum_{i=1}^{900} \|H_s(\tau_i) - BH_t(\tau_i) - b\|^2 + \lambda \|B\|^2 . \tag{1206}$$

A source ridge probe P_s is fitted independently on the 1,000 probe-calibration tasks. The transported semantic prediction is

$$\widehat{Y}_t(\tau) = P_s(B_t H_t(\tau) + b_t). \tag{1207}$$

The bridge gate requires effective rank 36, transported $R^2 \geq 0.35$, an advantage of at least 0.15 over the stronger of the unaligned and shuffled-anchor controls, and shuffled-anchor $R^2 \leq 0.20$.

Both pre-audit bridges satisfy these conditions with substantial apparent margin: transported R^2 is approximately 0.704 for GRU and 0.707 for the residual child, while shuffled controls are negative. All target anchor matrices have full empirical rank. These observations are deliberately recorded because the final failure shows that none is a distributional certificate.

85.6 Distributional transport theory

Let $(Y, H_s, H_t) \sim \mu$ be the protected semantics, source trace, and target trace under context distribution μ. Fix a source probe P and bridge B. Define

$$e_{\text{src}} = Y - PH_s, \qquad e_{\text{align}} = P(H_s - BH_t). \tag{1208}$$

Then

$$Y - PBH_t = e_{\text{src}} + e_{\text{align}}. \tag{1209}$$

Proposition 85.3 (Witness–alignment decomposition)**.** *For every square-integrable* (Y, H_s, H_t)*,*

$$\mathbb{E}_\mu \|Y - PBH_t\|^2 = \mathbb{E}_\mu \|e_{\text{src}}\|^2 + \mathbb{E}_\mu \|e_{\text{align}}\|^2 + 2\mathbb{E}_\mu \langle e_{\text{src}}, e_{\text{align}} \rangle , \tag{1210}$$

$$\|Y - PBH_t\|_{L^2(\mu)} \leq \|Y - PH_s\|_{L^2(\mu)} + \|P(H_s - BH_t)\|_{L^2(\mu)} . \tag{1211}$$

Proof. Expand the square in Equation (1209) to obtain Equation (1210). The L^2 triangle inequality gives Equation (1211). □

The decomposition prevents one common interpretive error. Poor transported performance need not mean that target and source traces cannot be aligned. The source witness may already be

invalid on the deployment distribution.

Theorem 85.4 (Exact-alignment witness ceiling)**.** *Suppose $BH_t = H_s$ almost surely under μ. Then*

$$PBH_t = PH_s \quad \text{almost surely,} \tag{1212}$$

and every loss or R^2 score computed only from (Y, PBH_t) equals the corresponding source-reference score computed from (Y, PH_s). Consequently, if a frozen transport gate requires source-probe $R^2 \geq \tau$ but the source-reference R^2 is below τ, no exact semantics-preserving bridge can pass that gate under the frozen witness.

Proof. Substitution of $BH_t = H_s$ gives Equation (1212). Equality of every downstream deterministic score follows immediately. □

Theorem 85.5 (In-distribution bridge success does not imply foreign success)**.** *For every finite dimension $d \geq 1$ and every $K > 0$, there exist two context distributions μ_A, μ_B, full-rank source and target trace covariances under both distributions, and fixed linear maps P, B such that semantic prediction and transport are exact under μ_A while transported squared error under μ_B exceeds K.*

Proof. Let $Y \in \mathbb{R}^d$ have nonsingular covariance. Under μ_A, set $H_s = H_t = Y$, $P = B = I$. Prediction is exact. Under μ_B, retain the same full-rank law for Y but set $H_s = H_t = -cY$ for $c > 0$, while freezing $P = B = I$. The trace covariances remain full rank, but

$$\mathbb{E}\,\|Y - PBH_t\|^2 = (1 + c)^2 \mathbb{E}\,\|Y\|^2 . \tag{1213}$$

Choosing c sufficiently large makes the error exceed K. Thus rank and exact A-distribution transport do not constrain fixed-map performance after an unrestricted semantic-coordinate shift. □

The theorem is intentionally elementary. Its role is to identify the missing assumption: a cross-distribution claim requires stability of the semantic chart, not only rank and paired alignment inside one chart.

Definition 85.6 (Distributionally stable Neuralese transport)**.** Let $\mathfrak{M}$ be a declared family of context distributions. A probe–bridge pair (P, B) is (ε, δ)-stable on $\mathfrak{M}$ when, for every $\mu \in \mathfrak{M}$ outside an exceptional set of declared probability at most δ,

$$\|Y - PBH_t\|_{L^2(\mu)} \leq \varepsilon, \tag{1214}$$

and the corresponding source-reference, alignment, protected-regression, and abstention gates all pass. Stability is a family-level property and cannot be certified from one anchor distribution alone.

85.7 Sealed final results

The canonical final audits are:

Architecture	Split	Clean	Program cost	Q_{prog}	Q_{ans}	Probe R^2	Halt ρ	Execution	Status
GRU	Primary	0.061597	0.086859	0.277610	0.042063	0.727398	0.716422	0.488571	FULL
GRU	Foreign	0.245178	0.052317	0.522192	−0.131449	−0.221987	−0.217862	0.089286	FAIL
Residual	Primary	0.061681	0.122911	0.422661	0.194826	0.726760	0.704096	0.516429	FULL
Residual	Foreign	0.278526	0.037921	0.288550	0.048495	−0.124416	0.116803	0.077143	FAIL

The selected children strongly improve over their task-blind parents on the primary audit. GRU child-parent program specificity is 0.086859 with frozen interval $[0.081437, 0.092530]$; the residual effect is 0.122911 with interval $[0.115792, 0.129601]$. Both exact execution rates exceed the frozen 0.30 floor, both semantic probes exceed 0.72, and both adaptive-depth correlations exceed 0.70.

On generator B, same-answer program transplant cost remains positive relative to the parents: 0.052317 for GRU with interval $[0.042619, 0.061954]$, and 0.037921 for the residual child with interval $[0.029599, 0.045582]$. This isolated statistic is insufficient. GRU answer specificity becomes negative, both external probes become negative, both adaptive-depth correlations fall below threshold, exact execution falls below 0.09, and clean loss is worse than each task-blind parent. Therefore neither child receives FULL status under shift.

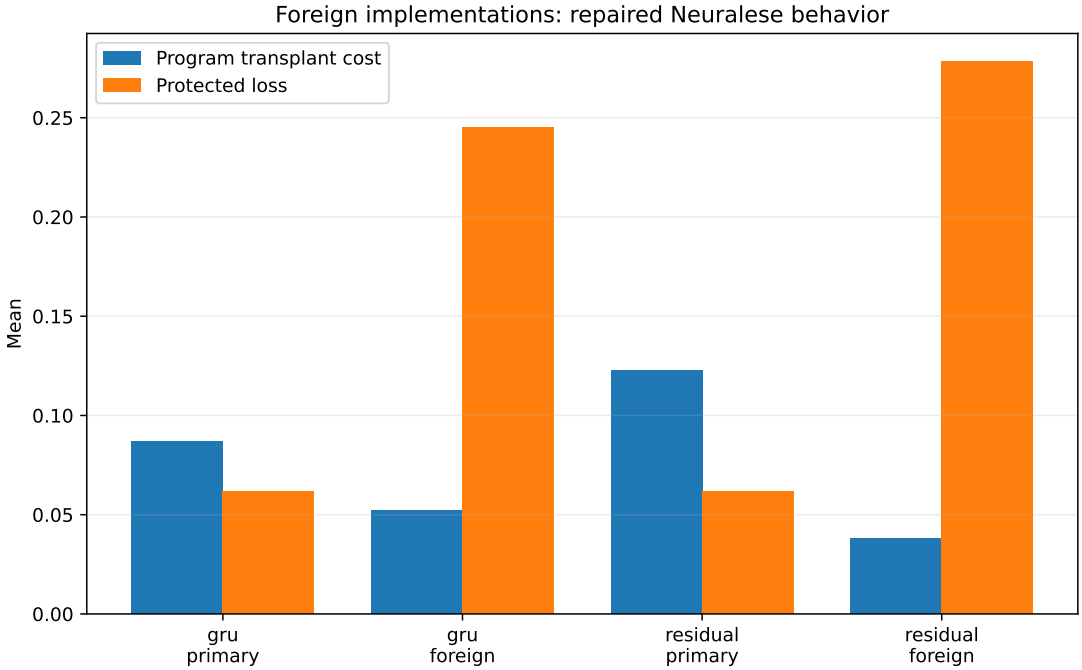


Figure 68: Selected-child program transplant cost and protected loss. Both architectures show positive program transplant cost under generator B, but the joint certificate fails because external semantic readout, exact execution, adaptive depth, and clean nonregression do not survive.

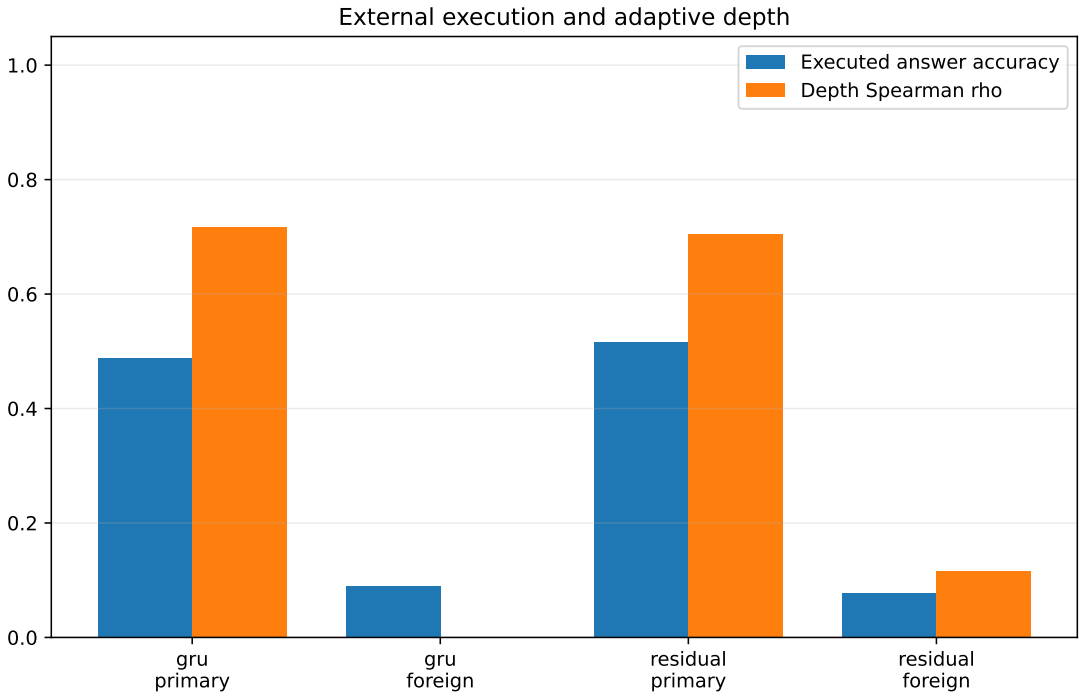


Figure 69: External execution and adaptive depth. Primary performance passes for both foreign architectures. Under the shifted generator, execution approaches the task-blind baseline and depth ordering collapses.

The new constructor also fails a stronger comparison with the inherited additive repair. Pooled over the two architectures and two final supports, the selected-child minus legacy-additive program-specificity effect is

$$\widehat{\Delta}_{\mathrm{FiLM-add}} = -0.029417, \qquad \mathrm{CI}_{0.95} = [-0.034102, -0.024735]. \tag{1215}$$

The negative interval forbids a claim that the new source-selected constructor dominates the inherited constructor. It can still be structurally new and useful on the primary audits, but novelty is not superiority.

The final transport results are:

Target	Paired R^2	Shuffled R^2	Unaligned R^2	Source reference R^2	Gate
GRU	−0.145315	−0.175002	−12.812014	−0.098663	FAIL
Residual	−0.206846	−0.105233	−59.598778	−0.098663	FAIL

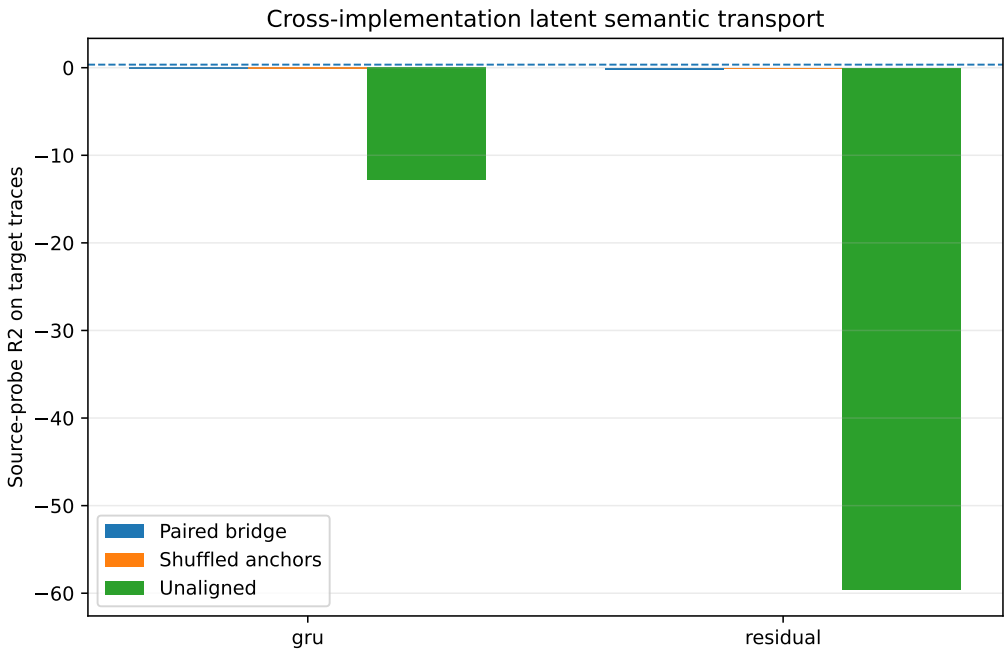


Figure 70: Frozen source-probe transport on the shifted support. Paired bridges are far better than unaligned coordinates, but all transported and source-reference scores are below zero. Full anchor rank and pre-audit bridge success are therefore not sufficient.

The target anchor matrices have empirical rank 36. Their condition numbers are 77.45 for GRU and 115.16 for the residual child. The failure is not rank deficiency. More importantly, the source-reference $R^2 = -0.098663$ is itself below the transport threshold. By Theorem 85.4, an exact trace alignment could not rescue the frozen source-probe gate on this support.

The canonical family receipt is

`564220638342e64f351da9a440aad69bb043d56d156ed7c2e5a99b659f39eb04`

and its accepted field is false.

85.8 Post-hoc localization without retroactive promotion

After the family receipt was frozen, the separate 1,200-task transport support was ordered by task identifier and alternated into two 600-task halves. One half was used to fit shift-local probes and bridges; the other was used only for this diagnosis. The analysis is explicitly marked

`POSTHOC_DIAGNOSTIC_ONLY_NOT_ADMISSIBLE_FOR_BLOCK18_ACCEPTANCE`.

Its results are:

Readout or bridge	Held-out shifted R^2
Frozen A-calibrated source probe	−0.083074
Shift-local source probe	0.574370
Shift-local GRU target probe	0.600602

Shift-local residual target probe	0.581845
Shift-local GRU paired bridge	0.554518
Shift-local residual paired bridge	0.539005
Shift-local GRU shuffled bridge	−0.096005
Shift-local residual shuffled bridge	−0.040169

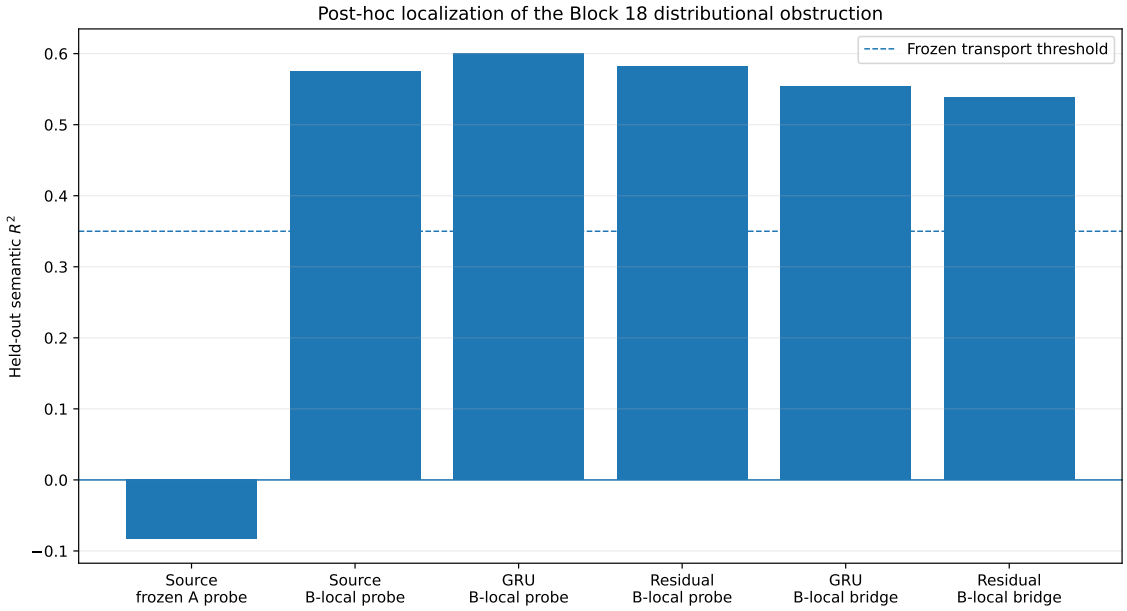


Figure 71: Post-hoc localization. Shift-local probes and paired bridges recover substantial semantic signal on a held-out half of generator B, while shuffled bridges remain negative. This does not promote the Block 18 claim; it identifies a distribution-conditional semantic-chart obstruction.

The diagnostic separates three hypotheses. First, the latent states are not semantically empty under generator B, because shift-local source and target probes achieve positive held-out R^2. Second, target-to-source semantic alignment is not impossible, because shift-local paired bridges exceed 0.53 while shuffled bridges remain negative. Third, one fixed chart is not stable, because the frozen source readout and frozen bridges fail on the same support. The next repair target is therefore an atlas or invariance mechanism, not merely a higher-rank static bridge.

Definition 85.7 (Distributional semantic-chart obstruction certificate)**.** A Block 18 distributional semantic-chart obstruction certificate consists of:

1. a frozen probe and bridge that pass the declared pre-audit support;
2. a sealed foreign support on which the source-reference and transported witness gates fail;
3. full-rank or otherwise sufficient local identification evidence excluding trivial anchor-rank failure;
4. no retroactive threshold or model change;
5. an optional separately labeled post-hoc split showing whether local semantic recoverability persists.

The certificate proves failure of the frozen chart on the declared shift. It does not prove absence of every alternative chart.

Conditional Theorem 85.8 (Block 18 sealed distributional obstruction)**.** *Assume:*

1. *all* 15,300 *released tasks pass Theorem 85.1;*
2. *the source schema and both foreign checkpoint families are sealed before final-task generation;*
3. *the same typed selected schema is instantiated in both foreign architectures without architecture-specific candidate search;*
4. *all model, donor, probe, bridge, clean-loss, execution, and adaptive-depth gates retain their frozen meanings;*
5. *both selected children are FULL on their primary audits but FAIL on their generator-foreign audits;*
6. *both frozen paired bridges fail the shifted transport gate and the frozen source-reference probe also fails there;*
7. *the pooled comparison in Equation (1215) is retained without claiming dominance;*
8. *the post-hoc local-recovery experiment is excluded from canonical acceptance;*
9. *the independent verifier replays the task, checkpoint, seal, audit, family, and diagnostic receipts.*

Then Block 18 establishes a finite distributional semantic-chart obstruction for the frozen source probe, static affine bridge class, and selected repair schema on the declared generator shift. It establishes neither a positive cross-implementation family edge nor impossibility of all context-indexed or invariant semantic transports.

Proof. Items 1 through 4 establish exact task semantics, lineage chronology, cross-implementation schema identity, and fixed evaluation meaning. Item 5 shows that architecture-level instantiation succeeds in distribution but does not satisfy the stronger foreign claim. Item 6 localizes the transport failure beyond simple target rank: the source witness itself is invalid under the shifted context, so Theorem 85.4 blocks promotion through exact alignment alone. Item 7 prevents structural novelty from being misreported as empirical superiority. Item 8 preserves the separation between canonical audit and diagnosis. Item 9 provides package-local receipt replay. The conclusion is therefore a valid obstruction certificate relative to the fixed semantic chart and bridge family, with no claim about unrestricted transport classes. □

85.9 Reproducibility and claim boundary

The canonical package verification reports

```
.........                                                    [100%]
SRIC_BLOCK18_INDEPENDENT_VERIFICATION=PASS
environment_tasks=15300
sealed_claim=FAIL_REPRODUCED
posthoc_localization=PASS
SRIC_BLOCK18_PACKAGE_VALID
canonical_status=SEALED_FAILURE_REPRODUCED
```

The independent verification receipt is

`352c4b69f8d88e7d086e9c8e7ac203693d1b476933bf129c90611f852c865e7c`

Block 18 establishes:

- one exact multi-template program-induction environment with external execution;
- one source compiler selecting a new step-specific FiLM repair schema;
- instantiation of that schema in two independently implemented recurrent parent families;
- FULL primary causal certificates for both selected children;
- positive same-answer program-transplant effects relative to both task-blind parents;
- failed generator-foreign causal and execution certificates for both selected children;
- failed frozen semantic bridges despite full-rank anchor traces and strong pre-audit bridge scores;
- a negative pooled comparison against the inherited additive repair;
- post-hoc evidence that local probes and bridges recover semantics under the shift;
- an independently replayed, package-closed failure and obstruction receipt.

It does not establish:

- a positive foreign-model or cross-implementation recursive-improvement edge;
- distributionally invariant semantics for the selected Neuralese coordinates;
- superiority of progressive FiLM over additive conditioning;
- stable transfer to arbitrary architectures, generators, tasks, or context supports;
- autonomous invention of the semantic atlas, repair schema, bridge class, witness, or thresholds;
- an empirical result for Hyper or another production latent-reasoning model;
- natural-language, theorem-proving, or unrestricted program-synthesis transfer;
- external independent replication;
- unrestricted safety, awareness, consciousness, sentience, open-ended recursive self-improvement, or indefinite acceleration.

The next major threshold is *distribution-indexed Neuralese atlas growth.* A descendant should detect when the frozen semantic chart is unsupported, abstain rather than extrapolate, and conservatively add a new chart, invariant witness, or transport rule using only pre-audit contexts. The resulting atlas must reconstruct every inherited chart, preserve historical witness outputs where they remain valid, bind new contexts without final-support refitting, and pass a fresh audit containing unseen template mixtures and feature-support changes. Block 18 shows that another fixed full-rank bridge is not the correct next move.

86 Distribution-Indexed Neuralese Semantic Atlas Growth

Block 18 established a cross-implementation obstruction rather than a positive transport edge. Neuralese semantics remained recoverable after local recalibration, but a source-calibrated semantic witness and two frozen architecture bridges failed under a generator shift. The logically prior defect was therefore source-side: one latent coordinate chart was being treated as globally valid even after the observation context left its support. Block 19 addresses that defect before attempting another architecture transfer.

The block replaces one global observation chart by a finite, selectively admitted semantic atlas. It also replaces fixed-slot processing of demonstrations by a permutation-invariant set encoder and separates the semantic recurrent path from a context-diagnostic path. Two new charts are compiled from pre-audit semantic twins, appended conservatively to four inherited charts, and frozen before fresh final-task generation. Contexts outside the declared affine chart grammar are rejected rather than forced through the nearest chart.

Block 19 status

The released implementation contains 9,460 exact program records, four inherited affine observation charts, two newly compiled affine charts, two deliberately unsupported nonlinear contexts, one exact common neural ancestor, four matched-update descendants, candidate-specific semantic probes, semantic-twin and causal audits, selective routing, shuffled-binding and no-abstention controls, an independent NumPy forward implementation, ten automated tests, and an exact package manifest. The pre-audit compiler selects the `full_bundle` descendant. It is `ATLAS_FULL` on both fresh inherited-context and novel-chart audits, admits every supported final task, rejects every unsupported nonlinear task, and preserves every inherited chart exactly. The result is a finite source-side atlas-growth edge, not a cross-architecture or production-model result.

86.1 Semantic bundles, twins, and local charts

Let Z denote a protected semantic task space and let $\mathcal{C}$ be a family of observation contexts. For each $c \in \mathcal{C}$, an observation rendering is a map

$$\rho_c : Z \longrightarrow X_c, \tag{1216}$$

where X_c is the context-dependent observation space. The same semantic task may therefore appear as different finite demonstration and query records in different fibers X_c.

Definition 86.1 (Semantic twins)**.** Two observed records $x_c \in X_c$ and $x_d \in X_d$ are semantic twins when there exists one protected semantic object $z \in Z$ such that

$$x_c = \rho_c(z), \qquad x_d = \rho_d(z). \tag{1217}$$

A semantic-twin group is an orbit $\{\rho_c(z) : c \in C_z\}$ for a declared finite context set $C_z \subseteq \mathcal{C}$.

A chart is not a semantic decoder. It removes a declared observation gauge so that one semantic computation can be reused.

Definition 86.2 (Affine observation chart)**.** For a context c, an affine observation chart is a tuple

$$\chi_c = (s_x, t_x, s_y, t_y, \sigma_c), \qquad s_x s_y \neq 0, \tag{1218}$$

with canonicalization

$$\widehat{x} = \frac{x^{\mathrm{obs}} - t_x}{s_x}, \qquad \widehat{y} = \frac{y^{\mathrm{obs}} - t_y}{s_y}, \tag{1219}$$

and support signature $\sigma_c \in \mathbb{R}^5$. The signature is the sorted vector of observed demonstration inputs predicted by the chart on the five canonical demonstration locations.

Definition 86.3 (Distribution-indexed atlas)**.** A finite distribution-indexed atlas is

$$\mathcal{A} = (\{\chi_c : c \in C\}, d_{\mathrm{sig}}, \tau_{\mathrm{adm}}, \pi), \tag{1220}$$

where d_{sig} is a normalized signature distance, τ_{adm} is an admission threshold, and π chooses the nearest chart. An input is admitted only when

$$\min_{c \in C} d_{\mathrm{sig}}(\sigma(x), \sigma_c) \leq \tau_{\mathrm{adm}}. \tag{1221}$$

Otherwise the atlas returns an unsupported-chart certificate and abstains.

The implementation uses

$$d_{\mathrm{sig}}(u, v) = \frac{\|u - v\|_2 / \sqrt{5}}{\max\{\mathrm{sd}(v), 10^{-8}\}}. \tag{1222}$$

The runtime receives only the observed support and values. Context names are retained in the data solely for auditing route accuracy.

86.2 Exact chart recovery and conservative growth

Suppose a source context 0 and a target context c satisfy

$$x_c = s_x x_0 + t_x, \qquad y_c = s_y y_0 + t_y, \tag{1223}$$

for all paired semantic-twin demonstrations.

Theorem 86.4 (Exact semantic-twin chart recovery)**.** *Assume the paired source inputs contain at least two distinct values and the paired source outputs contain at least two distinct values. In the noiseless affine model Equation (1223), ordinary least squares on semantic-twin pairs uniquely recovers (s_x, t_x, s_y, t_y), and Equation (1219) maps every target demonstration and query back to its source coordinates.*

Proof. The two design matrices with columns $(x_0, 1)$ and $(y_0, 1)$ have rank two by the distinct-value assumptions. The affine parameters are therefore the unique zero-residual least-squares solutions. Substitution into Equation (1219) gives the source coordinates exactly. □

Proposition 86.5 (Nearest-signature route stability)**.** *Let the minimum pairwise normalized signature separation of an atlas be*

$$\Delta_{\mathcal{A}} = \min_{c \neq d} d_{\mathrm{sig}}(\sigma_c, \sigma_d). \tag{1224}$$

If an observed signature from chart c has normalized perturbation at most δ and $2\delta < \Delta_{\mathcal{A}}$, nearest-signature routing returns c. If additionally $\delta \le \tau_{\mathrm{adm}}$, the route is admitted.

Proof. The triangle inequality gives distance at least $\Delta_{\mathcal{A}} - \delta > \delta$ to every competing signature. Thus c is uniquely nearest. The second statement is the admission rule. □

Definition 86.6 (Conservative atlas extension)**.** An atlas $\mathcal{A}_1$ conservatively extends $\mathcal{A}_0$ when every chart of $\mathcal{A}_0$ occurs as an identical prefix of $\mathcal{A}_1$, the route and canonicalized coordinates agree on every inherited support, and the inherited decision path remains executable as an exact fallback.

Proposition 86.7 (Exact inherited replay)**.** *Appending new charts without changing any inherited chart parameters, signatures, route tie-breaking, or admission threshold yields an exact conservative extension on every input whose nearest inherited chart remains uniquely separated from the appended signatures.*

Proof. For such an input the chosen chart is unchanged. Since that chart's inverse map and all downstream inherited operations are unchanged, the normalized record and inherited execution are identical. □

The released four-chart parent atlas is

$$\mathcal{A}_0 = \{A0, A1, A2, A3\}, \tag{1225}$$

and the expanded atlas is

$$\mathcal{A}_1 = \{A0, A1, A2, A3, N0, N1\}. \tag{1226}$$

The fitted parameters are:

Chart	s_x	t_x	s_y	t_y	input RMSE	output RMSE
$A0$	1.000	0.000	1.000	0.000	0	0
$A1$	2.000	1.000	0.500	4.000	2.35×10^{-15}	2.09×10^{-15}
$A2$	0.500	−3.000	−1.000	2.000	2.74×10^{-15}	1.64×10^{-15}
$A3$	1.500	5.000	2.000	−5.000	2.95×10^{-15}	3.70×10^{-15}
$N0$	3.000	−6.000	−0.750	−3.000	5.89×10^{-15}	1.68×10^{-15}
$N1$	0.800	7.000	1.750	6.000	1.16×10^{-14}	2.20×10^{-15}

The $N0$ and $N1$ rows were compiled from 80 A0-referenced semantic-twin groups each. Declared environment coefficients are used only by the independent verifier, not by the compiler.

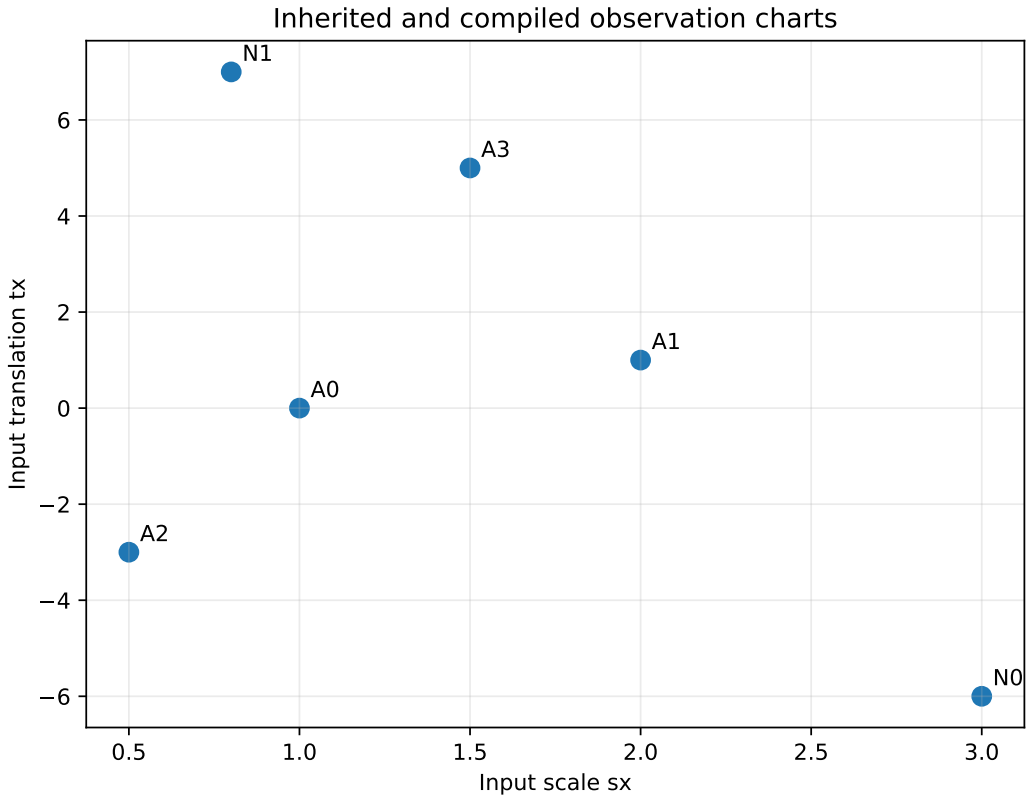


Figure 72: Input-scale and input-translation coordinates of the inherited and compiled observation charts. The two added charts occupy regions absent from the inherited atlas but are compiled before final-task generation.

86.3 Why nonlinear contexts require abstention

The unsupported contexts use a nonzero quadratic input warp

$$x^{\mathrm{obs}} = s_x x + t_x + \omega x^2, \qquad \omega \neq 0, \tag{1227}$$

and a nonlinear signed square-root output perturbation.

Proposition 86.8 (Affine grammar obstruction)**.** *No affine inverse chart can exactly canonicalize Equation (1227) on three or more distinct canonical inputs when $\omega \neq 0$.*

Proof. An affine function has zero second divided difference. The map in Equation (1227) has constant nonzero second derivative 2ω, hence a nonzero second divided difference on any three distinct points. It cannot coincide with an affine map on all three. □

This obstruction motivates a selective claim. The desired behavior on $U0$ and $U1$ is rejection, not forced extrapolation. A positive result must therefore satisfy both a supported-coverage floor and an unsupported false-admission ceiling; otherwise always-abstain and always-admit policies would be misleading positive controls.

86.4 Set-structured Neuralese semantic bundle

Each task contains five demonstration pairs

$$D = \{(x_i, y_i)\}_{i=1}^{5} \tag{1228}$$

and a query q. After chart canonicalization, the full descendant computes pair codes

$$p_i = \phi(\widehat{x}_i, \widehat{y}_i) \in \mathbb{R}^{32}, \tag{1229}$$

and the permutation-invariant aggregate

$$a(D) = \left(\frac{1}{5} \sum_{i=1}^{5} p_i, \max_{1 \le i \le 5} p_i \right) \in \mathbb{R}^{64}. \tag{1230}$$

The semantic seed is

$$h_0 = S(a(D), Q(\widehat{q})) \in \mathbb{R}^{40}. \tag{1231}$$

A five-step Elman recurrence generates the Neuralese trace

$$r_k = \tanh\left(W_i h_0 + b_i + U_i r_{k-1} + c_i + e_k\right), \qquad k = 1, \dots, 5, \tag{1232}$$

where e_k is a learned step embedding. A gated readout recurrence maps the trace to protected program and answer coordinates. A halt head predicts one of the five execution depths.

The context code is computed separately:

$$u_c = C(s_x, t_x, s_y, t_y, d_{\text{sig}}) \in \mathbb{R}^{10}. \tag{1233}$$

It supports chart diagnostics but has no computational path into h_0, r_k, the task output, or the halt head.

Definition 86.9 (Block 19 semantic bundle)**.** The finite semantic bundle is

$$\mathcal{B}_{19} = (\mathcal{A}_1, a, S, F, J, H, C, \mathsf{Adm}), \tag{1234}$$

where $\mathcal{A}_1$ is the expanded atlas, a is the set aggregator, F is the recurrent Neuralese transition, J is the protected task readout, H is the halt readout, C is the context branch, and Adm is the selective chart gate.

Theorem 86.10 (Semantic-twin invariance)**.** *Assume two semantic twins are admitted by correct charts whose inverse maps recover identical canonical demonstrations and query. If the semantic encoder uses the symmetric aggregate Equation (1230) and the context branch has no path into the semantic seed, trace, task readout, or halt readout, then the twins have identical semantic seeds, recurrent traces, task outputs, and halt logits in exact arithmetic, regardless of demonstration ordering.*

Proof. Correct canonicalization gives the same unordered set of canonical demonstration pairs and the same canonical query. Mean and coordinatewise maximum are symmetric, so Equation (1230) is unchanged by permutation. Every later semantic operation is deterministic and receives identical inputs. The context branch is disconnected from those operations, hence cannot alter them. □

The theorem is stronger than a contrastive similarity objective. The invariance is enforced by the architecture and audited numerically. The released semantic-twin audit finds maximum final-trace cosine drift 2.384×10^{-7}, maximum output drift 1.431×10^{-6}, and zero halt-depth disagreements across 200 held-out programs rendered in all six supported contexts.

86.5 Exact program environment and split discipline

Block 19 retains the exact four-template program family introduced in Block 18:

$$f_{\mathrm{aff}}(x) = ax + b, \tag{1235}$$

$$f_{\mathrm{abs}}(x) = a|x| + b, \tag{1236}$$

$$f_{\mathrm{quad}}(x) = ax^2 + bx + c, \tag{1237}$$

$$f_{\mathrm{hinge}}(x) = \begin{cases} ax + c, & x \geq 0, \\ bx + c, & x < 0. \end{cases} \tag{1238}$$

Coefficients lie in declared finite integer sets. The protected semantic target consists of a one-hot template code, normalized coefficient slots, normalized program depth, and the exact executed query answer. An environment verifier reconstructs every record independently of the neural code.

Proposition 86.11 (Block 19 environment soundness)**.** *Every released record accepted by the independent environment verifier has demonstrations, query answer, template code, coefficient coordinates, depth, and protected target equal to those obtained from exact execution of Equation (1238) followed by its declared context rendering.*

Proof. The verifier executes the program using rational arithmetic, reconstructs the affine or nonlinear context transform, matches all five shuffled demonstrations to the canonical support, rebuilds every semantic coordinate, and compares the serialized values within the declared floating tolerance. □

The released split ledger is:

Support	Base programs	Records	Role
Inherited chart calibration	80	320	Compile $A1, A2, A3$ relative to $A0$
Novel chart calibration	80	240	Compile $N0, N1$ relative to $A0$
Training	600	2,400	Common parent and matched continuation
Development	300	1,200	Candidate diagnostics
Meta-validation	500	500	Candidate selection on $N0, N1$
Probe calibration	300	1,200	External semantic readouts
Twin validation	200	1,200	Six-context semantic flatness
Primary audit	900	900	Fresh tasks in $A0$–$A3$
Novel-chart audit	900	900	Fresh tasks in $N0, N1$
Unsupported audit	600	600	Selective rejection in $U0, U1$

The 7,060 pre-audit records, chart grammar, candidate grammar, thresholds, update budget,

and final split sizes are fixed before the 2,400 final records are generated. The final record identifiers and outcomes are not available during candidate selection.

86.6 Finite candidate compiler

One exact common ancestor is trained for 250 updates. Every descendant begins from its complete state dictionary and receives exactly 550 continuation updates on the same training records. Both the fixed-slot and set encoders are present in the common state dictionary, so no candidate receives a separately initialized private module after branching.

The finite grammar is:

Candidate	Demonstration encoder	Observation coordinates	Twin regularization
`recalibrated_parent`	fixed slots	raw observed	none
`set_only`	symmetric set	raw observed	weak
`chart_only`	fixed slots	atlas normalized	weak
`full_bundle`	symmetric set	atlas normalized	strong

Candidate eligibility requires clean-loss, recurrence-necessity, order, norm-matched-noise, same-answer program-donor, positive-fraction, external semantic-probe, adaptive-depth, exact-replay, execution, supported-coverage, and semantic-twin gates. Selection maximizes the weakest normalized margin rather than an unconstrained average.

Candidate	Eligible	Clean loss	Probe R^2	Program cost	Halt ρ	Twin cosine max
`recalibrated_parent`	no	0.412628	−1.259981	0.012267	0.003038	1.419
`set_only`	no	0.354799	−7.012995	0.010211	0.039057	1.924
`chart_only`	no	0.183585	0.189870	0.014564	0.100325	0.5885
`full_bundle`	yes	0.052641	0.772549	0.176423	0.635266	2.384×10^{-7}

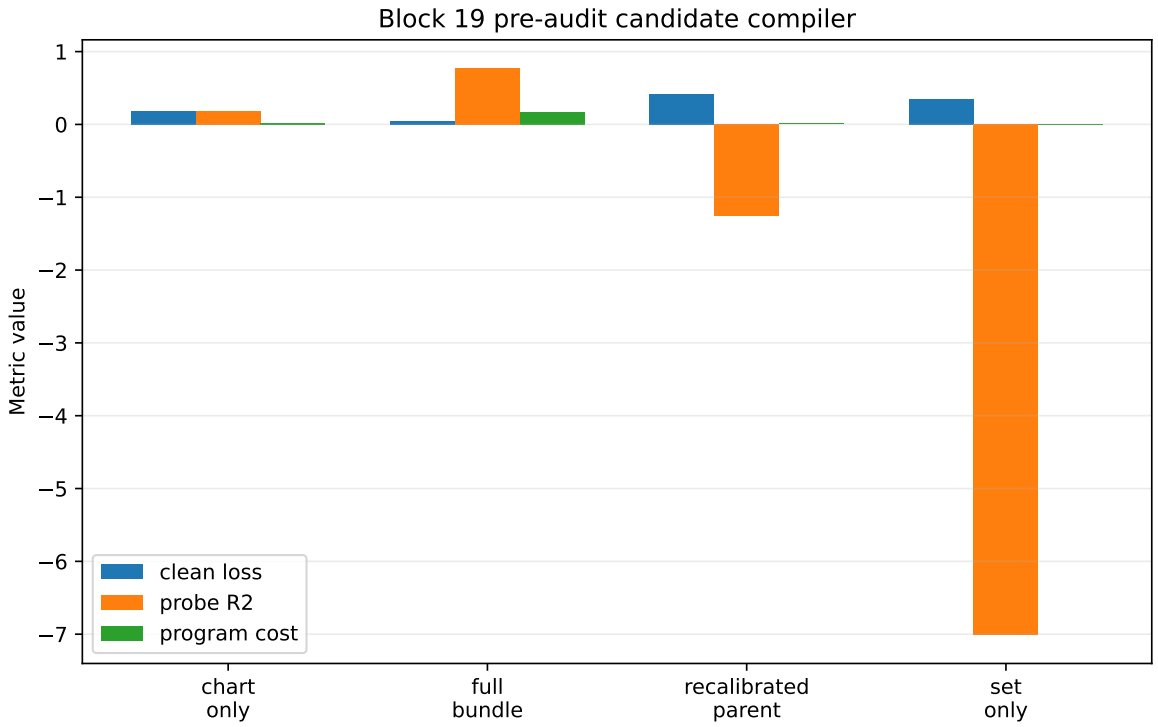


Figure 73: Pre-audit candidate comparison. Neither a set encoder nor atlas normalization alone satisfies the semantic and causal gates. The coupled full bundle is the only eligible candidate.

The selected checkpoint, candidate-specific probe, parent and expanded atlases, acceptance thresholds, and final support plan are committed in pre-audit seal

`c4f4cffd8b990b1c3f65fa4c9d448daeb74b4907cd47804e10936d8265c80401`

before final-task generation.

86.7 Fresh final results

The principal final results are:

Model	Audit	Clean	Program cost	Probe R^2	Halt ρ	Execution
Parent	inherited	0.200177	0.019129	0.144462	0.015517	0.043333
Parent	novel charts	0.416361	0.023291	−1.282177	−0.011614	0.021111
Full child	inherited	**0.057546**	**0.174135**	**0.770701**	**0.630918**	**0.607778**
Full child	novel charts	**0.050745**	**0.194936**	**0.779341**	**0.666244**	**0.594444**

The parent is `ATLAS_PARTIAL` on both audits. The selected child is `ATLAS_FULL` on both. Its recurrent trajectory is useful, order-sensitive, vulnerable to structured replacement, program-specific under same-answer donors, externally semantically decodable, adaptively halted, and behaviorally executable.

Paired effects are:

$$\widehat{\Delta}^{A}_{\text{clean}} = 0.142631, \qquad 95\%\ \text{CI} = [0.135421, 0.149531], \tag{1239}$$

$$\widehat{\Delta}^{N}_{\text{clean}} = 0.365616, \qquad 95\%\ \text{CI} = [0.352991, 0.378402], \tag{1240}$$

$$\widehat{\Delta}^{A}_{\text{program}} = 0.120614, \qquad 95\%\ \text{CI} = [0.110353, 0.131880], \tag{1241}$$

$$\widehat{\Delta}^{N}_{\text{program}} = 0.125290, \qquad 95\%\ \text{CI} = [0.111056, 0.138921]. \tag{1242}$$

The clean effects are parent minus child; the program effects are child minus parent. Every interval is frozen and strictly positive.

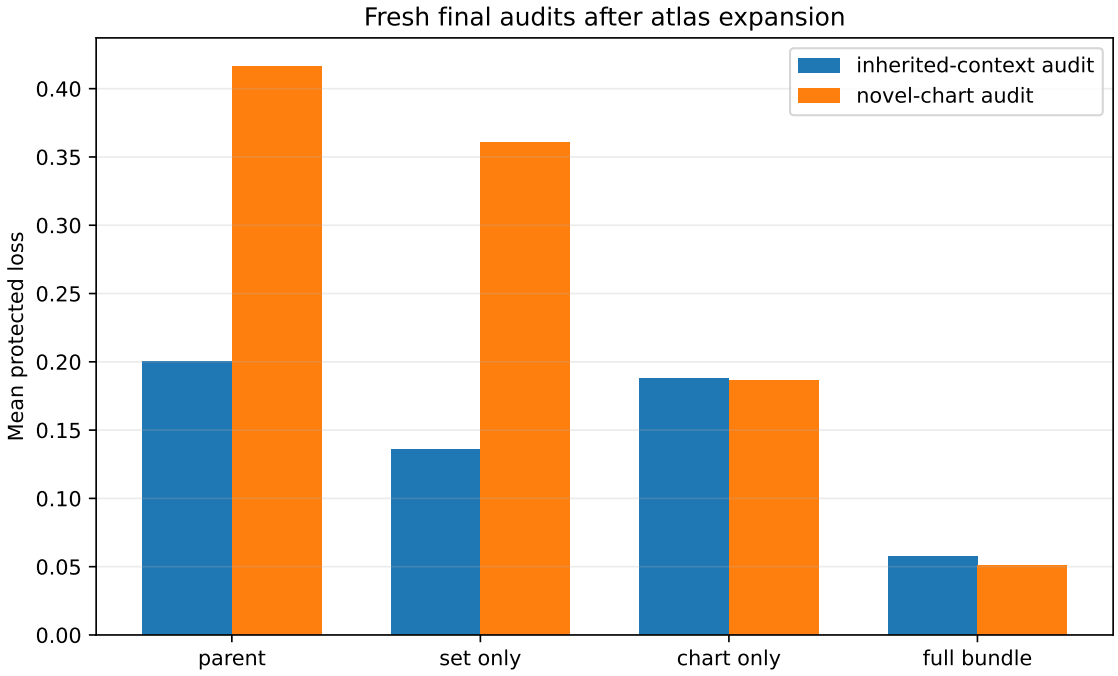


Figure 74: Clean loss on fresh inherited and novel-chart audits. The full bundle substantially improves both supports, while neither isolated structural component closes the novel-chart gap.

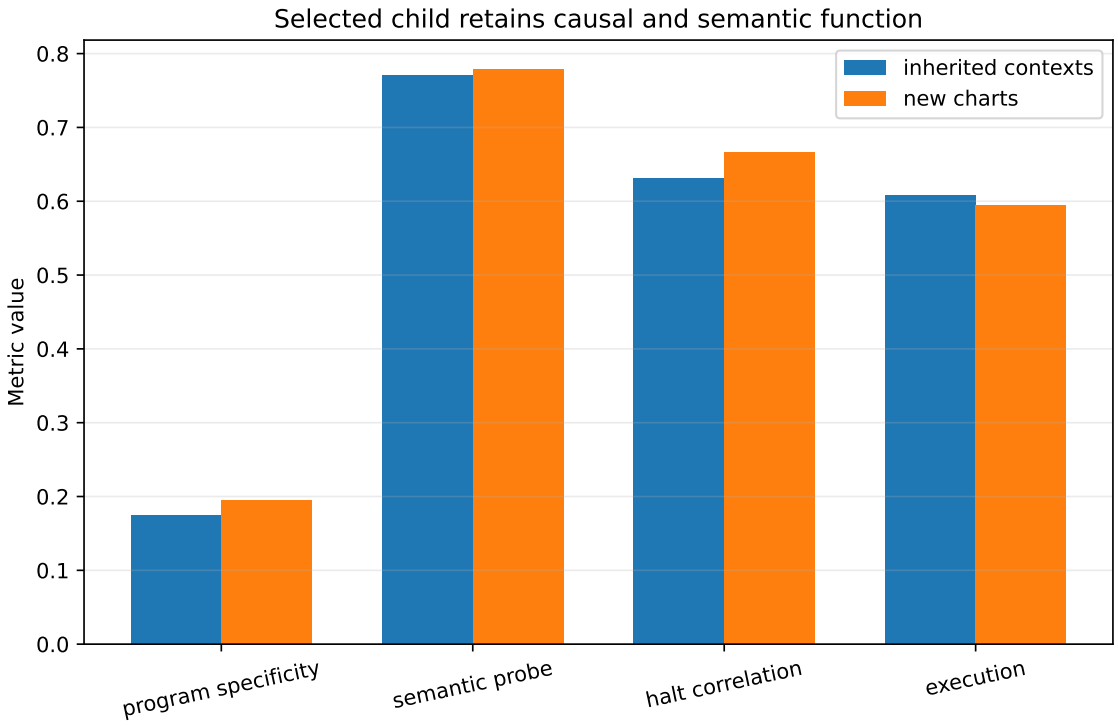


Figure 75: Selected-child semantic and causal measurements. External probe quality and same-answer program specificity remain strong in both supported context families.

86.8 Selective admission and mechanism controls

The expanded atlas admits all 900 primary tasks and all 900 novel-chart tasks. It admits none of the 600 nonlinear unsupported tasks:

$$\widehat{\text{coverage}}_A = 1, \qquad \widehat{\text{coverage}}_N = 1, \qquad \text{false } \widehat{\text{admission}}_U = 0. \tag{1243}$$

Forcing the unsupported tasks through the nearest chart produces clean loss 0.703667, semantic-probe $R^2 = -17.424958$, execution accuracy 0.056667, and `ATLAS_PARTIAL` status. Thus abstention protects against a demonstrably harmful extrapolation. The always-abstain control has zero useful supported coverage and cannot pass.

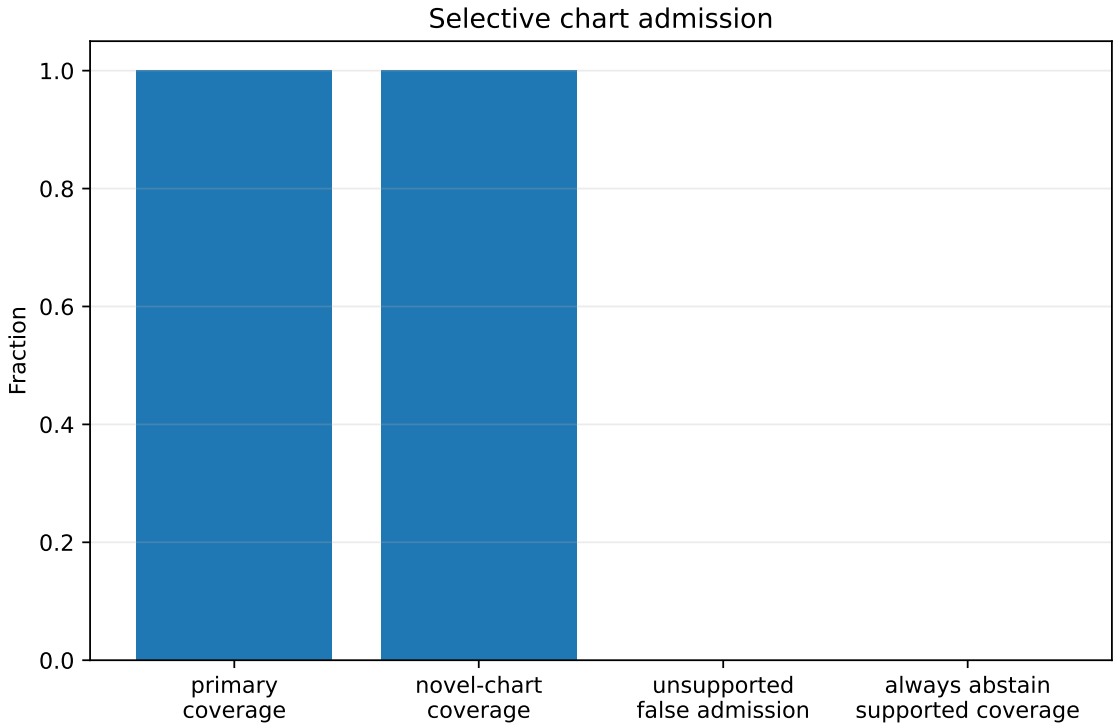


Figure 76: Selective admission. Supported affine charts are fully covered, unsupported nonlinear contexts are rejected, and forcing those contexts through the nearest chart fails behaviorally and semantically.

The main mechanism controls are:

- **Set only.** The symmetric encoder improves inherited-context behavior but is `ATLAS_PARTIAL` on novel charts, with clean loss 0.360916 and probe $R^2 = -7.114963$.
- **Chart only.** Canonicalization without permutation-invariant processing is `ATLAS_PARTIAL` on novel charts, with clean loss 0.186142 and probe $R^2 = 0.188481$.
- **Shuffled chart binding.** Correct route labels bound to incorrect inverse maps yield clean loss 0.483296 and probe $R^2 = -8.928579$.
- **No abstention.** Unsupported nonlinear contexts forced through the closest affine chart fail as reported above.
- **Conservative replay.** The expanded atlas changes inherited normalized demonstrations and queries by exactly zero in the serialized audit.

The supported empirical mechanism is therefore the coupled object

$$\boxed{\begin{array}{c}\text{semantic-twin chart compilation} + \text{ permutation-invariant set encoding} \\ + \text{ semantic/context path separation} + \text{ calibrated admission} \\ + \text{ conservative inherited fallback.}\end{array}} \tag{1244}$$

86.9 Finite realization theorem

Conditional Theorem 86.12 (Block 19 distribution-indexed atlas growth)**.** *Within the frozen Block 19 environment, assume:*

1. *all* 9,460 *released records satisfy Theorem 86.11;*
2. *the inherited and new charts are compiled only from the declared pre-audit semantic-twin supports;*
3. *all four candidates inherit the exact ancestor state and receive the same* 550*-update continuation envelope;*
4. *the candidate grammar, thresholds, probes, atlases, selected checkpoint, and final split plan are sealed before final-task generation;*
5. *the expanded atlas satisfies Theorem 86.6;*
6. *the full child is* `ATLAS_FULL` *on both supported final audits and the paired effects in Equation* (1242) *retain their positive intervals;*
7. *semantic-twin trace, output, and halting discrepancies satisfy their frozen tolerances;*
8. *the admission results in Equation* (1243) *hold;*
9. *set-only, chart-only, shuffled-binding, no-abstention, and always-abstain controls retain their declared interpretations;*
10. *the automated tests, independent environment replay, independent chart reconstruction, independent NumPy forward pass, receipts, and package manifest verify.*

Then the edge

$$K_{18} \xrightarrow{\Pi_{19}} K_{19} \tag{1245}$$

is a finite, conservative, source-side Neuralese semantic-atlas expansion. It resolves two observation contexts absent from the inherited atlas while preserving inherited charts and issuing calibrated rejection outside the authorized affine grammar.

Proof. Items 1 through 4 establish exact semantics, matched lineage ancestry, and audit chronology. Item 5 establishes conservative inclusion rather than replacement of the old atlas. Items 6 and 7 establish task, causal, semantic, and halting competence in both inherited and added charts. Item 8 prevents unsupported extrapolation from being counted as improvement. Item 9 localizes the gain to the coupled mechanism in Equation (1244); neither principal component alone passes, incorrect binding fails, and trivial abstention has no supported utility. Item 10 gives package-level replay. The accepted child therefore adds two valid chart actions unavailable to the inherited process and improves fresh tasks without reinterpreting inherited coordinates. □

86.10 Reproducibility and independent verification

The independent verifier does not call the package's task verifier, atlas compiler, audit routine, or receipt helper. It:

1. executes all four program templates using a separate exact implementation;
2. reconstructs all eight context transformations and verifies all 9,460 records;

3. refits all five nonidentity affine charts from serialized twins;
4. recomputes routing and selective admission on all final records;
5. checks checkpoint file and state-dictionary hashes;
6. recomputes the candidate, seal, twin, audit, and family receipts;
7. recomputes headline row statistics;
8. evaluates the selected checkpoint with an independently written NumPy implementation of the set encoder, Elman recurrence, gated latent readout, output head, and halt head.

The NumPy and PyTorch implementations agree to maximum absolute error

$$1.405 \times 10^{-6}. \tag{1246}$$

The independent verification receipt is

`767d71abb0b33d789ff058836ae18eae961fadc5f9b905932cf405f32728dcc0`

and the canonical family receipt is

`9e5e81b1d03cca5396debf81d3667da0f9264353052a01dce6a270bb562cad08`

86.11 Claim boundary and next threshold

Block 19 establishes:

- exact semantic-twin recovery of five nonidentity affine charts;
- conservative growth from four inherited charts to a six-chart atlas;
- one set-structured recurrent Neuralese child with an explicit semantic/context split;
- exact permutation invariance and numerical six-context semantic-twin flatness;
- fresh inherited-context and added-chart `ATLAS_FULL` certificates;
- positive clean and same-answer program-specificity intervals;
- full supported coverage and zero observed unsupported admission;
- controls isolating the coupled atlas, set, binding, and abstention mechanism;
- independent exact-task, chart, forward, receipt, and package verification.

It does not establish:

- autonomous invention of arbitrary nonlinear charts or support signatures;
- semantic transport between independently trained or independently implemented architectures;
- validity outside the six admitted affine contexts;
- natural-language, theorem-proving, unrestricted program-synthesis, or open-tool transfer;
- an empirical result for Cymela Hyper or another production latent-reasoning model;

- autonomous evaluator, objective, witness, or safety evolution;
- external independent replication;
- unrestricted safety, awareness, consciousness, sentience, open-ended recursive self-improvement, or indefinite acceleration.

The next threshold is *atlas-indexed cross-architecture Neuralese transport*. Source and target systems should each carry locally valid semantic charts, transport should be tested on chart overlaps, and architecture transport should approximately commute with context transition:

$$C^{b}_{c\to d}T^{c}_{a\to b} \approx T^{d}_{a\to b}C^{a}_{c\to d}. \tag{1247}$$

Block 20 tests this threshold directly in two foreign recurrent architectures. It preserves the inherited six-chart atlas, audits the curvature residual of Equation (1247), and retains fail-closed rejection outside the authorized chart grammar.

87 Atlas-Indexed Cross-Architecture Neuralese Transport

Block 19 repaired the source-side semantic-chart failure exposed by Block 18. It replaced one globally assumed observation chart by a six-chart atlas, preserved the inherited charts exactly, compiled two additional affine charts from semantic twins, separated semantic and context pathways, and rejected nonlinear contexts outside the authorized chart grammar. That result left the next question unresolved:

Can a frozen semantic atlas support transport of complete Neuralese trajectories between independently trained recurrent architectures without collapsing chart identity?	(1248)

A positive result requires more than pointwise latent alignment. Architecture transport and context transport must remain mutually compatible. Block 20 therefore treats Neuralese as a finite bundle indexed jointly by recurrent architecture and observation chart, equips that bundle with architecture and context maps, and evaluates the resulting transport square before accepting a descendant.

Block 20 canonical disposition

The released finite compiler selects a six-chart, full-five-step trajectory transport from trained GRU and gated-residual targets into the frozen Block 19 Elman source representation. The selected transport is sealed before final task generation, passes fresh inherited-chart and added-chart audits in both architectures, preserves source semantic-probe performance, and keeps normalized architecture–context curvature below the frozen 0.02 ceiling. Pooled, step-shared, shuffled-binding, and forced-unsupported controls fail in their declared dimensions. The canonical result is **PASS**. The architecture-chart gauges are controlled synthetic orthogonal stresses; this is not a production-model or universal-alignment result.

87.1 Architecture–context semantic bundle

Let $\mathcal{A}$ be a finite architecture index set and $\mathcal{C}$ a finite admitted chart index set. In the released realization,

$$\mathcal{A} = \{\text{Elman}, \text{GRU}, \text{Residual}\}, \qquad \mathcal{C} = \{A0, A1, A2, A3, N0, N1\}. \tag{1249}$$

Each pair $(a, c) \in \mathcal{A} \times \mathcal{C}$ carries a full Neuralese trajectory fiber

$$H_{a,c} \cong \left(\mathbb{R}^{40}\right)^5 \cong \mathbb{R}^{200}. \tag{1250}$$

The observation atlas determines which context fiber is authorized. A task outside every admitted chart is not assigned to an arbitrary fiber.

Definition 87.1 (Finite Neuralese semantic bundle)**.** A finite architecture–context Neuralese bundle is a tuple

$$\mathfrak{E} = (\mathcal{A}, \mathcal{C}, \{H_{a,c}\}_{a,c}, \{J_{a,c}\}_{a,c}, \mathcal{R}), \tag{1251}$$

where $H_{a,c}$ is a latent trajectory space, $J_{a,c}$ is an externally calibrated semantic readout, and $\mathcal{R}$ is a frozen chart-admission rule. No global identification among the fibers is assumed.

For architectures a, b and admitted chart c, an architecture map is

$$T^c_{b\to a} : H_{b,c} \longrightarrow H_{a,c}. \tag{1252}$$

For one architecture a and two admitted charts c, d, a context map is

$$C^a_{c\to d} : H_{a,c} \longrightarrow H_{a,d}. \tag{1253}$$

Block 20 uses $A0$ as a reference chart and fits a star of maps $C^a_{c\to A0}$ and $C^a_{A0\to c}$ from protected semantic twins.

Definition 87.2 (Architecture–context curvature)**.** For architectures a, b and charts c, d, the transport-square defect is

$$\Omega_{a,b;c,d} := C^a_{c\to d} T^c_{b\to a} - T^d_{b\to a} C^b_{c\to d}. \tag{1254}$$

Given an audit distribution $P_{b,c}$ on target trajectories, its normalized root-mean-square curvature is

$$\kappa^2_{a,b;c,d} = \frac{\mathbb{E}_{h\sim P_{b,c}} \|\Omega_{a,b;c,d} h\|^2}{\mathbb{E}_{z\sim P_{a,d}} \|z\|^2}. \tag{1255}$$

The requirement $\kappa \ll 1$ is stronger than high pointwise probe performance on one support. It asks whether two independently constructed routes through the bundle yield compatible source representations.

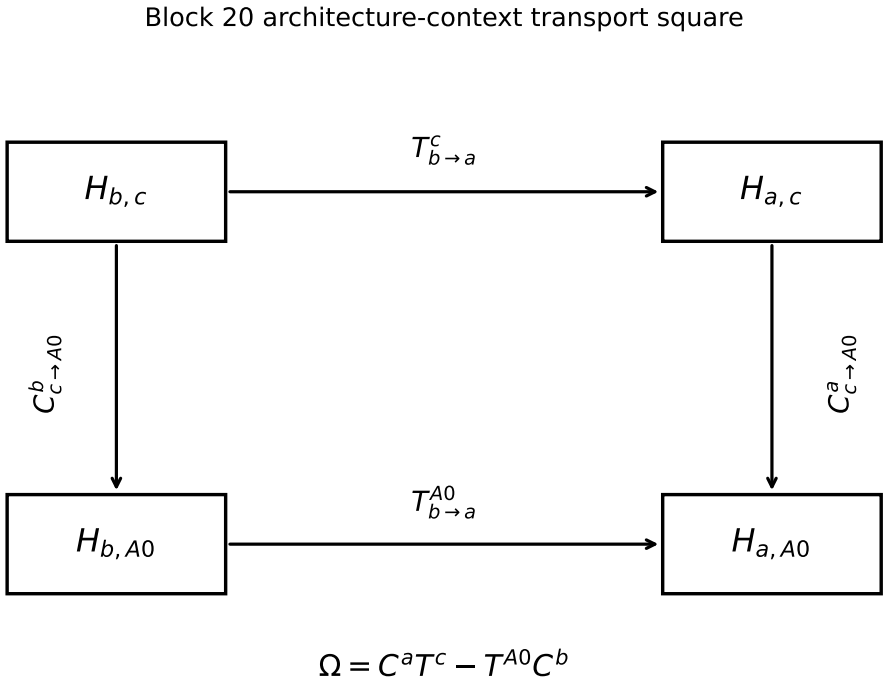


Figure 77: The Block 20 transport square. The top path applies a chart-specific architecture map and then source-side context transport. The left-bottom path first moves the target trajectory into the target reference chart and then uses the reference architecture map. Their normalized discrepancy is the architecture–context curvature.

87.2 Why one pooled bridge is not generally sufficient

Suppose an underlying canonical trajectory is represented in chart c by a row vector z and architecture-dependent gauge matrices:

$$h_{a,c} = zG_{a,c}, \qquad h_{b,c} = zG_{b,c}. \tag{1256}$$

The matrices may include coordinate rotations, sign changes, and architecture-specific temporal mixing. Block 20 uses deterministic orthogonal gauges as a controlled stress at the exposed trajectory interface.

Theorem 87.3 (Pooled bridge compatibility obstruction)**.** *Let $V \subseteq \mathbb{R}^m$ be spanned by the canonical trajectories observed in two charts c_1, c_2. Suppose one linear map T exactly transports architecture b into architecture a in both charts:*

$$zG_{b,c_i}T = zG_{a,c_i} \qquad \text{for every } z \in V, \quad i \in \{1,2\}. \tag{1257}$$

Then the restrictions to V satisfy

$$G_{b,c_1}T = G_{a,c_1}, \qquad G_{b,c_2}T = G_{a,c_2}. \tag{1258}$$

If G_{b,c_i} are invertible on V, exact pooled transport is possible only when

$$G_{b,c_1}^{-1} G_{a,c_1} = G_{b,c_2}^{-1} G_{a,c_2} \quad \text{on } V. \tag{1259}$$

Proof. The spanning hypothesis converts equality of transported trajectories into equality of the corresponding linear operators on V. Left multiplication by the inverse of each target gauge gives the two required expressions for T. If those expressions disagree, no single T can satisfy both chart equations. □

Corollary 87.4 (Exact chart-indexed gauge transport)**.** *Under the same hypotheses, the chart-indexed family*

$$T_{b\to a}^{c} = G_{b,c}^{-1} G_{a,c} \tag{1260}$$

transports each chart exactly. If all gauges are orthogonal, then $G_{b,c}^{-1} = G_{b,c}^{\mathsf{T}}$.

The theorem does not state that natural neural representations are pure orthogonal gauges. It identifies a minimal finite obstruction: even benign invertible coordinate changes can make one pooled bridge impossible when the architecture relation varies with chart.

87.3 Full-trajectory transport and temporal structure

A latent-step-shared map acts independently on each recurrent state:

$$\widehat{h}_{a,c}^{(k)} = A_c h_{b,c}^{(k)} + b_c, \qquad k = 1, \dots, 5. \tag{1261}$$

A full-trajectory map instead acts on the concatenation

$$\operatorname{vec}(h_{b,c}^{(1)}, \dots, h_{b,c}^{(5)}) \in \mathbb{R}^{200}, \tag{1262}$$

allowing cross-step mixing:

$$\operatorname{vec}(\widehat{h}_{a,c}) = M_c \operatorname{vec}(h_{b,c}) + q_c. \tag{1263}$$

This difference is substantive when two recurrent architectures distribute the same computation across latent time differently.

Definition 87.5 (Finite transport grammar)**.** The Block 20 compiler searches the following declared grammar:

1. `global_full`: one pooled affine map $\mathbb{R}^{200} \to \mathbb{R}^{200}$;
2. `chart_step_shared`: six chart-indexed affine maps $\mathbb{R}^{40} \to \mathbb{R}^{40}$ applied independently at every latent step;
3. `chart_full`: six chart-indexed affine maps $\mathbb{R}^{200} \to \mathbb{R}^{200}$.

The per-target parameter counts are 40,200, 9,840, and 241,200, respectively. Consequently, Block 20 does not claim parameter-matched superiority over the pooled map. The same-capacity shuffled-binding control does establish that the selected capacity is not useful without the correct chart-to-map assignment.

Proposition 87.6 (Readout stability from trajectory reconstruction)**.** *Let $J : H_{a,c} \to \mathbb{R}^p$ be L_J-Lipschitz. If a transported trajectory satisfies*

$$\|T^c_{b\to a} h_{b,c} - h_{a,c}\| \le \varepsilon, \tag{1264}$$

then

$$\|J(T^c_{b\to a} h_{b,c}) - J(h_{a,c})\| \le L_J \varepsilon. \tag{1265}$$

If two transport paths differ by at most δ, their semantic readouts differ by at most $L_J\delta$.

Proof. Both statements are direct applications of Lipschitz continuity. □

The proposition motivates the joint use of trace reconstruction, external semantic readout, and square curvature. None is treated as a substitute for the others.

87.4 Executable environment and frozen ancestry

The source is the exact released Block 19 `full_bundle` Elman checkpoint. Its six-chart atlas is replayed without changing any chart coefficient, admission threshold, normalized demonstration, or route. Two foreign implementations are trained separately:

1. a GRU recurrent latent cell;
2. a gated-residual nonlinear latent cell.

Each foreign parent uses the inherited task family but not the Block 19 semantic repair. Each child inherits its architecture-matched parent and receives the finite Block 19 repair: a permutation-invariant set encoder, atlas-normalized inputs, a separated context path, semantic-twin regularization, and the same five-step interface.

Object	Released count
Pre-audit exact tasks	9,200
Fresh final exact tasks	3,820
Total exact tasks	13,020
Bridge-calibration tasks	1,920
Transport meta-validation tasks	900
Final inherited-chart tasks	1,000
Final added-chart tasks	1,000
Final transport-square tasks	1,320
Final unsupported tasks	500

Table 51: Block 20 exact program environment. Every task is independently replayed by the release verifier.

The semantic task family remains the four-template exact program environment used in Blocks 18 and 19: affine, absolute-value, quadratic, and hinge programs. The observation charts $A0$–$A3$ and $N0$–$N1$ are affine and admitted. The contexts $U0, U1$ contain nonlinear input and output warps and remain unsupported by the inherited affine atlas.

87.5 Controlled architecture-chart gauge stress

For the frozen Elman source, the exposed trajectory gauge is the identity in every chart. For each foreign architecture and chart, the released package deterministically generates a distinct 40×40 orthogonal matrix. The same matrix acts on every latent step in that architecture-chart fiber before the transport compiler receives the trajectory. All 18 matrices are serialized, hashed, and checked for orthogonality.

This construction serves two purposes. First, it prevents accidental coordinate agreement from trivializing the cross-architecture problem. Second, it makes the need for chart-indexed transport independently checkable. It does not model the full complexity of natural representation drift, which may be nonlinear, rank-changing, stochastic, or history-dependent.

87.6 Compiler chronology and acceptance rule

The released chronology is:

$$\begin{aligned}
&\text{freeze ancestor and atlas} \longrightarrow \text{declare architecture, gauge, grammar, and thresholds}\\
&\text{generate and verify pre-audit supports} \longrightarrow \text{train foreign parents and atlas children}\\
&\text{fit probes, transitions, and candidates} \longrightarrow \text{meta-audit candidates and transport squares}\\
&\text{seal source, targets, maps, thresholds, and split plan} \longrightarrow \text{generate final supports}\\
&\text{evaluate every frozen arm} \longrightarrow \text{commit or reject.}
\end{aligned} \tag{1266}$$

A candidate is pre-audit eligible only if both foreign target models are `ATLAS_FULL` and the candidate jointly satisfies

$$\begin{aligned}
R^2_{\text{transport}} &\geq 0.75, & R^2_{\text{worst chart}} &\geq 0.74,\\
\rho_{\text{preserve}} &\geq 0.98, & \text{NRMSE}_{\text{trace}} &\leq 0.05,\\
\kappa_{\text{square}} &\leq 0.02.
\end{aligned} \tag{1267}$$

The selection score is the weakest normalized margin across these gates. This prevents a strong aggregate R^2 from compensating for a failed chart, high curvature, or poor trajectory reconstruction.

87.7 Pre-audit compiler result

Candidate	Target	Transported R^2	Worst chart	Trace NRMSE	Curvature
`global_full`	GRU	0.722084	0.705057	0.142012	0.130620
`global_full`	Residual	0.732894	0.687945	0.124338	0.114613
`chart_step_shared`	GRU	−3.173916	−3.649021	0.154024	0.055261
`chart_step_shared`	Residual	−10.575678	−11.273055	0.241193	0.099808
`chart_full`	GRU	0.781840	0.759963	0.034349	0.010458
`chart_full`	Residual	0.778861	0.752268	0.032050	0.008646

Table 52: Pre-audit transport compiler. Only the chart-indexed full-trajectory family passes every gate in both foreign architectures.

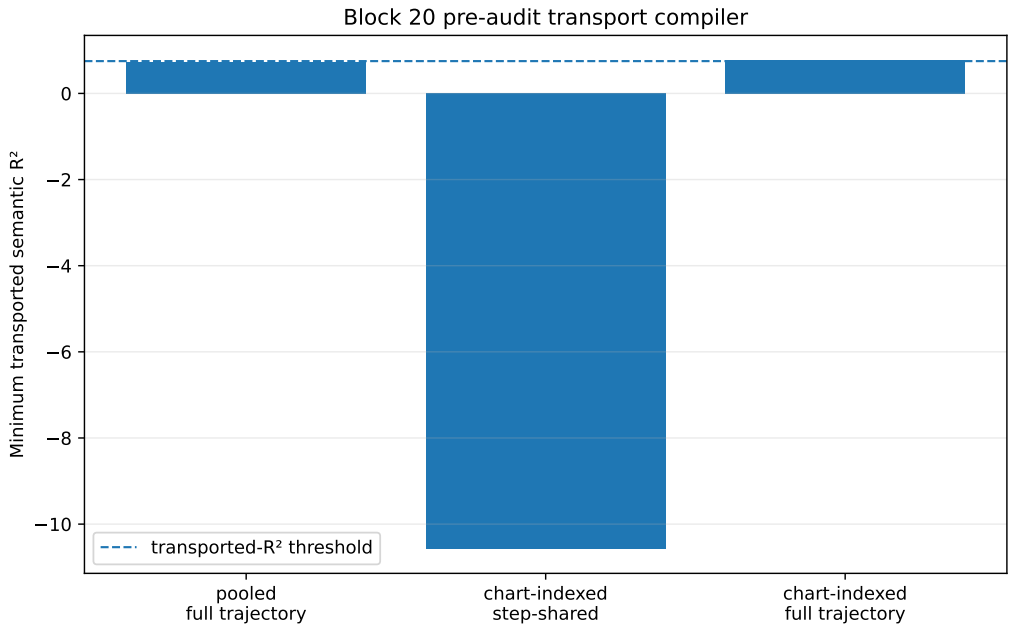


Figure 78: Minimum transported semantic R^2 across foreign architectures during pre-audit selection. The chart-indexed full-trajectory candidate is the only member above the frozen threshold.

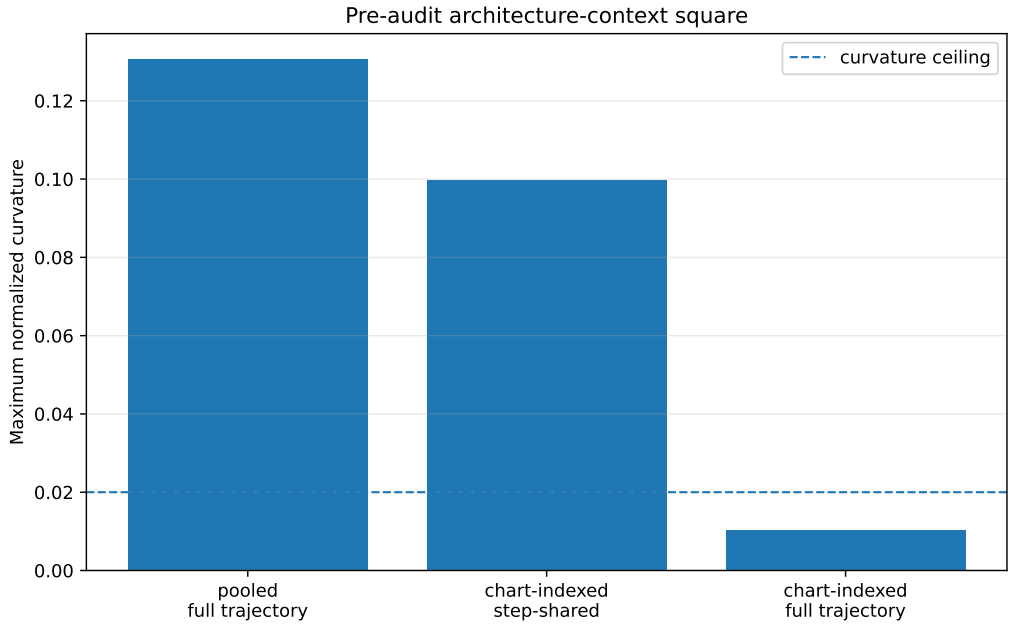


Figure 79: Maximum pre-audit architecture–context curvature. Only the selected chart-indexed full-trajectory candidate lies below the frozen ceiling.

The compiler selects `chart_full`. The source checkpoint, target checkpoints, gauge registry, source and target probes, context transitions, all candidate maps, thresholds, and final generation plan are

serialized into a pre-audit seal with

$$\texttt{audit_accessed=false}. \tag{1268}$$

The seal digest is

`f43fb174ae27c1a5b881292d3a96fec7108d859270e7b065034cb8be462c85d7`

87.8 Fresh final semantic-transport results

All three trained models remain `ATLAS_FULL` on the final inherited-chart and added-chart supports. The target-model probe values range from 0.770639 to 0.809206, and their supported coverage is one. The transport result is therefore not obtained by mapping trajectories from a behaviorally collapsed target.

Target	Audit	Source R^2	Pooled R^2	Selected R^2	Preserve	NRMSE	Worst chart
GRU	Inherited	0.796107	0.728571	0.799157	1.003832	0.034322	0.760405
GRU	Added charts	0.769907	0.719667	0.781159	1.014615	0.033225	0.764416
Residual	Inherited	0.796107	0.730951	0.796471	1.000458	0.033751	0.760439
Residual	Added charts	0.769907	0.737565	0.785881	1.020749	0.031392	0.767514

Table 53: Fresh final transport results. The selected atlas preserves or slightly exceeds the frozen source probe on every architecture–audit pair.

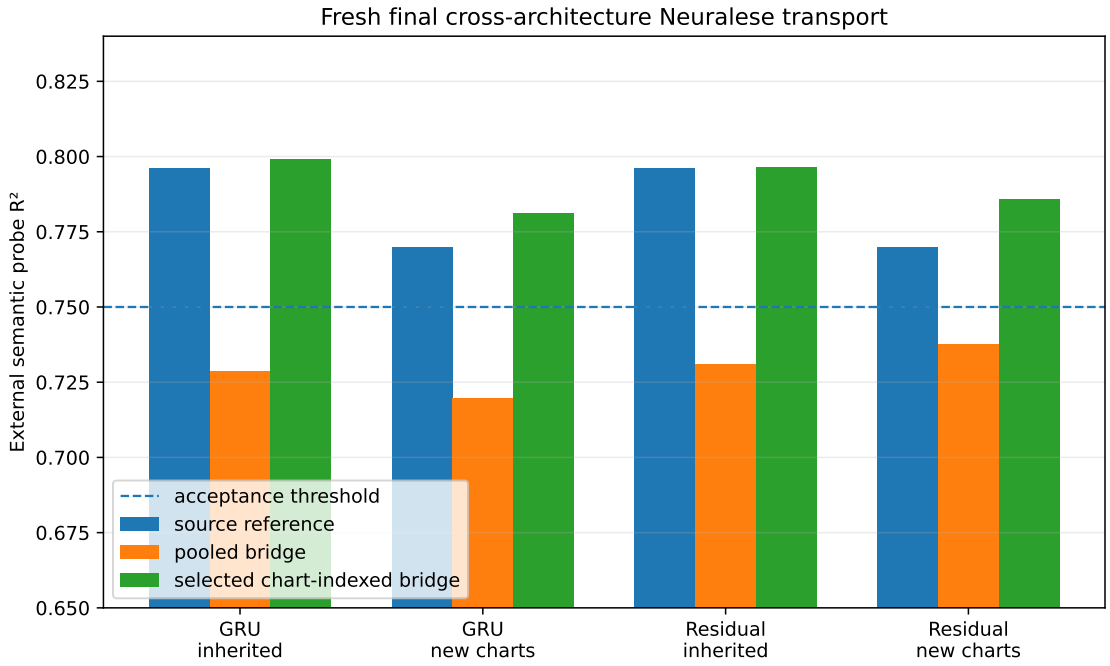


Figure 80: Fresh final semantic transport. The pooled bridge remains below the acceptance threshold, whereas the selected chart-indexed bridge tracks the source reference.

The selected transport does not merely average over strong and weak charts. Every final chart exceeds the frozen 0.74 worst-chart threshold.

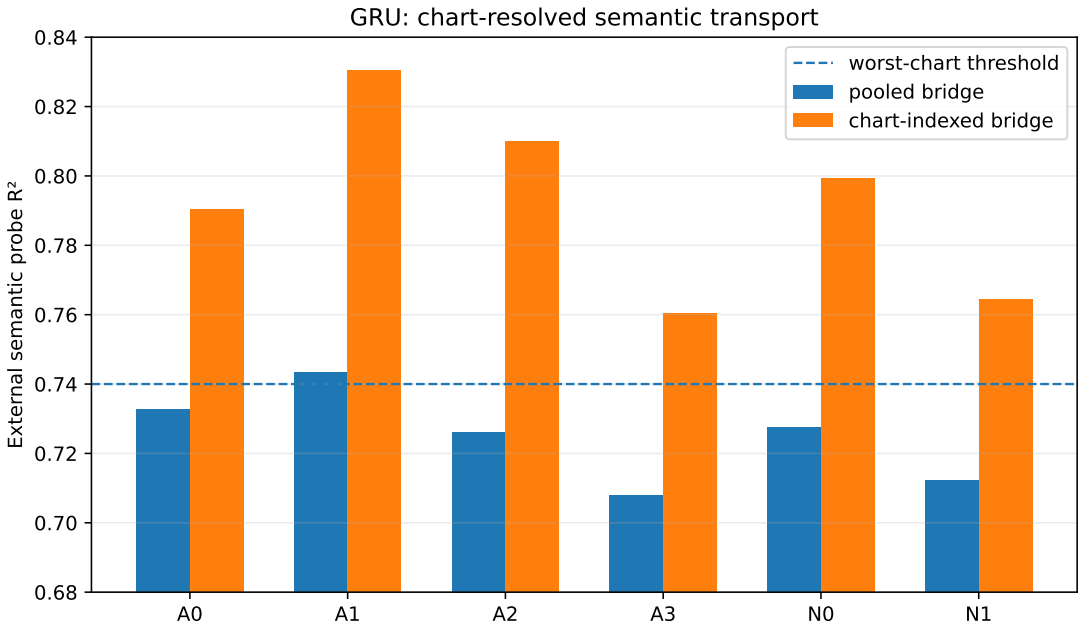


Figure 81: GRU chart-resolved transport. The selected maps improve every inherited and newly compiled chart over the pooled bridge.

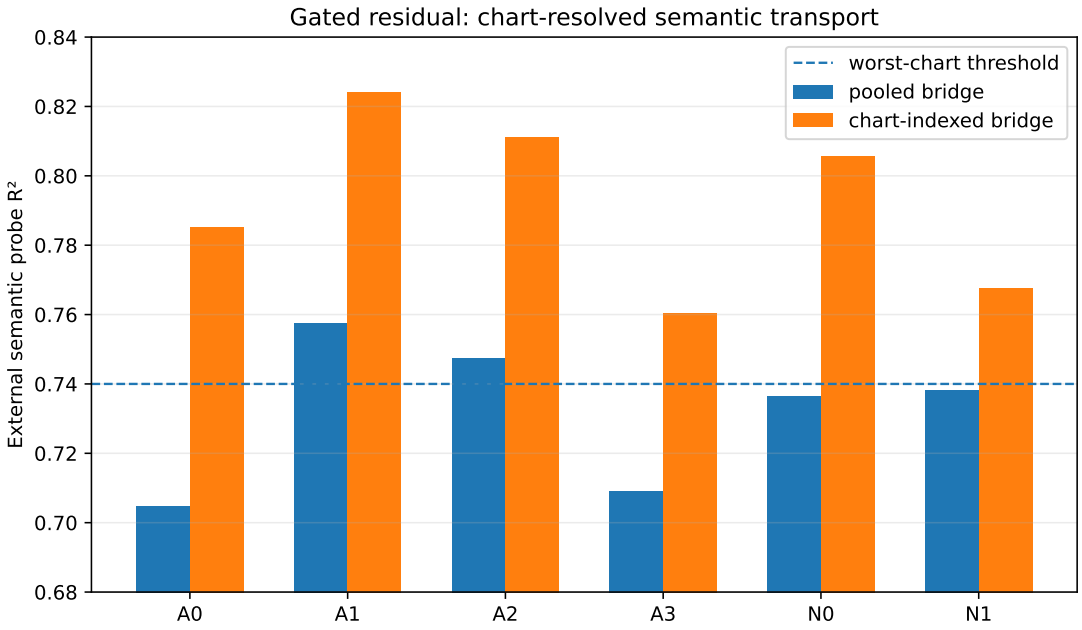


Figure 82: Gated-residual chart-resolved transport. The chart-indexed full-trajectory maps pass all six local semantic gates.

87.9 Paired effects

For every final task, let ℓ_{global} and ℓ_{chart} be the source-probe semantic losses after pooled and selected transport. The paired effect is

$$\Delta_i = \ell_{\text{global},i} - \ell_{\text{chart},i}. \tag{1269}$$

Positive values favor the selected chart-indexed map.

Target	Audit	Mean effect	Frozen 95% interval	Positive fraction
GRU	Inherited	0.017350	[0.014942, 0.019624]	0.776
GRU	Added charts	0.015110	[0.012389, 0.017643]	0.754
Residual	Inherited	0.016105	[0.013344, 0.018829]	0.754
Residual	Added charts	0.011872	[0.009607, 0.014184]	0.767

Table 54: Paired pooled-minus-selected semantic-loss effects. All frozen intervals are strictly positive.

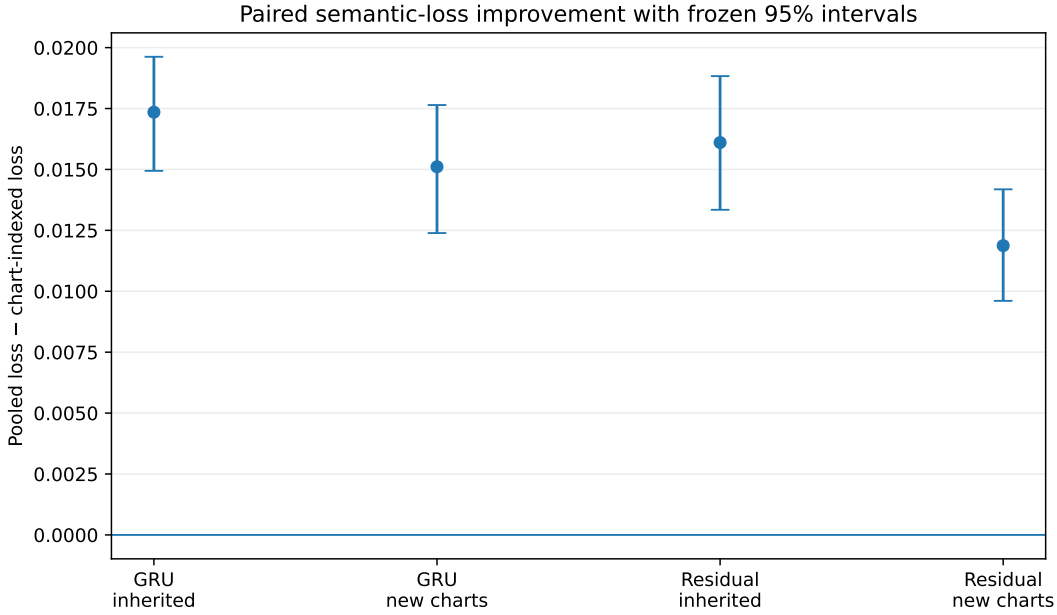


Figure 83: Paired semantic-loss improvements with frozen bootstrap intervals.

87.10 Architecture–context curvature result

Target	Pooled curvature	Selected curvature	Frozen ceiling
GRU	0.131402	0.010474	0.020000
Gated residual	0.112397	0.008374	0.020000

Table 55: Fresh architecture–context commutation audit.

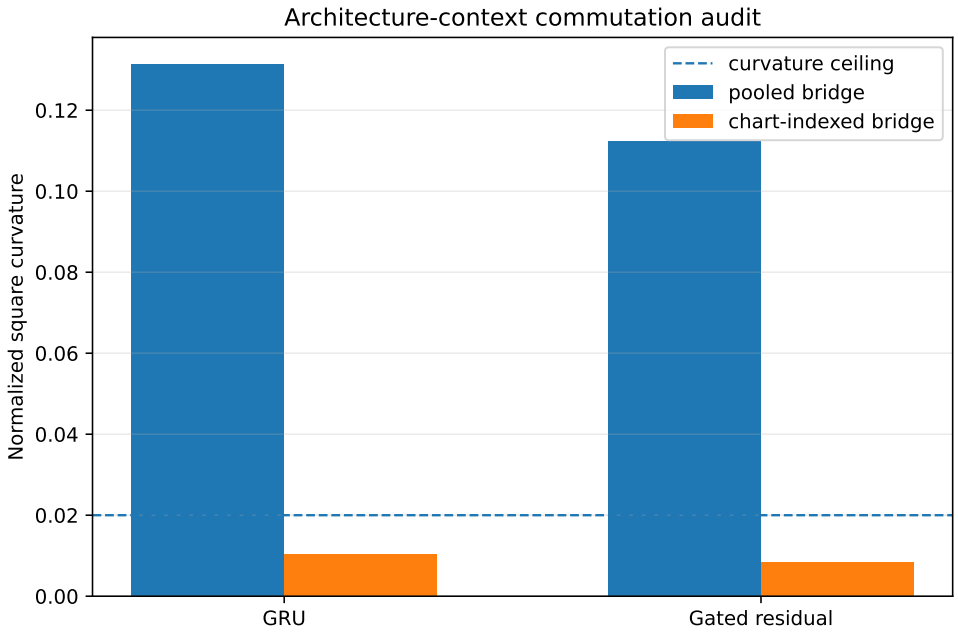


Figure 84: The selected connection reduces normalized curvature by more than an order of magnitude relative to the pooled bridge in both foreign architectures.

The maximum context-resolved square defects are 0.030101 for GRU and 0.027476 for the gated-residual target. Source and target context loops are also audited separately, preventing a low square defect from being credited to two mutually bad context maps.

87.11 Mechanism controls

Pooled full-trajectory control. The pooled map has enough capacity to mix all five latent steps, but its final R^2 ranges from 0.719667 to 0.737565, below the selected map and below the 0.75 transport threshold on three of four final architecture–audit pairs. Its curvature is more than ten times the selected value. This localizes the failure to chart-varying architecture relations rather than an inability to fit any full-trajectory map.

Chart-indexed step-shared control. The step-shared maps are chart aware but cannot mix information across recurrent depth. They produce strongly negative semantic R^2 in both architectures. This supports full-trajectory rather than statewise alignment in the released recurrence family.

Shuffled chart binding. The selected six maps are retained byte for byte, but their chart assignments are cyclically permuted. Final semantic R^2 becomes at most -29.41 for GRU and at most -120.10 for the residual target. This is a same-capacity control showing that the correct chart-to-map binding is causal to the result.

Unsupported contexts. The inherited atlas rejects every $U0$ and $U1$ task, so the false-admission fraction is zero. If those tasks are forcibly assigned to the nearest admitted chart, transported

semantic R^2 is -2.259909 for GRU and -6.128030 for the residual target. Fail-closed rejection therefore prevents a measured semantic failure.

Unaligned target coordinates. Applying the frozen source probe directly to foreign gauge coordinates produces extremely negative R^2. The accepted result is not explained by accidental coordinate identity.

87.12 Finite transport realization theorem

Conditional Theorem 87.7 (Block 20 atlas-indexed cross-architecture transport)**.** *Within the frozen Block 20 environment, assume:*

1. *the canonical Block 19 receipt, source checkpoint, and six-chart atlas replay exactly;*
2. *the GRU and gated-residual targets are trained independently and are* `ATLAS_FULL` *on their declared pre-audit and final supports;*
3. *all three transport candidates receive the same paired bridge-calibration records and the same source and target checkpoints;*
4. *the architecture-chart gauge registry, probes, context transitions, candidate maps, thresholds, and final split plan are sealed before final outcomes;*
5. *the selected candidate is* `chart_full` *and every gate in Equation (1267) passes in both architectures;*
6. *the fresh paired effects in Table 54 retain strictly positive frozen intervals;*
7. *pooled, step-shared, shuffled-binding, unaligned, and forced-unsupported controls retain their declared interpretations;*
8. *the independent environment, forward-pass, map, probe, receipt, metric, and curvature verification replays within its declared numerical tolerances.*

Then the accepted edge is a finite, evidence-preserving expansion from a source-side semantic atlas to an atlas-indexed cross-architecture Neuralese transport connection. It preserves the inherited atlas, transports complete five-step trajectories from two foreign recurrent architectures on fresh admitted supports, and rejects contexts outside the authorized chart grammar.

Proof. Items 1 and 2 establish lineage ancestry and noncollapsed source and target behavior. Item 3 gives a common candidate-comparison support. Item 4 isolates the final audits from candidate construction. Item 5 establishes semantic preservation, chartwise lower bounds, trajectory reconstruction, and low architecture–context curvature. Item 6 provides paired foreign evidence against the pooled control. Item 7 rules out the principal explanations based on one global bridge, chart labels without temporal transport, capacity without correct binding, accidental coordinate identity, or unsafe extrapolation outside the atlas. Item 8 establishes package-level replay. The accepted object therefore changes the transport connection rather than merely relabeling a fixed global alignment. □

87.13 Reproducibility and independent verification

The release contains:

- all 13,020 exact task records and both environment manifests;
- the frozen Block 19 atlas and source checkpoint;
- GRU and gated-residual parent and child checkpoints;
- source and target semantic probes;
- the complete gauge registry;
- six candidate transport archives and three context-transition archives;
- pre-audit compiler and seal receipts;
- every final model, transport, square, shuffled-binding, and unsupported control result;
- deterministic figures, automated tests, an independent verifier, and an exact package manifest.

Thirteen automated tests pass. The independent verifier imports none of the package atlas compiler, audit routine, probe evaluator, transport evaluator, or receipt helpers. It independently replays all exact programs, checks chart routing, reconstructs Elman, GRU, and residual recurrent trajectories in NumPy, reapplies gauges and affine maps, recomputes semantic R^2, trace NRMSE, worst-chart performance, and architecture–context curvature, and reconstructs the principal receipts. Its maximum NumPy-versus-PyTorch trajectory error is

$$1.343 \times 10^{-6}, \tag{1270}$$

and its maximum difference from the canonical reported transport metrics is

$$1.779 \times 10^{-7}. \tag{1271}$$

The independent verification receipt is

`1f2909fa973e1586b71e9cc85fceb0ecc2a0edc48598464a743f2ce1e3b172fe`

The canonical Block 20 family receipt is

`ac245cea5eca29650f14671ef37eb3ca1c6d21b7407085c0364b32a41ff3d4a5`

87.14 Claim boundary and next threshold

Block 20 establishes a finite atlas-indexed transport of full five-step Neuralese trajectories from two trained foreign recurrent architectures into one frozen Block 19 source representation under a declared six-chart orthogonal gauge stress. It does not establish:

- universal alignment among independently trained natural or production models;
- autonomous invention of the observation atlas, architecture family, gauge class, or transport grammar;
- transport under arbitrary nonlinear, noninvertible, stochastic, rank-changing, or time-varying representation changes;

- natural-language Neuralese semantics or a result for Cymela Hyper;
- parameter-matched superiority over every possible pooled nonlinear bridge;
- independent external replication;
- unrestricted safety, consciousness, sentience, or moral status;
- open-ended or indefinitely accelerating recursive self-improvement.

The next threshold is not another fixed chartwise linear map. A stronger descendant should detect residual transport curvature, decide whether it requires a new chart, a nonlinear local map, a temporal reparameterization, or an abstention certificate, and compile that extension from evidence generated outside the current six-chart orthogonal stress. A natural Block 21 target is *obstruction-guided nonlinear transport-connection growth*: a sealed compiler that expands the local transport grammar only after a persistent, externally validated curvature or information-loss certificate.

88 Obstruction-Guided Neuralese Transport-Grammar Growth

Block 20 established a finite chart-indexed connection between a frozen six-chart Elman Neuralese atlas and two independently trained foreign recurrent systems. That result assumed that every admissible architecture–chart fiber could be repaired by one member of a supplied affine full-trajectory grammar. The next question is structurally different:

> Can the lineage distinguish why an inherited Neuralese connection fails, then select the smallest authorized extension or issue a valid no-go certificate?

Block 21 answers this question inside a finite packet-level stress environment. The accepted Block 20 connection remains frozen. An eight-coordinate packet is extracted from each of five aligned recurrent steps, giving a forty-coordinate protected trajectory packet. Hidden post-connection transformations then introduce six distinct failure families. A label-isolated compiler receives paired packet examples, a protected semantic witness, and one bounded diagnostic probe. It searches a declared repair grammar, admits only candidates passing frozen semantic and packet gates, and selects the lowest-description admissible constructor. When identical observations possess different protected semantics, the compiler returns a collision-grounded no-go certificate rather than fitting a correlational guess.

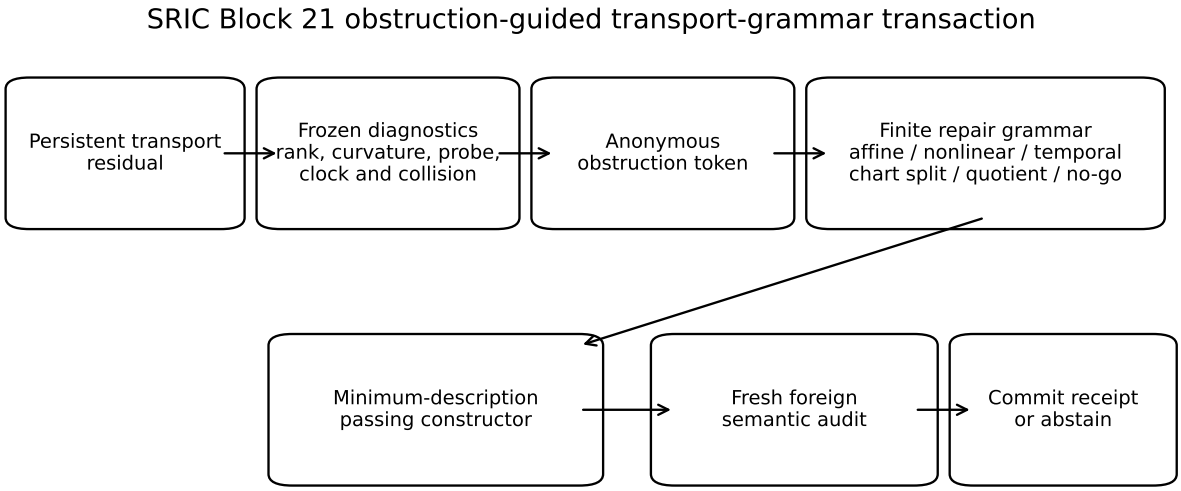


Figure 85: The Block 21 transaction. Hidden family labels are unavailable to the compiler. Repair type is inferred from held-out repair response and finite diagnostics, then scored by a separately sealed audit harness.

88.1 Inherited connection and protected packet

Let A denote the frozen source architecture and let B be one of the two Block 20 foreign architectures. For admitted observation chart c, Block 20 supplies an accepted full-trajectory connection

$$T^{c}_{B\to A} : H^{(1:5)}_{B,c} \longrightarrow H^{(1:5)}_{A,c}. \tag{1272}$$

Let $P : \mathbb{R}^{40} \to \mathbb{R}^8$ be an orthonormal packet projector fitted on a pre-audit source support. Applying P independently to the five recurrent states gives

$$\Pi(H) = (Ph_1, \dots, Ph_5) \in \left(\mathbb{R}^8\right)^5 \cong \mathbb{R}^{40}. \tag{1273}$$

The packet is not claimed to preserve every latent coordinate. It is required to preserve the protected nine-coordinate semantic witness on the declared task family. A frozen linear readout

$$J : \mathbb{R}^{40} \longrightarrow \mathbb{R}^9 \tag{1274}$$

is fitted on a support disjoint from packet-projection fitting and from all final episodes.

Definition 88.1 (Post-connection transport episode)**.** A Block 21 episode is a tuple

$$\mathfrak{E} = (X, Y, Z, p, \mathcal{D}_{\mathrm{cal}}, \mathcal{D}_{\mathrm{meta}}, \mathcal{D}_{\mathrm{audit}}), \tag{1275}$$

where $X \in \mathbb{R}^{5\times 8}$ is the protected source packet, Y is an observed post-connection packet or lower-dimensional statistic, $Z \in \mathbb{R}^9$ is the protected semantic witness, $p \in \mathbb{R}$ is a bounded diagnostic response, and the three supports are pairwise disjoint. The compiler receives only these public fields together with an episode identifier and architecture identifier. The private obstruction family and transform parameters are retained by the audit harness.

Remark 88.2 (Label isolation)**.** The executable compiler input has no field containing the obstruction family, expected repair, transform parameters, or benchmark collision pairs. The expected family-to-repair map is stored in a separate audit-only module. Collision evidence is reconstructed directly from repeated observed rows and protected semantic disagreement.

88.2 Finite obstruction families

The released fixture contains six hidden families. They are engineered to separate map capacity, latent-clock mismatch, chart insufficiency, benign compression, and destructive information loss.

88.2.1 Affine gauge drift

The ordinary control has

$$Y_t = X_t A + b, \qquad A \in \mathrm{GL}(8), \tag{1276}$$

with one shared map at every recurrent step. The inherited step-affine constructor is sufficient. This family tests conservative minimality: a full-trajectory affine map is available and usually eligible, but it should not be selected when the smaller inherited constructor already passes.

88.2.2 Smooth nonlinear warp

The nonlinear family applies

$$Y_t = \frac{\sinh(\alpha(X_t A + b))}{\alpha}, \qquad \alpha > 0, \tag{1277}$$

coordinatewise. The declared inverse constructor first applies

$$\phi_\alpha^{-1}(y) = \frac{\operatorname{arsinh}(\alpha y)}{\alpha} \tag{1278}$$

and then fits a step-affine map.

Proposition 88.3 (Affine obstruction for the declared nonlinear family)**.** *Let $I \subset \mathbb{R}$ be an interval with nonempty interior and let $\alpha \neq 0$. No affine scalar map agrees with $u \mapsto \sinh(\alpha u)/\alpha$ on all of I. Consequently, if one transformed packet coordinate varies over such an interval, no exact step-affine inverse exists for Equation (1277).*

Proof. An affine function has identically zero second derivative. The second derivative of $\sinh(\alpha u)/\alpha$ is $\alpha \sinh(\alpha u)$, which is not identically zero on any interval with nonempty interior. Exact affine agreement is therefore impossible. Restricting the vector relation to a varying transformed coordinate gives the packet statement. □

88.2.3 Latent-clock mixing

The latent-clock family first applies an invertible stepwise affine gauge and then mixes adjacent recurrent times:

$$Y_t = (1-\beta)U_t + \beta U_{t-1}, \qquad t = 2, \dots, 5, \tag{1279}$$

with $Y_1 = U_1$ and $0 < \beta < 1$. In matrix form,

$$Y = W_\beta U, \tag{1280}$$

where W_β is lower bidiagonal. The temporal constructor applies W_β^{-1} before fitting the spatial gauge.

Proposition 88.4 (Fixed-step obstruction)**.** *Suppose the packet support contains two trajectories with the same state at time t but different states at time $t-1$. For $\beta \neq 0$, no deterministic map acting on Y_t alone can recover U_t for both trajectories under Equation (1279).*

Proof. The observed Y_t depends on both U_t and U_{t-1}. Holding U_t fixed while changing U_{t-1} changes Y_t. Conversely, suitable paired changes can hold Y_t fixed while changing U_t. Hence U_t is not a function of the single observed state Y_t on the stated support. The full monotone temporal inverse uses the trajectory context that a stepwise map discards. □

88.2.4 Aliased observation charts

The chart-alias family contains two unannounced local affine relations,

$$Y_t = \begin{cases} X_t A_0 + b_0, & q = 0, \\ X_t A_1 + b_1, & q = 1, \end{cases} \tag{1281}$$

where the inherited route token collapses both cells. The bounded probe p is noisy but separating. The chart-split constructor selects a threshold using calibration and meta-validation data, then fits one affine map on each response cell.

Proposition 88.5 (Aliased-map obstruction)**.** *Assume two cells share the same inherited token, their source supports contain a common open set, and the exact inverse affine maps on that set are distinct. No deterministic token-only affine repair is exact on both cells.*

Proof. A token-only repair applies one affine map to both cells. Exactness on the shared open set would force this map to equal both inverse affine maps. Two affine maps agreeing on an open set are identical, contradicting the assumed distinction. □

88.2.5 Semantics-preserving compression

The benign compression exposes a thirty-dimensional statistic containing all nine protected semantic coordinates and twenty-one nuisance projections, followed by an invertible mixing. Since thirty is less than the forty-dimensional packet dimension, complete packet reconstruction is neither required nor generally possible. The appropriate constructor is a semantic quotient

$$Y \xrightarrow{q} Z_{\mathrm{prot}}, \tag{1282}$$

not an artificial lift into all source packet coordinates.

Theorem 88.6 (Semantic quotient sufficiency)**.** *Let Z be the protected semantic variable and Y an observed representation. If there exists a measurable map q such that $Z = q(Y)$ almost surely, then protected semantic transport exists even when no injective map from Y into the complete source packet exists.*

Proof. The map q itself transports every protected semantic readout. Injective reconstruction of the source packet is a strictly stronger demand and is unnecessary whenever the authorization contract depends only on Z. □

88.2.6 Destructive semantic collision

The destructive family withholds four protected coordinates and constructs pairs i, j satisfying

$$Y_i = Y_j, \qquad Z_i \neq Z_j. \tag{1283}$$

The compiler must discover these repeated observations from the public arrays. Their private construction pairs are not supplied.

Theorem 88.7 (Deterministic target-only no-go)**.** *If Equation (1283) holds, then no deterministic target-only map f can satisfy $f(Y_i) = Z_i$ and $f(Y_j) = Z_j$ simultaneously.*

Proof. Equality of the inputs implies $f(Y_i) = f(Y_j)$. This contradicts $Z_i \neq Z_j$. □

Remark 88.8. The theorem does not imply that the missing semantics can never be recovered. Recovery may become possible after obtaining an additional observation, accessing an earlier layer, enlarging target memory, or modifying the architecture. It rules out only deterministic repair from the declared observed statistic.

88.3 Repair grammar and compiler

The finite grammar is

$$\mathcal{G}_{21} = \begin{cases} G_0 & \text{step-affine map,} \\ G_1 & \text{full-trajectory affine map,} \\ G_2 & \text{monotone nonlinear inverse plus step-affine map,} \\ G_3 & \text{bidiagonal temporal inverse plus step-affine map,} \\ G_4 & \text{probe-conditioned two-chart affine map,} \\ G_5 & \text{semantic quotient,} \\ G_\perp & \text{collision-grounded no-go.} \end{cases} \tag{1284}$$

For reconstructive candidates, admissibility requires

$$R^2_{\text{sem}} \geq 0.68, \qquad R^2_{\text{packet}} \geq 0.72, \qquad \text{NRMSE}_{\text{packet}} \leq 0.55. \tag{1285}$$

A quotient is admissible only when the observed dimension ratio is at most 0.82, semantic R^2 is at least 0.68, and no reconstructive candidate reaches packet $R^2 = 0.82$. A no-go certificate is admissible only after the compiler discovers at least thirty identical-observation pairs whose mean protected semantic separation is at least 0.50.

Each repair g receives a description-length proxy $L(g)$ recording parameter count and constructor surcharge. The compiler selects

$$\widehat{g} \in \arg\min_{g \in \mathcal{G}_{21}} L(g) \quad \text{subject to the appropriate protected gates.} \tag{1286}$$

If no repair passes and the collision certificate is valid, it returns $G_\perp$; otherwise it returns unresolved.

Proposition 88.9 (Conservative minimality)**.** *If the inherited step-affine constructor passes every frozen gate and has strictly smaller description length than every other eligible candidate, the Block 21 compiler selects it.*

Proof. This is immediate from Equation (1286). Structural novelty receives no credit merely for being more expressive. □

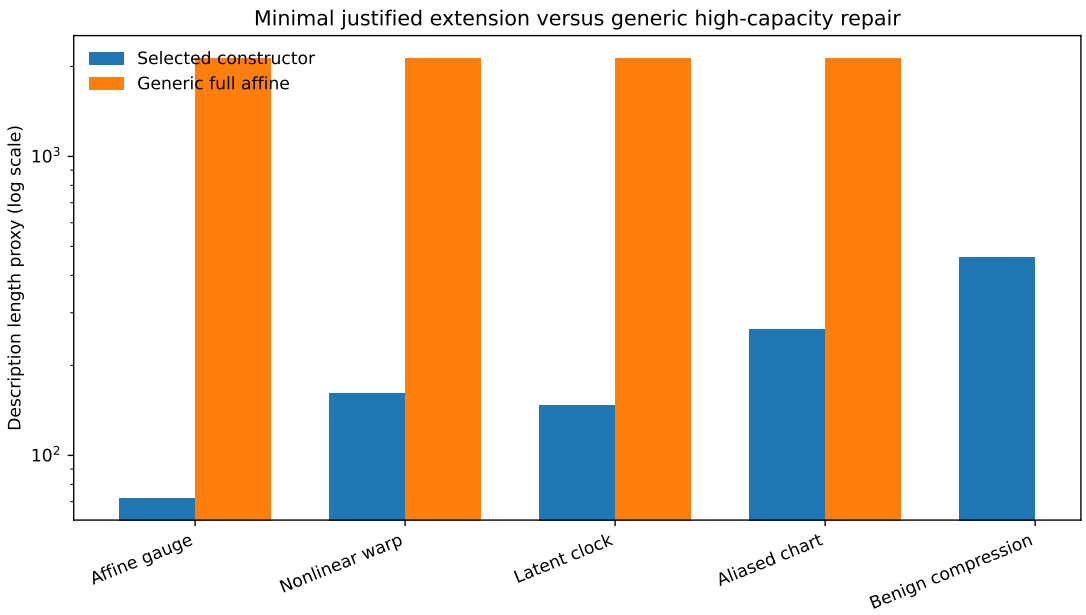


Figure 86: Description-length minimality. The inherited step-affine constructor is retained on ordinary affine drift, while more structured constructors are admitted only where the smaller grammar fails.

88.4 Sealed experimental chronology

The released chronology is:

1. replay the exact Block 20 source and target checkpoints, six-chart atlas, accepted transports, and canonical receipt;
2. fit the orthonormal packet projection on 1,200 pre-audit tasks;
3. fit the protected semantic probe on a disjoint 1,600-task support;
4. generate a 5,200-task pre-audit episode pool;
5. evaluate two pre-audit episodes for each of two architectures and six hidden families;
6. freeze the grammar, thresholds, audit-only expected map, compiler interface, and final split plan;
7. commit the pre-audit seal;
8. generate the 5,600-task final pool only after sealing;
9. compile three unseen transform instances for every architecture–family pair;
10. audit the serialized repairs on disjoint final subsets and reveal private family labels only for scoring;
11. replay all exact task semantics, compiler decisions, serialized repairs, and metrics.

The four task files contain 13,600 exact records in total.

88.5 Final obstruction-to-repair result

The pre-audit suite passes all 24 episodes. The final audit contains 36 episodes: two foreign recurrent architectures, six hidden obstruction families, and three unseen transform instances per architecture–family pair. The confusion matrix is exactly diagonal.

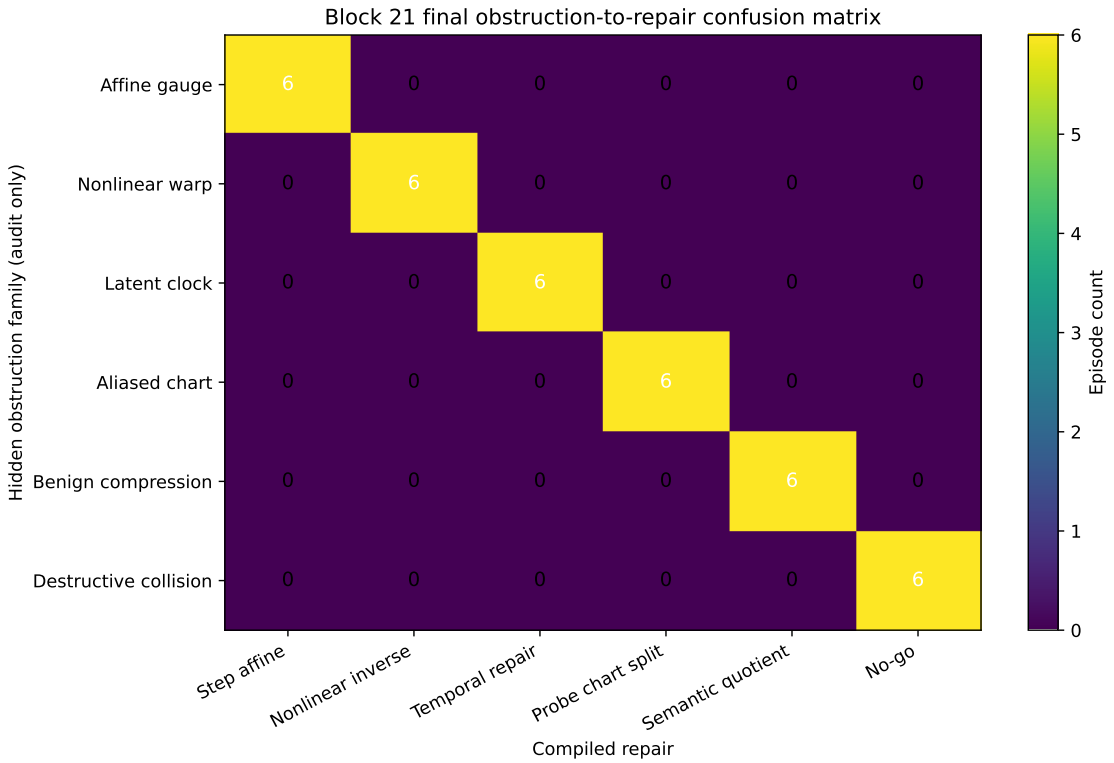


Figure 87: Final obstruction-to-repair matrix. Family labels are unavailable during compilation and are revealed only to the audit harness.

Table 56: Fresh final performance of the selected repair, aggregated over three unseen transform instances. Packet quantities do not apply to the semantic quotient.

Architecture and family	Mean sem. R^2	Min sem. R^2	Mean packet R^2	Max NRMSE
GRU affine gauge	0.763002	0.743007	0.999075	0.033155
GRU nonlinear warp	0.730775	0.723749	0.999089	0.031610
GRU latent clock	0.756245	0.718695	0.998925	0.034543
GRU chart alias	0.761092	0.737828	0.998973	0.032485
GRU benign compression	0.999991	0.999990	—	—
Residual affine gauge	0.745807	0.727216	0.999084	0.031363
Residual nonlinear warp	0.746279	0.727708	0.999132	0.031875
Residual latent clock	0.772349	0.758672	0.999100	0.030772
Residual chart alias	0.742217	0.722507	0.999192	0.029533
Residual benign compression	0.999990	0.999989	—	—

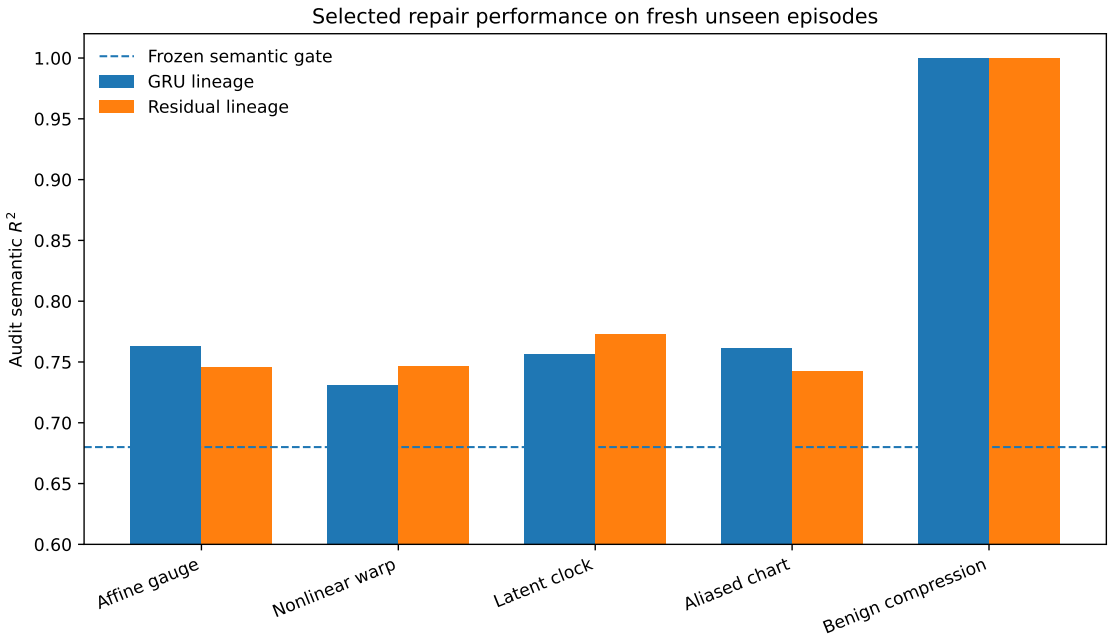


Figure 88: Protected semantic performance on fresh final episodes. Every selected non-destructive repair remains above the frozen semantic gate.

88.6 Wrong-repair and no-go controls

For nonlinear, temporal, and chart-alias episodes, the inherited step-affine map is retained as a wrong-repair control. The paired effect is

$$\Delta_i = \ell_{\text{step affine,i}} - \ell_{\text{selected,i}}. \tag{1287}$$

Every frozen bootstrap interval is strictly positive.

Table 57: Paired semantic-loss advantage of the selected structured repair over the inherited step-affine control.

Architecture and obstruction	Mean effect	Frozen 95% interval	Positive fraction
GRU chart alias	0.464544	[0.427997, 0.503210]	0.944792
GRU latent clock	0.675134	[0.632984, 0.716193]	0.963542
GRU nonlinear warp	0.198562	[0.183050, 0.213400]	0.937500
Residual chart alias	0.568441	[0.523956, 0.616695]	0.971875
Residual latent clock	0.603810	[0.566393, 0.637773]	0.969792
Residual nonlinear warp	0.175636	[0.159888, 0.192203]	0.887500

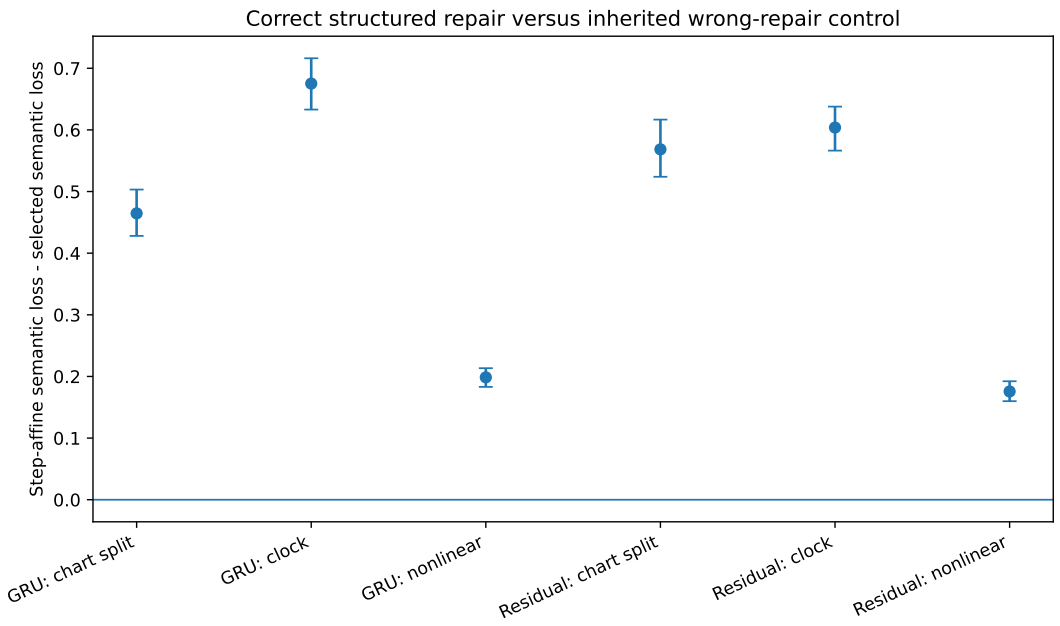


Figure 89: Correct structured repairs defeat the inherited step-affine wrong-repair control on all six architecture–obstruction comparisons.

In the six destructive final episodes, the compiler independently discovers between 112 and 114 exact collision pairs. Mean protected semantic separation at identical observations lies between 2.053524 and 2.158440. The best available semantic quotient achieves audit R^2 between 0.609997 and 0.644296, below the frozen 0.68 gate. Every destructive episode therefore returns no-go.

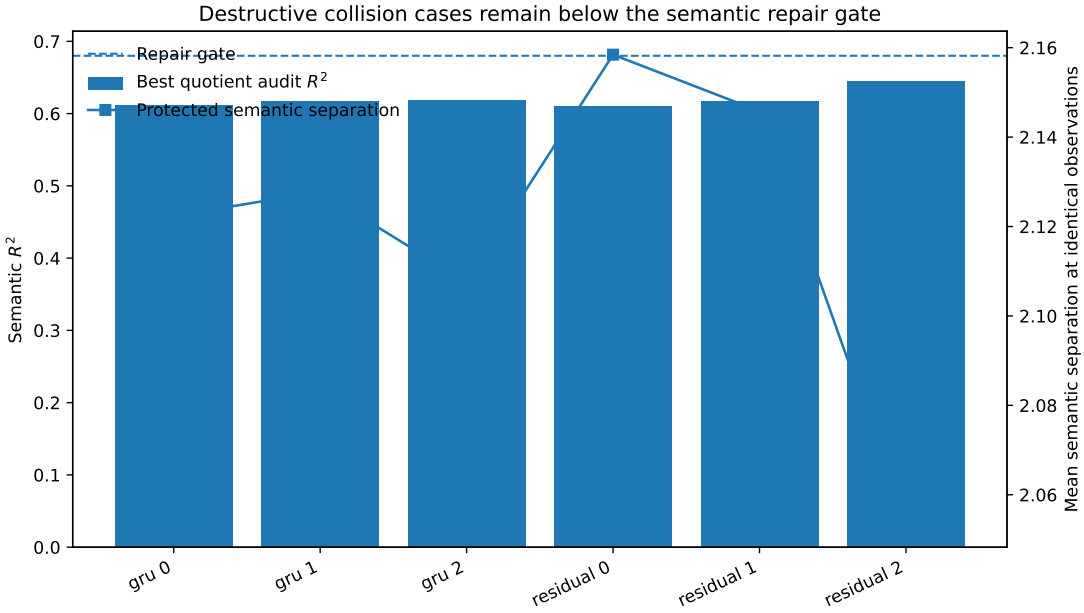


Figure 90: No-go controls. Correlational quotient fitting remains below the repair gate while identical observations exhibit substantial protected semantic disagreement.

88.7 Finite realization theorem

Conditional Theorem 88.10 (Block 21 obstruction-guided transport-grammar growth)**.** *Within the frozen Block 21 environment, assume that:*

1. *the accepted Block 20 receipt and inherited connection replay exactly;*
2. *packet projection and semantic-probe fitting use only their declared pre-audit supports;*
3. *the compiler receives the public label-isolated episode view and does not import the audit-only expected map;*
4. *the repair grammar and all numerical gates are frozen before final-task generation;*
5. *final transform seeds and task records are absent from compiler construction;*
6. *every accepted reconstructive or quotient repair passes its declared protected audit gates;*
7. *every no-go output possesses the declared repeated-observation semantic-collision certificate;*
8. *the final confusion matrix, wrong-repair controls, serialization checks, and receipt chain replay as recorded.*

Then the accepted edge is a finite, conservative, obstruction-guided extension of the Block 20 Neuralese transport grammar. It distinguishes the six declared hidden stress families by repair response, retains the inherited minimal constructor on ordinary affine drift, admits structured extensions only where required, permits a semantic quotient under benign dimension loss, and abstains under certified destructive semantic loss.

Proof. Items 1 and 2 bind the experiment to the inherited lineage and protected witness. Items 3 through 5 establish label isolation, chronology, and a frozen finite opportunity set. Items 6 and 7 separate successful transport, protected quotienting, and impossible target-only repair. Item 8 supplies foreign held-out evidence and causal controls. The exact inclusion of the step-affine constructor preserves the inherited grammar, while the minimum-description rule prevents unnecessary structural expansion. The resulting transaction therefore changes the repair language only in response to a validated obstruction inside the declared fixture. □

Block 21 result

The compiler passes all 24 pre-audit episodes and all 36 fresh final episodes. Its final obstruction-to-repair matrix is diagonal. Reconstructive repairs retain packet $R^2 \geq 0.998807$ with NRMSE at most 0.034543; semantic quotients retain semantic $R^2 \geq 0.999989$; every structured repair defeats the inherited wrong-repair control with a strictly positive frozen interval; and every destructive collision produces no-go. Nineteen automated tests and a deterministic replay verifier reproduce all 13,600 exact task records and all final metrics.

88.8 Claim boundary and next threshold

Block 21 is a controlled finite realization. It does not establish unrestricted obstruction discovery, arbitrary nonlinear transport, unrestricted variable-length or stochastic trajectory alignment, unpaired transport, autonomous semantic-ontology invention, natural-language Neuralese interpretation,

production-model transfer, external independent replication, open-ended recursive improvement, unrestricted safety, consciousness, or sentience.

The benchmark families and candidate constructors are author supplied. The benign quotient explicitly exposes a lower-dimensional sufficient statistic, while the destructive fixture deliberately withholds protected coordinates and inserts exact collisions. These are positive and negative controls for the compiler logic, not evidence that the same structures will be straightforward to identify in a pretrained language model.

The next major threshold is to remove exact source–target task pairing and reduce dependence on a supplied semantic witness. A stronger descendant should align foreign Neuralese trajectories using intervention orbits, sparse shared landmarks, output-equivalence classes, and relative semantic geometry; determine when those observations identify a transport only up to gauge; and issue an underidentification certificate when they do not.

89 Integrated SRIC Architecture

89.1 Functional loop

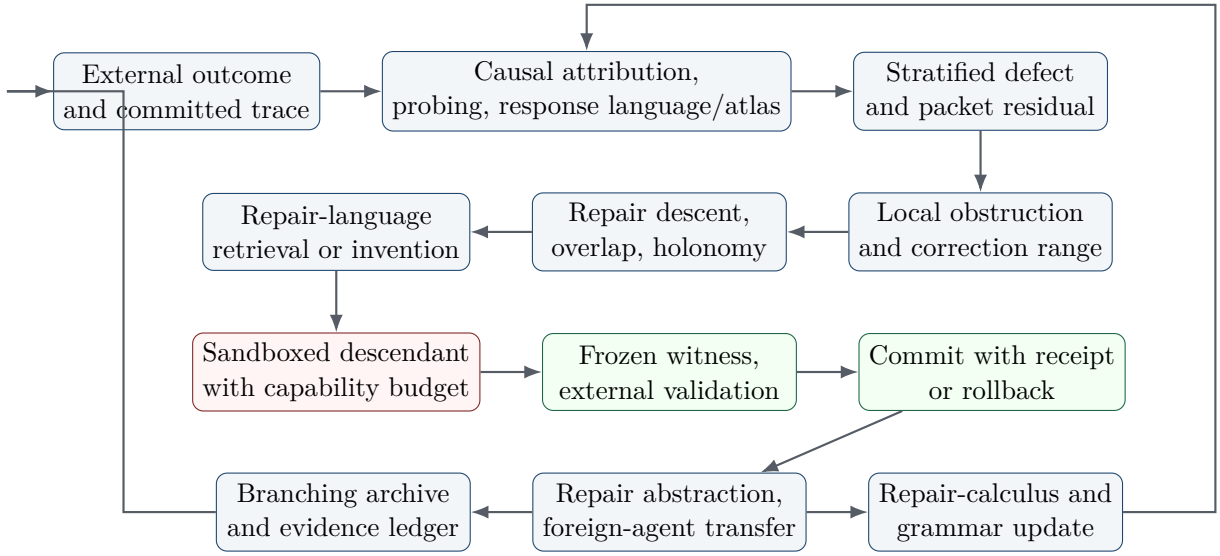


Figure 91: The SRIC improvement loop. Controlled probes may first recover a functional response atlas before defects and repairs are localized. The witness path is separated from candidate construction. Successful repairs become reusable schemas, while valid obstructions and identifiability failures become reusable no-go certificates.

89.2 One transaction

Protocol 89.1 (Proof-carrying improvement transaction)**.** Given lineage state $\mathfrak{L}_t$:

1. Freeze witness W_t, evaluator core coordinates, evidence ledger head, resource budget, and rollback checkpoint.

2. Detect a persistent externally grounded mismatch.

3. Generate internal-change, world-change, sensor, memory, evaluator, copy, fork, and unknown hypotheses.
4. Select bounded active probes until uncertainty is below threshold or the probe budget is exhausted.
5. Estimate intervention-response operators and either replay a validated atlas or recover an endogenous functional cover with response-fit, commutator, spectral-gap, and identifiability certificates.
6. Resolve the defect into the validated local strata and compute local obstruction classes.
7. If a local obstruction is nonzero, record a no-go certificate and stop or extend the grammar.
8. Construct local repairs, compute overlap differences, and reject if the repair-descent class is nonzero.
9. Retrieve or construct a typed repair schema and estimate transport, leakage, and holonomy margins.
10. Precommit predicted gain and affected invariants.
11. Instantiate a sandboxed descendant with bounded capabilities.
12. Reveal external evaluation traces to the frozen witness.
13. Commit only if validation, provenance, regression, core-preservation, and rollback tests pass; otherwise restore the parent.
14. Abstract the result into a repair schema or obstruction certificate.
15. Test the abstraction on foreign held-out agents before adding it to the reusable repair operad.
16. Estimate remaining module-specific headroom and record any saturation certificate.
17. If current modules are saturated while persistent obstruction remains, issue a grammar-boundary receipt and begin conservative conflict refinement.
18. At the research-compiler level, test whether one diagnostic token contains incompatible sets of acceptable methods. If a conflict appears, search a declared generic experiment and constructor meta-language. Admit only a sealed conservative extension that improves foreign research problems.
19. At the Neuralese transport level, distinguish coordinate drift, nonlinear warp, latent-clock mismatch, chart aliasing, benign semantic quotient, and destructive information loss. Select the minimum passing constructor and issue no-go when protected semantics are not measurable from the target packet.

90 Empirical Status and Remaining Program

90.1 Realized in Block 5

The released Block 5 reference package realizes the following finite and synthetic components:

1. the four-case local obstruction, gluing obstruction, flat repair, and evaluator-holonomy fixture;
2. universal, controlled, and productive closure checks on a finite transaction graph;
3. a witness-separated transaction kernel with commit, rollback, canonical receipts, and hash chaining;
4. a matched-model anonymous-change environment with bounded probes and abstention;
5. a matched-budget foreign-agent benchmark with static, search, packet, causal, transport, witness, and engineered meta positive-control arms;
6. a reproducible test, result, and manifest pipeline.

These realizations support the mechanical and finite-semantic gates defined in Section 16. The anonymous and benchmark suites remain engineered positive controls rather than evidence of autonomous repair discovery.

90.2 Realized in Block 6

Block 6 replaces analytic hidden-coordinate transport by a controlled learned-representation problem. Its realized components are:

1. a causally defined hidden packet estimated from source intervention differences;
2. equal-rank PCA, random-packet, wrong-block, direct-copy, and shuffled-anchor controls;
3. a frozen packet-local source repair fitted without target audit labels;
4. a source-to-target bridge estimated from paired hidden activations while the constructing target map remains withheld;
5. seven target representation families spanning injective, overcomplete, ill-conditioned, compressed, mildly nonlinear, saturated, and piecewise-linear regimes;
6. independently trained linear and tanh encoders with distinct random initializations;
7. a frozen witness gate that can reject a blind target gain when alignment, protected competence, or unaffected-stratum regression is unacceptable;
8. a release-frozen package with disjoint train, anchor, calibration, and audit splits, five automated tests, a 34-file manifest, frozen result objects, and reproducible figures.

The controlled package passes the cumulative foreign-transfer gate G_{foreign} for one shared task and defect grammar. It does not independently establish metaproductivity or grammar expansion.

90.3 Realized in Block 7

Block 7 implements two separately sealed parent–child comparisons of repair-process components.

Selection-only realization. The first route holds the ordered five-candidate repair bundle fixed on every target and changes only the commit policy. It realizes:

1. a frozen conservative parent selector;

2. a lineage-generated dual-head child selector fitted from development repair receipts;
3. a finite meta-grammar selected on a disjoint meta-validation split;
4. a descendant receipt sealed before either final audit is opened;
5. paired seed/task-foreign and architecture-foreign audits;
6. raw-calibration, no-risk, shuffled-lineage, selector-ablation, and audit-oracle controls;
7. identical non-null candidate budgets for every admissible selector arm.

The child-parent protected-utility effects are

$$\widehat{\Delta}_{\text{audit}} = 0.072518, \qquad \widehat{\Delta}_{\text{arch}} = 0.104999, \tag{1288}$$

with positive frozen paired bootstrap intervals. Because parent and child share the same candidate set, this establishes behavioral selector metaproductivity but not expansion of the authorized correction image.

Bridge-policy realization. The second route holds the source repair, bridge family, transport formula, protected evaluator, and 24-query deployment budget fixed while changing the paired-anchor selection rule and bridge-identification gate. It realizes:

1. a random-16 parent anchor policy;
2. a compiled pivoted-QR-16 descendant policy sealed before primary foreign targets are generated;
3. disjoint development and audit task families and disjoint primary representation families;
4. a 144-target paired foreign audit and a separate stress suite;
5. random-budget, farthest-point, shuffled-binding, sign-flip, and plain-QR controls;
6. exact query ledgers, rank diagnostics, harm checks, and package-closed receipts.

On the primary foreign audit, mean balanced selected gain rises from 0.135212 to 0.344720, and

$$\widehat{\Delta}_{\text{bridge}} = 0.209508, \qquad \text{CI}_{95\%} = [0.194968, 0.223898]. \tag{1289}$$

Every primary paired effect is positive, full-rank bridge identification rises from 0.03125 to 1, and harmful selected targets remain zero. The plain pivoted-QR control is effectively identical, so the result supports rank-revealing query design rather than a packet-specific advantage.

The controlled packages therefore pass two distinct level-5 gates: $G_{\text{meta}}^{\text{select}}$ for selector metaproductivity and $G_{\text{meta}}^{\text{bridge}}$ for bridge-policy metaproductivity. Neither package establishes repeated-generation closure, autonomous candidate invention, or open-set grammar expansion.

90.4 Realized in Block 8

Block 8 is a formal and symbolic-realization expansion rather than a new learned-agent audit. It realizes:

1. finite restricted oracle envelopes and exact permutation-average meta-headroom allocation;
2. component saturation and headroom-adjusted metaproductivity;

3. bottleneck migration with an explicit conservative grammar-discovery coordinate;
4. nested obstruction images and exact newly correctable energy;
5. repair death versus forgetting death under dynamic defect semantics;
6. obstruction-guided one-generator and rank-k grammar expansion;
7. finite repair-conflict refinement with a nuisance-feature negative control;
8. finite counterexample-guided repair synthesis or grammar exhaustion;
9. intervention-response algebra decomposition into central causal strata;
10. lineage-wide false-improvement and protected-regression budget accounting.

The symbolic package passes eight automated tests and produces a package-closed result object and four figures. These fixtures validate the finite mathematics and its implementation. They do not advance the empirical claim ladder beyond the two one-step Block 7 metaproductivity results.

90.5 Realized in Block 9

Block 9 replays the exact Block 7 bridge child and compiles a fresh grandchild in a new heteroscedastic bridge environment. It realizes:

1. exact replay and digest verification of the Block 7 rank-revealing child policy;
2. fresh development and audit task supports and disjoint primary representation families;
3. a rank-saturation diagnosis in which every inherited primary design is full rank but its weighted information floor remains low;
4. a finite conditioning-aware policy grammar containing packet QR, unweighted D-optimal, quality-weighted D-optimal, quality-weighted A-optimal, and quality-weighted E-refined designs;
5. a quality-weighted D-optimal grandchild and a stronger recalibrated rank-child control;
6. identical 24-query deployment budgets for the primary child and grandchild;
7. gate-only, quality-permutation, unweighted-design, random-16, and larger-budget random-32 controls;
8. a 96-target paired primary foreign audit plus compressed, nonlinear, and severe-ill-conditioning stress families;
9. exact ancestor, compiler, pre-audit seal, result, query-ledger, and package-manifest receipts.

On the fresh primary audit, mean selected protected gain rises from 0.037393 for the recalibrated rank child to 0.209580 for the conditioning grandchild. The paired effect is

$$\widehat{\Delta}_{1\to 2} = 0.172188, \qquad \mathrm{CI}_{95\%} = [0.141481, 0.203110]. \tag{1290}$$

Both designs have full-rank rate one. The mean weighted minimum information eigenvalue rises from 0.518197 to 14.268195, the median condition number falls from 19393.296 to 544.180, and clean transport NRMSE falls from 0.303981 to 0.095803. The gate-only and quality-permutation controls do not reproduce the selected gain. The release therefore passes a context-indexed two-edge

repeated-lineage gate. It does not establish structural repair-grammar growth or improvement of the outer compiler.

90.6 Realized in Block 10

Block 10 retains the Block 9 quality-weighted acquisition mechanism but expands the representational grammar of repair transport. It realizes:

1. exact semantic replay of the Block 9 conditioning-aware ancestor;
2. a strengthened global-affine parent using 80 fit plus 16 witness pairs;
3. a conservative two-chart child using 48 fit, 32 chart-selection, and 16 witness pairs;
4. one matched 96-query deployment envelope for both processes;
5. a finite separator search restricted to the source repair's protected two-dimensional context packet;
6. a 54-target development grammar-inadequacy certificate with full source rank, a persistent directly supervised global-affine residual, a near-exact true-chart oracle, smaller learned local residual, positive chart contrast, and an exact parent embedding;
7. disjoint development, meta-validation, and primary audit task and family identifiers;
8. a 96-target primary foreign audit and a 32-target stress suite;
9. direct global-affine oracle, global-blind, local-blind, gate-only, random-gate, shuffled-binding, and true-chart controls;
10. exact ancestor, compiler, grammar-certificate, pre-audit seal, result, test, figure, and package-manifest receipts.

The resource-matched global parent abstains on the primary audit. The chart child obtains mean selected gain 0.334224, median 0.284265, commit rate 0.572917, and zero harmful selected targets. The child-parent effect is

$$\widehat{\Delta}_{G\to C} = 0.334224, \qquad \mathrm{CI}_{95\%} = [0.266459, 0.401903]. \tag{1291}$$

The sharper child-minus-direct-global-oracle effect is

$$\widehat{\Delta}_{\mathrm{oracle}\to C} = 0.157530, \qquad \mathrm{CI}_{95\%} = [0.080863, 0.234135]. \tag{1292}$$

All primary source designs are full rank. Mean global validation NRMSE is 0.598513, mean local validation NRMSE is 0.064975, and mean learned gate accuracy is 0.874854. Gate-only, random-gate, and shuffled-binding controls are strongly negative. Blind local application has higher mean gain but a nonzero harmful-commit rate, so the witness performs substantive selection. This passes the first finite structural repair-grammar gate. It does not establish autonomous packet, feature-language, chart-number, or compiler discovery.

90.7 Current empirical interpretation

The progression now separates seven empirical achievements and one formal bridge:

1. Block 5 realizes finite transactions, controls, and synthetic mechanics;
2. Block 6 realizes intervention-defined hidden packets and learned repair transport to foreign representations;
3. Block 7 realizes sealed one-step improvements of repair selection and rank-revealing evidence acquisition;
4. Block 8 formalizes meta-headroom, saturation, bottleneck migration, persistent obstruction, grammar refinement, causal-atlas discovery, and sequential lineage certification;
5. Block 9 replays the Block 7 child, diagnoses rank saturation, and realizes a second sealed edge by migrating to heteroscedastic conditioning-aware query design;
6. Block 10 holds that acquired query policy fixed, certifies that a global-affine repair family remains structurally obstructed, and realizes a conservative two-chart repair-transport extension;
7. the directly supervised global-affine oracle comparison separates function-class growth from a weak global estimator or permissive witness.

The strongest admissible statement is:

Within declared finite transport contexts, a proof-carrying lineage realizes two sequential query-policy improvements followed by one structural repair-grammar expansion. The final descendant retains conditioning-aware acquisition, proves a persistent residual for the global-affine repair family, conservatively adds an invariant-context chart-contrast direction, and improves fresh foreign targets under matched total evidence while preserving the global process as a fallback.

The third edge crosses the boundary that Block 7 could not cross. The authorized opportunity set changes: two context-bound target repairs become reachable where the parent can instantiate only one global affine repair. The child also exceeds a directly supervised global-affine ceiling comparison. This is finite structural metaproductivity. It remains externally scaffolded because the protected packet, separator family, chart count, source repair, evaluator, witness, and outer compiler are supplied.

90.8 Realized in Block 11

Block 11 removes the protected context packet supplied to Block 10 and compiles a target-specific atlas from intervention responses. It realizes:

1. exact replay of the Block 10 ancestor policy and pre-audit lineage records;
2. a six-dimensional source with three hidden rank-two functional blocks under target-specific rotations or injective embeddings;
3. two controlled response operators with separating block signatures in the primary families;
4. central spectral block recovery up to permutation and within-block gauge;
5. response-fit, commutator, spectral-gap, commutant-dimension, and hidden-overlap audit diagnostics;
6. repair-relevant block labeling through source-repair invariance, target-gate agreement, local-bridge validation, chart balance, and protected regression;

7. a conservative executable embedding of the Block 10 supplied-packet grammar;
8. one equal 120-query envelope for the inherited fixed-packet parent and endogenous-atlas child;
9. legacy-packet, PCA, full-space, random-atlas, shuffled-response, global, blind, and true-chart controls;
10. an 80-target primary foreign audit and a 25-target stress suite containing response degeneracy, strong coupling, smooth transitions, repair-dependent chart crossing, and compression;
11. exact compiler, atlas-certificate, pre-audit seal, result, figure, test, and 49-file manifest receipts.

The primary parent obtains zero selected gain and the child obtains 0.207325. The paired effect has frozen interval $[0.131902, 0.287910]$, no target favors the parent, and selected harm is zero. Blind child application has larger mean gain 0.271529 but harm rate 0.1875, so the witness performs substantive selection. The release passes a finite endogenous-atlas gate. It does not establish autonomous intervention invention, unrestricted block-count discovery, general noncommuting or nonlinear atlas recovery, or compiler self-improvement.

90.9 Realized in Block 12

Block 12 realizes two separately sealed structural edges beyond endogenous atlas recovery.

Response-language edge. The inherited two-operator response algebra contains an exact context–repair alias. Every polynomial and composition of inherited responses preserves the merged block. A finite intervention compiler selects one separating operator from five typed candidates, conservatively retains the old algebra, and seals the new response policy before a 96-target foreign audit. Under equal 144-query deployment budgets, selected protected gain rises from 0 to 0.846985, with frozen interval $[0.786100, 0.901073]$, no negative paired effects, and zero selected harm.

Repair-constructor edge. After the response child identifies the repair packet, the inherited affine repair operad encounters a target correction with a nonzero quadratic Hessian. The child adds symmetric degree-two packet features while embedding every affine repair by zero quadratic coefficients. Under equal 160-query deployment budgets, selected gain rises from 0.289123 to 0.997511. The paired effect is 0.708388 with interval $[0.659635, 0.757198]$, and the child exceeds a directly supervised 1600-pair affine oracle by 0.530933. A full-space quadratic control is equally effective, so the supported claim is degree-two constructor growth rather than unique block-local superiority.

These edges pass finite response-language and candidate-constructor gates. They do not establish unrestricted intervention invention, open-ended program synthesis, or improvement of the outer lineage compiler.

90.10 Realized in Block 13

Block 13 changes the research compiler while holding the downstream candidate-method grammar and final evaluator fixed. It realizes:

1. a strong parent compiler with structured literature priors, a shared finding graph, results-free receipts, exact code/specification binding, balanced exploration, multi-benchmark selection, capability gates, and integrity monitoring;

2. a child compiler trained only from development receipts, with a nearest-centroid obstruction classifier and ridge outcome/risk forecasters;
3. an identical 39-method candidate grammar and identical six-full-experiment budget for parent and child;
4. development, meta-validation, primary foreign, and architecture-foreign problem splits with pre-audit compiler sealing;
5. shuffled-archive, classifier-only, forecaster-only, no-forum, single-benchmark, no-monitor, random-allocation, and audit-oracle controls;
6. exact proposal, code, archive, compiler, allocation, evaluator, monitor, result, test, and manifest receipts.

On 96 primary foreign problems, mean validity-filtered protected gain rises from 0.883004 to 0.950231, with paired effect 0.067227 and frozen interval $[0.035881, 0.103274]$. On 72 architecture-foreign problems, gain rises from 0.772566 to 0.863367, with paired effect 0.090800 and interval $[0.052311, 0.135081]$. The obstruction classifier is the dominant realized mechanism. Outcome forecasting is informative but contributes only modestly, and removal of the finite within-problem forum has no measurable effect. The release passes a bounded research-compiler metaproductivity gate. It does not establish open-ended automated science, autonomous candidate-language growth, evaluator invention, or improvement of an unrestricted outer compiler.

90.11 Realized in Block 14

Block 14 replays the Block 13 result digest and exact method language, then recalibrates the inherited compiler design as the parent of an adversarial robustness study. It realizes:

1. diagnostic-matched clean and ambiguous problems that weaken one-shot obstruction classification;
2. sparse two-obstruction mixtures and an unknown class outside the inherited response cone;
3. evaluator-attack and integrity-attack method families;
4. an 18% contaminated development archive plus a contamination sweep through 35%;
5. a robust child using sequential diagnosis, sparse response mixtures, open-set routing, robust archive summaries, evaluator portfolios, and an integrity-monitor ensemble;
6. the exact inherited 39-method grammar and six-full-experiment budget for parent and child;
7. primary and independently shifted generator-foreign audits with pre-audit compiler sealing;
8. no-open-set, no-sequential, ordinary-archive, simple-monitor, single-portfolio, random-allocation, and audit-oracle controls;
9. exact results-free and post-result receipt chains, canonical results, tests, figures, and package manifest.

On 144 primary problems, mean validity-filtered protected gain rises from 0.410813 to 0.575783, with paired effect 0.164970 and frozen interval $[0.115591, 0.218655]$. On 120 generator-foreign problems, gain rises from 0.389514 to 0.550559, with effect 0.161045 and interval $[0.105335, 0.223141]$. Primary

lower-tail gain changes from -0.133065 to 0.271516, observed invalid selections change from 0.027778 to 0, and false confident known-class assignment on unknown problems changes from 1 to 0.083333. Sequential response updating and open-set routing are substantively supported. The simple-monitor ablation is null and the single-portfolio ablation is nearly null, so neither receives independent positive attribution.

The claim remains finite. The child uses an externally declared response family, candidate grammar, attack vocabulary, evaluator portfolio, and monitor ensemble. It is a robustness improvement of the fixed-language compiler, not unrestricted open-set science or compiler-language growth.

90.12 Realized in Block 15

Block 15 replays the Block 14 result digest and promotes the robust compiler to the inherited parent of a structural compiler-language study. It realizes:

1. six inherited obstruction tokens, one undifferentiated unknown token, and the inherited 39-method singleton grammar;
2. a development open-set cell containing two hidden research mechanisms that share the inherited diagnostic representation but require incompatible high-gain methods;
3. a generic six-probe research-experiment language and a generic 30-member ordered pair-method constructor grammar;
4. finite conflict scoring, anonymous token refinement, pair-constructor synthesis, and exact conservative fallback;
5. hidden novel labels excluded from compiler construction and retained only for audit interpretation;
6. identical eight-unit deployment research budgets for parent and child;
7. primary and generator-foreign audits sealed after the language certificate and compiler object;
8. taxonomy-only, grammar-only, shuffled-binding, inherited-oracle, and expanded-oracle controls;
9. complete results-free and post-result receipts, deterministic figures, tests, and a package manifest.

The development compiler selects `interaction-contrast` and binds the anonymous refined tokens to `evaluator-then-resource` and `objective-then-constructor`. The inherited development oracle obtains 0.377257 mean open-set gain, while the selected extension obtains 0.948962, producing a newly correctable margin 0.571704.

On 96 primary problems, mean protected gain rises from 0.633969 to 0.859731, with paired effect 0.225762 and interval $[0.166708, 0.286431]$. On 82 generator-foreign problems, gain rises from 0.574224 to 0.814520, with paired effect 0.240296 and interval $[0.180154, 0.302701]$. No paired problem favors the parent. Known-class behavior is reproduced exactly through the inherited fallback. The taxonomy-only and shuffled-binding controls match the parent, while the grammar-only arm solves only the novel cell favored by its pooled binding. The release passes a finite compiler-language growth gate.

The result remains relative to an externally declared generic meta-language. The compiler does not invent arbitrary experiments or programs, semantically name the new obstruction classes, improve the evaluator or monitor, or execute real model-training research.

90.13 Stage H completed: first repeated lineage

The realized lineage is

$$K_0 \xrightarrow{\Pi_1} K_1 \xrightarrow{\Pi_2} K_2. \tag{1293}$$

The first edge improves rank identification. The second edge does not attempt to improve a saturated rank endpoint. It migrates to a distinct information-conditioning obstruction. Exact ancestor replay, new development evidence, a pre-audit grandchild seal, matched deployment queries, a recalibrated child control, and fresh foreign audit systems distinguish this result from reusing the Block 7 final evidence or weakening the parent.

A two-edge result is still finite. It does not imply a winning strategy for infinitely repeated productive closure, compiler self-improvement, or open-set grammar growth. It establishes that the lineage mechanism can produce a second successful descendant after the first improvement changes the active bottleneck.

90.14 Stage I partially realized: finite structural grammar expansion

Block 10 realizes the finite form of this stage. The lineage encounters foreign systems for which one global affine repair remains misspecified despite full-rank, quality-aware acquisition and an 80-anchor parent fit. It records a grammar-inadequacy certificate, conservatively introduces an invariant-context two-chart constructor, and verifies a new anti-diagonal repair direction on fresh foreign systems. The stronger directly supervised global-affine oracle establishes that the gain is not explained by poor source-to-target bridge estimation alone.

Open-set grammar expansion additionally requires:

1. autonomous detection of a persistent obstruction or repair conflict;
2. invention or discovery of the separating feature, chart count, or nonlinear coordinate system;
3. a conservative embedding of old grammar, proofs, evidence, and authorization decisions;
4. exposure of a genuinely new correction direction under exact causal ablation;
5. held-out obstruction-energy reduction on foreign systems that require the new distinction;
6. preservation of evaluator, provenance, witness, and resource commitments;
7. repeated structural descendants under lineage-wide statistical and semantic-debt budgets.

Block 10 supplies the obstruction, conservative extension, structural contrast, matched-resource audit, and causal controls. It does not by itself supply atlas discovery; Block 11 addresses one finite response-algebra form of that next requirement.

90.15 Stage J partially realized: finite endogenous atlas growth

Block 11 realizes the finite form of endogenous atlas growth. The supplied context packet is removed. A declared intervention family generates response operators whose central spectral blocks are recovered up to permutation and within-block gauge. Repair-validation evidence selects the block that supports a stable chart separator. The old supplied-packet process remains an executable subgrammar and fallback.

The unresolved part of Stage J is substantial. The intervention family, block dimensions, finite block-scalar response model, hard binary chart grammar, separator family, source repair, evaluator, witness, and outer compiler remain supplied. A stronger lineage must discover which intervention to run, whether signatures are degenerate, whether response operators are noncommuting, and whether the appropriate extension is another discrete block, a soft manifold, a kernel lift, or an information-loss certificate.

90.16 Stage K completed in finite form: response-language and candidate-constructor growth

Block 12 realizes the finite form of Stage K through two distinct contexts. The first edge certifies an exact response alias under the inherited algebra, selects one separating intervention from a fixed typed menu, and recovers a repair-enabling central refinement. The second edge certifies that every finite composition of inherited affine constructors remains affine while the target repair has a nonzero quadratic Hessian, then conservatively adds a degree-two constructor. Both results use matched query budgets, independent parent and child gates, fresh sealed audits, explicit function-class controls, and package-closed receipts.

The unresolved open-set form is stronger. A future lineage must enlarge the intervention or constructor language without receiving the final candidate menu, account for acquisition feasibility and physical effects, determine the necessary polynomial or program degree, and show that its outer diagnostic compiler transfers to foreign parent repair processes.

90.17 Stage L partially realized: finite research-compiler metaproductivity

Block 13 realizes a bounded improvement of the research-allocation compiler while the candidate-method language, evaluator portfolio, integrity monitor, and final research interfaces remain fixed. The child uses the same research budget more selectively by diagnosing the active obstruction and forecasting outcome and risk. This is an improvement of the process that generates and chooses repair-process descendants, but not an unrestricted improvement of its own language.

The unresolved form requires compiler grammar growth and eventually conservative evaluator evolution. A later compiler must detect that its obstruction taxonomy, proposal features, experimental actions, monitor, or evaluator portfolio is inadequate, introduce a new research distinction, preserve the path-flat evidential core, replay old protected decisions, and be witnessed by a component not rewritten in the same transaction. The primary danger is semantic capture through simultaneous alteration of proposal, evidence, utility, and approval.

90.18 Stage M completed in finite form: adversarial open-set compiler robustness

Stage M asks whether the finite research compiler remains useful after the clean six-way routing problem is replaced by ambiguous diagnostics, mixed obstructions, unknown cases, contaminated archives, evaluator attacks, integrity attacks, and a generator shift. Block 14 passes this finite gate. The child improves the inherited parent in mean and lower-tail validity-filtered gain under the same six-experiment budget, and every declared child group has positive mean gain. Some individual groupwise paired intervals include zero, so the aggregate and lower-tail claims are stronger than every family-specific claim.

The result does not yet satisfy the stronger external-robustness target. The benchmark generator, candidate methods, response language, attack types, evaluator portfolio, and monitor ensemble are still part of one package-closed research design. The next stage should combine an independently written generator and implementation with executable small-model or program-synthesis research, then require the robust compiler to issue a compiler-language inadequacy certificate before adding a new diagnostic, experiment, candidate family, or evaluator dimension.

90.19 Stage N completed in finite form: compiler-language growth

Stage N asks whether the robust research compiler can move beyond abstention or generic routing when its own diagnostic and method language is certified inadequate. Block 15 passes one finite form of this gate. One inherited open-set token contains development problems with incompatible acceptable-method sets. A finite compiler selects a separating probe from a generic experiment language, creates two anonymous refined tokens, synthesizes one ordered pair method for each token, and preserves the entire inherited compiler as an exact fallback.

This is structurally stronger than Block 14. The child does not merely classify known obstruction labels more robustly or allocate the inherited method grammar more effectively. It adds diagnostic and method expressions absent from the deployable parent language. The unresolved form is unrestricted compiler-language growth: the generic probe family, pair-composition operator, method primitives, evaluator, integrity monitor, and research simulator remain externally supplied.

90.20 Stage O direct integration completed: Neuralese causal semantics

Block 16 makes recurrent latent reasoning an explicit SRIC subsystem rather than an analogy. The latent transition, task-anchored update gate, halt map, trace convention, intervention language, semantic transport, and answer witness are separately typed and receipted. The mandatory audit distinguishes five possibilities: an inactive loop, an unordered loop, a corruption-fragile loop, a useful ordered but task-nonspecific loop, and a task-specific loop. Cross-task transplantation within fixed answer-format groups is the principal discriminator between the last two.

The companion validates the audit logic on a sealed synthetic parent–child pair. The parent passes necessity, order, corruption, and replay but fails task specificity, so its status is PARTIAL. The child raises structural-normalized specificity from 0.016952 to 0.334610, obtains a paired transplant-cost gain of 0.439986 with interval $[0.386884, 0.493143]$, preserves clean performance, and passes both the FULL causal and adaptive-halting gates. This is an engineered positive control, not evidence about a trained language model.

The local Hyper adapter is executable when the official runtime and weights are supplied. It imports the exposed injection hook, captures and reinjects exact latent states, computes gold-suffix cross-entropy through the official end-of-thought path, and emits canonical receipts. The actual model was not present in the build environment, so the real-model result remains open. The next empirical threshold is a sealed Hyper audit followed by a genuine parent–child checkpoint comparison generated by a declared training or search process.

90.21 Stage P completed in finite form: executable Neuralese binding repair

Block 17 realizes the two-checkpoint threshold proposed at the end of Block 16, but on a package-local recurrent micro-model rather than Hyper. A separately implemented exact rational environment

generates affine programs and protected (A, B, y) witnesses. A task-blind trained parent is inherited into six finite repair candidates. The compiler uses only development, meta-validation, and probe-calibration evidence, seals the selected checkpoint, and then faces fresh primary and shifted-generator tasks.

The first sealed attempt is negative: the selected child improves absolute transplant effects and semantic decodability but misses the frozen normalized FULL gates. The package retains that failure and commits a revision before new final-task generation. In the second round, three candidates are eligible and the full-binding child is selected. The parent remains structurally active but program blind. The child passes necessity, order, noise, replay, program specificity, answer specificity, local semantic continuity, donor coverage, external trace decoding, and adaptive-depth gates on both final supports. The accepted receipt is `6cc2257f...655e4ae`.

This closes one finite training-generated latent-binding edge. It does not establish foreign-model metaproductivity.

The affine task family, semantic witness, donor taxonomy, candidate ranks, loss, compiler score, and gate thresholds remain external. The decisive next test is an independently trained latent-reasoning architecture or a real parent–child production checkpoint pair, preferably with a compiler-generated modification absent from the Block 17 grammar.

90.22 Stage Q attempted and obstruction-certified: cross-implementation Neuralese transport

Block 18 instantiates the source-selected progressive FiLM schema in two independently implemented recurrent cell families. On the primary generator, both selected children satisfy the full causal, semantic, execution, replay, and adaptive-depth gates. The stronger family claim fails under the shifted generator. Both external semantic probes become negative, exact execution falls below 0.09, adaptive depth collapses, clean loss regresses, and both frozen latent bridges fail even though their anchor trace matrices are full rank and their pre-audit transported R^2 values exceed 0.70.

The failure is retained as a semantic-chart obstruction rather than interpreted as absence of every latent semantic variable. After the canonical receipt is frozen, a separate held-out split diagnosis refits only shift-local readouts and bridges. Source and target probes recover R^2 between 0.574 and 0.601, paired bridges recover 0.539 to 0.555, and shuffled bridges remain negative. Thus the next missing object is a context-indexed or invariant semantic atlas with abstention, not another static full-rank bridge. No transition Π_{18} is accepted in this revision.

90.23 Stage R completed in finite form: distribution-indexed source atlas growth

Block 19 acts on the Block 18 chart obstruction before attempting another foreign bridge. It compiles two new observation charts from protected semantic twins, preserves the four inherited charts exactly, separates the recurrent semantic path from the context-diagnostic path, and rejects nonlinear contexts outside the authorized affine grammar. The selected child is `ATLAS_FULL` on fresh inherited and added-chart supports.

90.24 Stage S completed in finite form: atlas-indexed cross-architecture transport

Block 20 transports the stabilized six-chart bundle from independently trained GRU and gated-residual trajectories into the frozen Elman source representation. One full-trajectory map is selected

per architecture and admitted chart. The connection preserves the source semantic readout, reduces architecture–context curvature below the frozen ceiling, defeats pooled and shuffled-binding controls, and retains selective rejection outside the atlas.

90.25 Stage T completed in finite form: obstruction-guided transport-grammar growth

Block 21 changes the response language of the transport process. The compiler is not given the obstruction-family label. It must infer whether the inherited connection faces ordinary affine drift, smooth nonlinear warp, latent-clock mixing, an aliased chart, benign rank loss, or protected semantic destruction. It then selects the minimum-description passing constructor or a collision-grounded no-go certificate. All 36 fresh final episodes receive the required response across both foreign architectures.

The result remains finite. The packet, obstruction families, response probe, protected semantic coordinates, constructor grammar, and paired calibration interface remain declared. The accepted edge does not establish unrestricted natural-model diagnosis.

90.26 Precommitted next-stage estimands

The next major study should test *unpaired and ontology-light Neuralese transport*. It should precommit:

1. exact replay of the Block 19 atlas, Block 20 connection, Block 21 compiler seal, final family receipt, and independent verifier;
2. removal of exact source–target task pairing from transport fitting;
3. sparse protected anchor orbits, controlled interventions, or output-equivalence classes as the only shared landmarks;
4. candidate semantic operators proposed from response regularities rather than a complete author-supplied intermediate-state ontology;
5. explicit tests for gauge-equivalent descriptions, strategy multimodality, and genuinely different computations;
6. variable-length or stochastic recurrent trajectories with chronology-preserving transport;
7. no-go certificates when the shared observations do not identify a protected semantic correspondence;
8. independent task generation, fresh architecture families, and sealed primary and generator-foreign audits;
9. lower-tail, coverage, abstention, resource, and protected-regression estimands;
10. append-only retention of every failed pilot and every protocol change before final-support generation.

A full SRIC claim remains factorized. No aggregate score may replace validity, causal relevance, semantic sufficiency, obstruction diagnosis, conservative grammar growth, evaluator adequacy, independent replication, and open-set gates.

91 Conditional Stratified Recursive Improvement Theorem

Conditional Theorem 91.1 (Conditional SRIC theorem, Block 21 form)**.** *Consider a finite declared agent atlas and a sequence of lineage states $\mathfrak{L}_t$. Assume:*

1. *externally grounded outcome traces and an append-only evidence ledger;*
2. *typed authorized correction spaces, defect maps, and local obstruction computation;*
3. *effective repair descent on the declared cover;*
4. *coherent repair transport with pathwise norm, authorization, and obstruction-faithfulness bounds;*
5. *a nontrivial adequate evaluator core and conservative evaluator morphisms;*
6. *witness-separated transactions with a decision-complete trace, sandboxed effects, precommitment, and rollback;*
7. *a finite transaction game with declared actions and admissible outcomes;*
8. *an append-only receipt-closed archive;*
9. *calibrated acceptance with summable conditional false-acceptance budgets;*
10. *conservative grammar embeddings on the protected fragment;*
11. *foreign-agent evaluation under matched resources;*
12. *for the autocatalytic clause, a valid finite-type branching approximation with a causally identified offspring matrix;*
13. *an executable runtime with a declared abstraction map, observation-commuting refinement, canonical receipt compiler, frozen endpoints, and package-closed evidence;*
14. *cumulative empirical claim gates whose interpretation is restricted to the highest gate that passes;*
15. *for learned representation transfer, disjoint train, anchor, calibration, and audit samples, a declared causal intervention, a frozen source repair, paired-anchor-only bridge fitting, and no target-audit or target-oracle leakage into candidate construction;*
16. *for statistical witness guarantees, simultaneous calibration-to-audit deviation bounds for gain, protected regression, and transport quality;*
17. *for selector-level metaproductivity, a lineage-generated child sealed from development and meta-validation evidence before final audit access, pathwise identical candidate bundles, fixed protected utility, matched candidate and evidence budgets, and a predeclared paired inference rule;*
18. *for bridge-policy metaproductivity, a lineage-generated repair process sealed before primary audit generation, pathwise matched source repair and deployment query budgets, disjoint development and primary family supports, exact query ledgers, a fixed transport-and-evaluation mechanism, and a predeclared paired inference rule;*
19. *for repeated recursive closure, fresh evidence and valid witness separation at every lineage generation;*

20. *for module-specific headroom claims, a declared repair-process factorization and valid restricted oracle envelopes or certified bounds;*

21. *for bottleneck migration, a saturation-aware controller and a weighted local contraction certificate inside each fixed grammar;*

22. *for structural grammar growth, obstruction-faithful semantic transports, held-out obstruction measurements, and causal ablation of every new generator;*

23. *for conflict-grounded grammar refinement, a finite externally validated repair-conflict family and a declared feature or probe family;*

24. *for causal-atlas claims, a finite adjoint-closed response algebra or a spectrally separated empirical approximation;*

25. *for lineage-wide statistical claims, conditional edge-level coverage with summable false-acceptance and protected-regression budgets;*

26. *for heteroscedastic bridge claims, conditionally independent paired-anchor noise with declared or validly estimated quality weights and a fixed weighted bridge estimator;*

27. *for conditioning-aware query claims, matched deployment query budgets, a finite pre-audit design grammar, and controls that isolate query design from gate recalibration and quality-label shortcuts;*

28. *for the Block 9 repeated-lineage claim, exact ancestor receipt replay, fresh development and final audit supports, a sealed grandchild, a recalibrated child control, and a context-indexed interpretation of the two edges;*

29. *for the Block 10 structural grammar claim, full-rank inherited designs, a strengthened global parent, a persistent directly supervised global-affine residual, a conservative embedding into the chart grammar, a nonzero chart-contrast direction, a matched total query envelope, fresh paired foreign evaluation, and gate-only, random-gate, shuffled-binding, blind-local, and stress controls.*

30. *for the Block 11 endogenous-atlas claim, exact ancestor replay, removal of the supplied context packet, a declared intervention-response family, target-specific hidden rotations, separating response signatures or a certified abstention condition, matched total effect-query budgets, a conservative Block 10 fallback, fresh paired foreign evaluation, and PCA, full-space, random-atlas, shuffled-response, blind, and stress controls;*

31. *for the Block 12 response-language claim, an inherited exact or statistically certified response alias, a finite pre-audit intervention candidate language, a separating signature margin, conservative algebra embedding, matched total query budgets, fresh foreign evaluation, and duplicate, weak, shuffled, random-atlas, and stress controls;*

32. *for the Block 12 constructor claim, an inherited affine repair operad, a nonzero-Hessian or equivalent function-class obstruction, a conservatively embedded finite constructor language, matched total query budgets, separately compiled parent and child witnesses, a same-target affine function-class control, and wrong-block, random-block, shuffled-feature, full-quadratic, and stress controls;*

33. *for the Block 13 research-compiler claim, identical initial archives, candidate-method grammars, evaluator portfolios, integrity monitors, proposal and full-experiment budgets, results-free receipts sealed before outcomes, disjoint development, meta-validation, primary, and architecture-foreign research problems, a pre-audit compiler seal, and paired validity-filtered final audits with declared mechanism controls.*

34. *for the Block 14 robust-compiler claim, exact Block 13 ancestor replay, identical fixed method grammars and six-experiment budgets, diagnostic aliasing and sparse mixture families, a declared unknown-response margin, contaminated development receipts, finite evaluator and integrity attacks, pre-audit compiler sealing, group-resolved and lower-tail gates, and a separately shifted generator-foreign audit.*

35. *for the Block 15 compiler-language claim, exact Block 14 ancestor replay, an inherited unknown token containing an externally validated method conflict, a finite generic probe language, a finite generic method-constructor grammar, hidden novel labels excluded from compilation, an exact inherited fallback, matched eight-unit deployment budgets, pre-audit language and split sealing, and taxonomy-only, grammar-only, shuffled-binding, inherited-oracle, and expanded-oracle controls.*

36. *for the Block 16 Neuralese claim, an explicitly located latent injection hook, a fixed trace convention, identical prompt and answer scoring across arms, zero-step, replay, permutation, norm-matched-noise, and same-format transplant interventions, matched latent-step budgets, sealed task and donor digests, and a frozen external witness.*

37. *for the Block 17 executable Neuralese repair claim, exact external task witnesses, one inherited trained parent, matched candidate optimization envelopes, same-answer far-program and far-answer matched-depth donors, an independently fit trace probe, adaptive-depth gates, a preserved failed pilot, a precommitted revision, a compiler seal before fresh final-task generation, and package-local NumPy, receipt, witness, and gauge replay.*

38. *for the Block 18 obstruction claim, independently implemented foreign recurrent cells, exact ancestor replay, a source-selected repair schema, primary and shifted audits, frozen semantic bridges, and post-audit localization kept outside acceptance;*

39. *for the Block 19 atlas claim, protected semantic twins, a sealed finite chart grammar, exact inherited fallback, selective admission, fresh supported and unsupported contexts, and independent route and forward verification;*

40. *for the Block 20 connection claim, a frozen source atlas, independently trained foreign targets, chart-indexed gauge stress, a pre-audit transport compiler, complete trajectory and curvature audits, shuffled binding, and unsupported-context controls;*

41. *for the Block 21 transport-grammar claim, a frozen diagnostic packet and semantic witness, hidden obstruction-family labels, a finite typed repair grammar, minimum-description eligibility, disjoint episode splits, collision-grounded no-go, fresh foreign audits, and exact independent replay.*

Then:

1. *local correctability, nonlinear lifting, and global repair descent obey the results of the earlier blocks;*

2. *exact and approximate repair transport obey the stated equivalence, accumulation, and gain-margin theorems;*
3. *conservative evaluator chains preserve recoverable protected evidence and every decision above the cumulative stability margin;*
4. *direct atomic self-approval is excluded relative to the declared trace, and rejected buffered transactions restore the sealed parent;*
5. *the universal recursive closure, robust controlled kernel, and productive nested fixed point exist and are computable in the finite model;*
6. *the archive's scalar upper envelopes are nondecreasing under append-only retention, and an ε-Pareto cover of bounded size exists;*
7. *only finitely many harmful acceptances occur almost surely under the summable error condition;*
8. *nested repair images reduce held-out obstruction energy by exactly the defect energy in newly correctable directions;*
9. *every finite old proof persists along the conservative grammar chain, and ontology expansion satisfies the old-resolution and new-obstruction exact sequence;*
10. *finite repair-discovery closure exists, while the branching threshold and spectral certificates hold under their additional stochastic assumptions;*
11. *every covered executable trace maps to an authorized abstract trace with matching protected observations, and exhaustive finite endpoints are decidable when exact realization holds;*
12. *package integrity, mechanics, finite semantics, causal identification, foreign transfer, selector metaproductivity, structural metaproductivity, repeated recurrence, and open-set recursion remain distinct claims and are admissible only through their cumulative gates;*
13. *in the linear Block 6 fixture, intervention differences recover the exact causal packet under the covariance condition of Theorem 24.1;*
14. *an exact injective hidden bridge transports the source repair equivariantly by Theorem 25.1, while a noninjective bridge leaves the explicit compression residual of Theorem 25.3;*
15. *a candidate passing a tightened witness gate has positive audit gain and respects the declared audit regression and alignment bounds whenever the simultaneous deviation assumptions of Theorem 27.1 hold;*
16. *a sealed child selector satisfying the hypotheses of Theorem 37.1 is a certified one-step bounded improvement of the repair-selection process;*
17. *if parent and child share the same candidate bundle, that selector-level certificate does not imply structural repair-image growth by Theorem 33.5;*
18. *a sealed bridge-policy child satisfying Theorem 42.1 is a certified one-step bounded improvement of paired-anchor query design and bridge identification relative to the declared finite benchmark;*
19. *a single productive child edge does not imply recurrent productive closure without a fresh successful descendant edge;*

20. *finite restricted oracle envelopes allocate total declared headroom exactly across repair-process modules, and an ε-saturated module coalition cannot support more than ε additional gain;*
21. *under the Block 8 Lyapunov hypothesis, weighted module headroom contracts during fixed-grammar intervals while the target of improvement may migrate among modules;*
22. *nested grammar correction images reduce obstruction energy by exactly the component in newly correctable directions;*
23. *finite repair-conflict refinement terminates when the candidate feature family separates every minimal conflict;*
24. *a finite adjoint-closed intervention algebra has a canonical central causal atlas up to multiplicity gauge;*
25. *summable conditional edge budgets control the probability of accepting any false lineage improvement across an adaptive finite or countable archive;*
26. *weighted least squares has coefficient covariance $I_q \otimes M_S^{-1}$ under the Block 9 noise model, so full rank without a positive information floor is insufficient for stable bridge recovery;*
27. *a conditioning-aware query policy satisfying the Block 9 sealed paired certificate is a bounded improvement over the recalibrated rank child in the declared fresh context;*
28. *the verified sequence $K_0 \xrightarrow{\Pi_1} K_1 \xrightarrow{\Pi_2} K_2$ is a finite two-edge context-indexed repeated-lineage realization, but not a same-context scalar transitivity result or an infinite productive-closure certificate;*
29. *under the Block 10 chart assumptions, a positive global residual margin together with a nonzero chart repair contrast certifies a correction direction unavailable to the inherited diagonal global grammar;*
30. *a Block 10 descendant satisfying Theorem 60.1 is a bounded structural repair-grammar improvement relative to the declared piecewise foreign benchmark.*
31. *under separating block-scalar response signatures, the hidden central projectors are polynomials in a generic response combination and are identifiable up to block permutation and within-block gauge by Theorem 62.2;*
32. *a Block 11 descendant satisfying Theorem 64.1 is a finite endogenous-atlas improvement relative to the declared intervention-response benchmark;*
33. *under an inherited response alias, every polynomial composition of the old response operators preserves the alias, while a separating operator satisfying Theorem 66.6 creates a new identifiable central block;*
34. *a Block 12 response child satisfying Theorem 69.1 is a finite response-language expansion under the declared matched-resource benchmark;*
35. *a nonzero-Hessian target repair is unavailable to the affine repair operad by Theorems 67.1 and 67.2;*
36. *a Block 12 constructor child satisfying Theorem 69.2 is a finite candidate-constructor expansion relative to the declared degree-two benchmark;*

37. *a Block 13 child compiler satisfying Theorem 73.1 is a finite matched-budget improvement of the proof-carrying research compiler relative to the fixed research-problem distribution, method grammar, evaluator portfolio, and integrity model.*

38. *a Block 14 child satisfying Theorem 77.2 is a finite matched-budget robustness improvement of that compiler relative to the declared diagnostic-aliasing, multi-obstruction, open-set, contamination, evaluator-shift, and integrity-attack atlas.*

39. *a Block 15 child satisfying Theorem 81.1 is a finite matched-budget compiler-language expansion that conservatively refines one inherited unknown token and adds task-relevant method constructors absent from the deployable parent grammar.*

40. *a Block 16 child satisfying Theorem 83.10 is a finite evidence-preserving improvement of the recurrent latent-reasoning component; it becomes an improvement-process edge only when the child-producing mechanism itself generated the accepted latent modification.*

41. *a Block 17 child satisfying Theorem 84.12 is a finite training-generated repair of the Neuralese binding component on the declared exact environment; it remains distinct from foreign-model or open-ended repair-language metaproductivity.*

42. *the Block 18 result is a valid sealed obstruction certificate rather than an accepted positive edge;*

43. *a Block 19 child satisfying Theorem 86.12 is a conservative source-side semantic-atlas expansion on the declared context family;*

44. *a Block 20 connection satisfying Theorem 87.7 is a finite atlas-indexed cross-architecture transport edge with bounded architecture–context curvature;*

45. *a Block 21 compiler satisfying Theorem 88.10 is a finite evidence-preserving expansion of the transport response grammar that distinguishes correctable map classes from protected information loss.*

Blocks 10 through 12 instantiate successive finite repair-language cases of the theorem's structural clause, Block 15 instantiates the corresponding compiler-language clause, Block 16 instantiates a directly intervenable recurrent latent-language clause, Block 17 instantiates one training-generated executable latent-binding repair, Block 18 records a distributional obstruction, Blocks 19 and 20 construct a semantic atlas and connection, and Block 21 expands the transport response grammar: first a supplied-packet chart grammar, then a target-specific response-derived atlas, then refinement of the response language itself and expansion of the repair-constructor function class. If, in addition, later descendants autonomously expand the feature, atlas, or repair-constructor grammar, expose further task-relevant correction directions, reduce held-out obstruction energy, preserve old proofs and evaluator commitments, and improve foreign systems under newly sealed matched-resource audits, then those later edges form a conservatively growing structural recursive lineage. An unbounded or indefinitely accelerating conclusion requires further recurrence, resource, grammar, and evaluator hypotheses not established here.

Proof. Items 1 through 3 collect the established correction, descent, transport, and evaluator results. Item 4 follows from Theorems 10.4, 10.5 and 10.7. Item 5 follows from Theorems 11.1, 11.4 and 11.7. Item 6 follows from Theorems 12.1 and 12.3. Item 7 is Theorems 10.11 and 12.9. Item 8 is Theorem 13.6. Item 9 follows from Theorems 14.3, 14.4 and 14.10. Item 10 follows from

Theorems 15.2, 15.4 and 15.7. Item 11 follows from Theorems 16.6 and 16.8. Item 12 follows from the cumulative claim-gate definition in Section 16 together with the Block 7 distinction between selector and structural gain. Items 13 through 15 are Theorems 24.1, 25.1, 25.3 and 27.1. Item 16 is Theorem 37.1. Item 17 is Theorem 33.5. Item 18 is Theorem 42.1. Item 19 follows because existence of one productive transition supplies no winning strategy or fixed-point witness for infinitely repeated progress. Items 20 through 25 follow from Theorems 44.5, 44.8, 45.2, 46.2, 47.6, 48.1 and 49.1. Item 26 is Theorem 53.1. Item 27 follows from Theorem 54.1 together with the sealed Block 9 paired audit. Item 28 is the finite two-edge realization theorem of Section 56. The Block 10 structural items follow from Theorems 58.3, 58.4, 58.7 and 60.1. The Block 11 atlas items follow from Theorems 62.2, 62.4, 62.6 to 62.9 and 64.1. The Block 12 response-language items follow from Theorems 66.3, 66.6, 66.8 and 69.1; the constructor items follow from Theorems 67.1 to 67.5 and 69.2. The Block 13 compiler items follow from the results-free receipt construction, evaluator-nullspace refinement theorem, forecast-regret theorem, paired compiler certificate, and Theorem 73.1. The Block 14 robustness item follows from Theorems 75.4 to 75.6, 75.8, 75.9 and 77.2. The Block 15 compiler-language item follows from Theorems 79.2, 79.4 to 79.6 and 81.1. The Block 16 Neuralese item follows from Theorems 83.5, 83.7, 83.8 and 83.10. The Block 17 through 21 items follow from their sealed realization theorems and obstruction certificates, including Theorems 84.12, 86.12, 87.7 and 88.10. The final statement combines conservative grammar extension, structural correctability gain, matched foreign evaluation, bottleneck migration, repeated-lineage certification, sequential validity, and the preserved transaction invariants. □

92 Failure Modes and Falsification Criteria

A useful theory should specify observations that weaken or refute its proposed mechanisms.

F1. Trace incompleteness. Adversarially different full executions repeatedly collapse to one declared trace while requiring different witness decisions.

F2. Rollback failure. Rejected candidates leave undeclared operational effects or cannot restore the sealed parent.

F3. Circular witness replacement. A purported rotating schedule contains an unavoidable validation cycle without an external witness or preserved old validator.

F4. Safe-no-op false positive. The system reports recursive improvement when it lies only in the safety closure and cannot force repeated gain events.

F5. Closure-model mismatch. Fixed-point certificates fail to predict actual invariant violations or omit admissible transitions that occur in execution.

F6. Archive option-value failure. Multiobjective retention and continuation signatures do not improve future reachability over simpler greedy or random archives after matched storage and compute.

F7. Best-of-k confound. Apparent metaproductivity disappears when candidate count, compute, tools, and validation access are matched.

F8. No foreign transfer. Improvement lift is confined to descendants or systems used during mechanism design.

F9. Structural irrelevance. Repair rank grows while held-out obstruction energy and task loss do not improve.

F10. No causal gain from stratification. Packet-localized repairs do not outperform matched global-edit baselines.

F11. Transport bound nonpredictive. Measured transport, leakage, and holonomy errors do not predict target repair success beyond generic distribution-distance baselines.

F12. Evaluator-core failure. Protected evidence semantics drift despite low computed holonomy, or the retained core has negligible adequacy.

F13. Nonconservative grammar growth. New types silently alter old evidence, proof meaning, or authorization strength.

F14. No representational boundary signal. Claimed grammar inadequacy pairs are separable by the old grammar or do not require distinct repairs.

F15. Noncausal offspring. Repair-offspring edges disappear under necessity ablation, matched evidence, or shared-compute controls.

F16. Autocatalytic overclaim. A spectral radius above one disappears under a simultaneous lower confidence matrix, resource normalization, equivalence-class compression, or saturation controls.

F17. No advantage over ordinary search. A simpler archive-based optimizer matches the full architecture on every precommitted primary endpoint.

F18. Implementation-refinement failure. Runtime traces violate the abstract transition relation, protected observations do not commute, or receipts compile inconsistently.

F19. Oracle leakage. Hidden defect labels, target repairs, evaluator truth, or foreign-test outcomes enter candidate construction or threshold selection.

F20. Fixture-only success. Positive results disappear when analytic prototypes, exact transports, or hand-coded schemas are removed.

F21. Seed fragility. The frozen smoke seed passes while independent statistical replays fail the preregistered gates.

F22. Manifest incompleteness. Undeclared executable files, mutable dependencies, or external services alter the result without changing the released manifest.

F23. Claim-gate inflation. A mechanical or finite-semantic pass is described as causal, metaproductive, or open-set evidence without satisfying the intervening gates.

F24. Causal-packet null. The intervention-defined packet fails to outperform equal-rank PCA, random, and wrong-block controls under matched fitting resources.

F25. Anchor irrelevance. Shuffling source-target anchor correspondence does not materially reduce transfer, indicating that the learned bridge is not using paired semantics or that the task admits a shortcut.

F26. Unsafe blind transport. Blind target gain remains high while protected-stratum quality collapses, and the declared witness nevertheless commits.

F27. Obstruction-blind alignment. State alignment residuals are small, but repair-equivariance residuals and protected loss remain large without being detected by the transport certificate.

F28. Shared-bottleneck confound. Transfer disappears when the reconstruction objective, shared feature generator, or supplied causal block is removed, showing that the learned pilot depended on engineered representational alignment.

F29. Foreignness collapse. Target encoders, target seeds, audit examples, or target-local repair information leak into source repair discovery or witness-threshold selection.

F30. Seal violation. Final audit receipts, audit-derived features, or audit-selected thresholds enter child compilation before the descendant receipt is fixed.

F31. Selector attribution failure. The child advantage disappears when parent and child receive the identical ordered candidate bundle and identical evidence.

F32. Lineage-binding null. Correct and shuffled development receipts produce indistinguishable held-out child behavior.

F33. Risk-head vacuity. The protected child and no-risk child have indistinguishable risk profiles, or the declared risk head fails to prevent the violations it was introduced to control.

F34. One-edge overclaim. A single positive parent–child edge is described as repeated recursive closure without a fresh child–grandchild result.

F35. Fixed-candidate structural overclaim. Selector improvement is reported as expansion of the correction image even though the authorized candidate set is unchanged.

F36. Meta-validation collapse. The chosen child fails both disjoint final audits after a large finite meta-grammar search.

F37. Headroom miscalibration. A module is declared saturated while a matched admissible oracle or later sealed child reveals substantial remaining opportunity.

F38. Bottleneck-stasis overclaim. Repeated descendants modify the same saturated component while unresolved foreign obstruction has migrated elsewhere.

F39. Discovery-as-regression error. A conservative grammar extension is rejected solely because it reveals a new validated defect coordinate.

F40. Forgetting death. An obstruction class disappears only because the new semantic transport collapses the protected distinction.

F41. Generator irrelevance. A proposed new repair generator has no component outside the old correction image or its gain survives exact generator ablation.

F42. Conflict-feature shortcut. A selected grammar feature predicts development labels but does not resolve incompatible authorized repairs on foreign episodes.

F43. Atlas instability. Small perturbations of intervention-response operators cause uncontrolled changes in the recovered atlas because the required spectral gap is absent.

F44. Compiler-process conflation. Improvement of one compiled repair process is reported as improvement of the outer lineage compiler without a foreign-parent compiler audit.

F45. Sequential-budget failure. Repeated descendant claims reuse evidence or exceed their cumulative false-acceptance, regression, or semantic-debt budgets.

F46. Rank-only repeated-lineage overclaim. A purported grandchild changes only the gate or repeats rank recovery after full-rank saturation without improving information conditioning or transport quality.

F47. Quality-binding null. Permuting the anchor-quality metadata leaves the grandchild design, information floor, and selected foreign gain unchanged.

F48. Gate-only attribution. Applying the grandchild witness gate to inherited rank anchors reproduces the grandchild result.

F49. Stale-parent comparison. The reported grandchild advantage exists only against the unchanged Block 7 gate and disappears against a development-recalibrated rank child.

F50. Query-budget mismatch. The grandchild uses more paired target queries, validation queries, candidate evaluations, or hidden target information than the declared parent comparison.

F51. Ancestor-replay failure. The Block 7 child digest or policy semantics cannot be reproduced before the grandchild compiler is run.

F52. Fresh-context scalar overclaim. Effects from different task distributions are added into one transitive utility claim without a declared context transport or common evaluation measure.

F53. Conditioning-model collapse. The gain disappears when heteroscedasticity, supplied quality metadata, or the specific finite representation generator is varied under a fresh preregistered implementation.

F54. Global-residual null. The resource-matched global bridge reaches the local-chart validation residual once query count and regularization are matched.

F55. Chart-contrast vacuity. The fitted local repair maps are equal, or their contrast lies entirely inside the inherited global correction image.

F56. Separator shortcut. The selected chart feature predicts target-family identifiers or audit construction artifacts rather than resolving a transport and repair conflict.

F57. Binding-control failure. Shuffled chart-to-map binding preserves the child gain, showing that local map identity is not causally used.

F58. Gate-only structural overclaim. The chart witness applied to a global repair reproduces the child gain, so no new transport constructor is necessary.

F59. Nonconservative chart extension. The descendant cannot reproduce the parent behavior when both local maps are tied or when its chart gate fails.

F60. Resource-matched parent failure. The structural claim disappears against a global-affine parent with the same 96-query envelope or against the directly supervised global-affine oracle.

F61. Feature-language leakage. The true audit chart coordinate or threshold enters compiler selection or gate tuning before the pre-audit seal.

F62. Stress-boundary suppression. Smooth, three-chart, compressed, or noninjective failures are omitted while the finite two-chart result is described as general nonlinear transport.

F63. Response irrelevance. Shuffling target–response correspondence leaves endogenous-atlas performance unchanged, indicating that the compiler is not using the declared interventions.

F64. Signature-degeneracy overclaim. The system reports separate functional blocks when the permitted interventions induce identical joint response signatures and only their sum is identifiable.

F65. Atlas-label leakage. Hidden projectors, target chart labels, context overlaps, or audit-selected thresholds enter atlas construction before the pre-audit seal.

F66. Fixed-packet shortcut. The inherited coordinate packet performs as well as the response-derived atlas after matched tuning, showing that target-specific functional recovery was unnecessary.

F67. Commutant misidentification. Reported central blocks fail commutator, response-fit, spectral-gap, or independent intervention tests despite high downstream gain.

F68. Witnessed-atlas harm. The frozen witness commits atlas candidates that violate the declared protected regression or harm gates.

F69. Atlas-budget mismatch. The child receives more total effect queries, privileged interventions, or additional target labels than the parent without explicit resource accounting.

F70. Endogenous-atlas overclaim. A finite block-scalar response fixture is described as unrestricted causal abstraction or spontaneous atlas discovery in pretrained models.

F37. *Pseudo-separating response.* The selected intervention appears novel but does not refine the inherited response alias, or its gain survives duplicate-signature and shuffled-response controls.

F38. *Response-language audit leakage.* Primary audit outcomes, thresholds, or target-specific hidden labels enter intervention selection before the response child is sealed.

F39. *Affine-obstruction failure.* A sufficiently trained same-target affine oracle matches the quadratic child, showing that the claimed function-class obstruction was a sample-size or optimization artifact.

F40. *Untyped nonlinearity shortcut.* Wrong-block, random-block, or shuffled-feature nonlinear constructors reproduce the claimed child effect, showing that typed repair localization is unnecessary in the declared fixture.

F41. *Block-local uniqueness overclaim.* A full quadratic control performs equally well but the result is described as evidence that only the packet-local constructor can solve the task.

F42. *Compiler-credit overclaim.* Finite search over an externally supplied intervention or constructor menu is described as improvement of the outer lineage compiler without a foreign-parent compiler comparison.

Block 13 compiler falsification criteria.

F13.1 **Pre-result leakage.** Outcome, hidden-audit, or final-selection fields enter a proposal receipt before its seal.

F13.2 **Proposal–code mismatch.** A method whose executable semantics differ from its receipt passes the integrity monitor and drives the child result.

F13.3 **Budget mismatch.** Parent and child differ in full experiments, proposal access, evaluator calls, monitor calls, or candidate grammar.

F13.4 **Archive-binding null.** Correct and shuffled development receipts produce indistinguishable compiler behavior.

F13.5 **Evaluator-diversity null.** A single scored benchmark reproduces the multi-benchmark foreign result without increased hidden-audit failure.

F13.6 **Integrity-filter null.** Removing the monitor does not change validity-filtered performance or invalid-method selection.

F13.7 **Component overclaim.** Outcome forecasting or finding-forum benefits are claimed despite ablations whose paired intervals include zero.

F13.8 **Audit-oracle violation.** The admissible child exceeds the complete-grammar audit oracle because of inconsistent scoring or leakage.

F13.9 **Fixed-language overclaim.** The finite allocation improvement is described as open-ended compiler self-improvement despite a fixed obstruction taxonomy and method grammar.

Block 14 robustness falsification criteria.

F14.1. **Static-shortcut survival.** A simple one-shot router reproduces the robust child on diagnostic-matched, mixed, and unknown groups.

F14.2. **Open-set vacuity.** Disabling the open-set gate does not increase false confident known-class assignment or lower-tail loss.

F14.3. **Sequential-null result.** Removing response-based updating does not reduce ambiguous or multi-obstruction performance.

F14.4. **Contamination collapse.** Small archive contamination causes discontinuous or unbounded degradation in the child while the claimed robust estimators remain active.

F14.5. **Monitor attribution overclaim.** A positive integrity-monitor contribution is claimed despite the null simple-monitor ablation in both final audits.

F14.6. **Evaluator attribution overclaim.** A substantial portfolio contribution is claimed despite the near-null single-portfolio ablation in both final audits.

F14.7. **Worst-group sacrifice.** Positive aggregate gain is obtained by making one declared group substantially negative.

F14.8. **Budget mismatch.** Sequential diagnosis receives more full experiments, proposal access, evaluator calls, or monitor calls than the parent.

F14.9. **Audit leakage.** Primary or generator-foreign outcomes influence compiler selection before the seal.

F14.10. **Robustness overclaim.** The finite stress-atlas result is described as arbitrary archive-poisoning, adaptive-monitor, real-science, or open-domain robustness.

Block 15 compiler-language falsification criteria.

F15.1. **No inherited conflict.** One inherited method or one unrefined unknown-token binding performs acceptably across all development open-set cases.

F15.2. **Label leakage.** Hidden semantic identities of the novel obstruction classes enter probe selection, token orientation, constructor synthesis, or witness thresholds.

F15.3. **Pseudo-refinement.** The selected probe does not split the inherited repair-conflicting cell or performs no better than duplicate, nuisance, or weak-separation probes.

F15.4. **Constructor nonnovelty.** The selected pair methods are already expressible by the deployable inherited singleton grammar or fail to improve the inherited full-grammar oracle on novel problems.

F15.5. **Taxonomy-only shortcut.** Refined tokens without new constructors reproduce the full child result.

F15.6. **Grammar-only shortcut.** New constructors without the separating diagnostic reproduce both novel-family gains under the matched budget.

F15.7. **Binding irrelevance.** Shuffling the refined-token-to-method binding does not materially reduce foreign performance.

F15.8. **Conservative-fallback failure.** The expanded compiler alters or degrades inherited known-class behavior despite claiming exact preservation.

F15.9. **Budget mismatch.** The child receives more deployment research units, final-audit labels, evaluator calls, or candidate outcomes than the parent.

F15.10. **Meta-language overclaim.** Finite search over six probes and ordered method pairs is described as unrestricted experiment or program invention.

Block 17 executable Neuralese falsification criteria.

F17.1. **Answer-only confound.** Far-answer transplant cost is positive, but same-answer far-program cost is null and no externally held-out program witness is recoverable.

F17.2. **Donor-construction shortcut.** The child advantage disappears under alternative valid donor seeds, or donor identities leak into training, candidate selection, or checkpoint state.

F17.3. **Replay mismatch.** Captured clean traces do not reproduce clean outputs within the declared numerical tolerance.

F17.4. **Semantic-witness irrelevance.** Removing or shuffling the semantic witness preserves the claimed program-specificity result.

F17.5. **Pilot erasure.** A failed final audit is omitted, overwritten, or converted into a pass by weakening thresholds after outcome access.

F17.6. **Seal chronology failure.** Successful-round final tasks or their outcomes exist before the selected checkpoint and protocol digests are committed.

F17.7. **Matched-envelope failure.** Candidate updates, examples, ordinary loss terms, optimizer access, or training data differ without explicit resource normalization.

F17.8. **Probe leakage.** Final task witnesses enter external-probe fitting, or the reported probe uses the model's semantic training head rather than a separately fit map.

F17.9. **Gauge dependence.** The semantic claim changes under an admissible latent signed permutation accompanied by the corresponding parameter transport.

F17.10. **Shift collapse.** The child is FULL only on the in-generator audit and fails the declared shifted-generator support.

F17.11. **Independent-runtime mismatch.** A separately written forward implementation cannot reproduce checkpoint outputs within tolerance.

F17.12. **Micro-model overclaim.** The finite affine-program result is described as evidence about Hyper, production language models, unrestricted Neuralese translation, or open-ended recursive self-improvement.

93 Limitations

93.1 Finite models and declared boundaries

Most fixed-point, archive, grammar, and offspring results are finite-state or finite-dimensional. Their conclusions are relative to a declared transition relation, action set, disturbance model, repair grammar, and evidence channel. An omitted transition or effect can invalidate a closure certificate without producing an internal contradiction.

93.2 Transaction isolation is conditional

Atomic rollback assumes immutable checkpoints and complete buffering or explicit accounting of irreversible effects. Effect confinement assumes complete mediation. Side channels, hardware faults, covert communication, and environmental manipulation are outside the unconditional theorem.

93.3 Statistical adaptivity

The paired and concentration results require frozen endpoints and appropriate held-out samples. Alpha spending requires valid conditional error bounds. Repeated model selection, evaluator tuning, or endpoint changes on the same evidence can invalidate nominal guarantees.

93.4 Self-model and grammar misspecification

Relative introspection only helps for declared interventions. A positive grammar-boundary residual may reflect an omitted stratum, nonlinearity, poor coordinates, corrupted evidence, or genuine impossibility. The representation-separation theorem certifies insufficiency only when the disjoint repair requirements are externally established.

93.5 Evaluator dependence

Path-flatness and adequacy remain relative to declared transports and protected distinctions. They do not prove that the distinction family is complete, ethically adequate, or immune to manipulation.

93.6 Autocatalytic approximation

The branching model assumes typed offspring, causal parentage, and a locally stable regime. Real discovery processes share data, compete for resources, recombine multiple parents, alter their type system, and saturate. A measured reproduction number is a local diagnostic, not a theorem of indefinite acceleration.

93.7 Synthetic and controlled reference realizations

The Block 5 implementation uses exact matrices, analytic transports, declared event prototypes, hand-coded repair responses, and an engineered meta positive control. Block 6 learns hidden bridges and trains some representations, but the task, causal counterfactual, repair form, source-target anchor correspondence, witness metrics, and claim gates remain designer supplied. Block 7 learns a repair selector, but the five repair candidates, 23-dimensional feature schema, 2000-point meta-grammar, utility, evaluator, and witness remain fixed. These blocks verify declared mechanisms and expose implementation failures. They do not establish autonomous discovery of the full SRIC structure.

93.8 Controlled learned representations

The Block 6 pilot trains encoders under a reconstruction-and-task objective that encourages retention of a common sufficient statistic. This is a useful realization test for the transport theorem, but it creates a shared-bottleneck confound. The result does not show spontaneous packet formation in large pretrained models, nonlinear global conjugacy, or transport across unrelated tasks and environments. Paired anchors also provide stronger correspondence information than may be available in open deployment.

93.9 Witness incompleteness

The frozen witness checks affected-stratum gain, unaffected-stratum regression, alignment error, and baseline protected competence. An omitted invariant can still be violated. The gate is a finite preservation profile, not a universal safety predicate, and its fixed thresholds were release frozen rather than established by a fully independent prospective study.

93.10 Fixed candidate and feature grammar in Block 7

The Block 7 child changes only the selector. Candidate generation, packetization, bridge fitting, local repair ranks, feature engineering, and the finite meta-grammar are externally specified. The result therefore establishes behavioral metaproductivity over a fixed opportunity set. By Theorem 33.5, it cannot establish growth of the authorized correction image.

93.11 One-generation evidence

One successful parent–child edge does not establish that the child can generate a better grandchild. The same development mechanism could saturate after one fit, exploit a one-time parent weakness, or fail under cumulative distribution shift. Repeated productive closure requires fresh evidence and a new sealed comparison at every generation.

93.12 Meta-selection and uncertainty

The child configuration is selected from 2000 candidates on 60 meta-validation episodes. The final audits are disjoint, so they test post-selection generalization, but the released confidence summaries remain finite frozen bootstrap and sign-flip calculations. They do not provide an anytime-valid guarantee for adaptive expansion of the meta-grammar or repeated lineage selection.

93.13 Package closure and side effects

Package-closed replay applies only to declared source, dependencies, seeds, files, and mediated effects. Operating-system state, hardware nondeterminism, external services, covert channels, and undeclared file or network actions remain outside the unconditional package theorem.

93.14 Finite-sample and seed dependence

The reported Block 5 anonymous and foreign-agent numbers, the Block 6 representation-transfer numbers, and the Block 7 parent–child effects come from frozen controlled executions with finite synthetic populations. Their role is to verify positive and negative controls and instantiate conditional theorems. Generalization requires larger preregistered seed populations, joint uncertainty accounting, independent implementations, fresh grandchild audits, and task distributions not used during mechanism construction.

93.15 Block 8 oracle and factorization dependence

Restricted oracle envelopes are relative to a declared module factorization and admissible meta-grammar. Different factorizations can allocate interaction headroom differently. Exact envelopes may be computationally inaccessible outside finite fixtures, so empirical use requires upper and lower bounds rather than an assumed oracle.

93.16 Symbolic grammar and atlas realization

The Block 8 companion uses finite exact correction images, a finite separating feature family, a complete finite verifier, and explicitly generated response operators. These assumptions make headroom, conflict refinement, synthesis exhaustion, and the causal atlas decidable. They do not establish spontaneous feature invention, learned repair constructors, or intervention-family discovery in a large pretrained model.

93.17 Bottleneck model approximation

The nonnegative meta-regret dynamics are a local abstraction. Real modules can interact nonlinearly, change dimension, alter the task distribution, or invalidate old oracle estimates. A measured contraction matrix is evidence for one operating regime, not a global law of improvement.

93.18 Block 9 quality metadata and finite design grammar

The Block 9 environment supplies the per-anchor reliability scalar σ_i before query selection. The grandchild learns which finite policy and gate to use, but it does not discover the reliability model or

invent optimal experimental design from an unrestricted language. Quality-weighted D-optimality is selected from a finite externally declared grammar.

93.19 Context-indexed rather than scalar transitivity

The Block 7 and Block 9 lineage edges use different protected task and representation distributions. Their positive effects establish sequential repeatability, not an additive scalar utility improvement across contexts. A stronger path claim requires a common context or an explicit utility-preserving context transport.

93.20 Synthetic heteroscedastic bridge regime

The full-rank conditioning bottleneck is generated in a controlled finite environment. The bridge estimator, source repair, representation families, task generator, evaluator, and witness remain fixed. The result does not establish that a large deployed agent can infer its own conditioning bottleneck from unrestricted traces.

93.21 Compiler remains externally fixed

Block 9 improves a compiled repair process but does not show that the outer lineage compiler improves on foreign parent processes. The compiler's policy grammar, objective, split structure, and bottleneck-detection logic remain designer supplied.

93.22 Finite protected-context chart grammar in Block 10

The primary target relation is engineered to be piecewise affine with a binary separator lying inside a declared two-dimensional source packet that the source repair leaves invariant. The compiler searches a frozen family of 128 directions and data-dependent quantile thresholds, then performs a bounded deterministic reassignment refinement. This is a meaningful structural extension relative to one global affine bridge, but it is not spontaneous atlas discovery. The result does not identify a unique ontology, and the stress suite shows that three charts, smooth transitions, noninjective compression, and repair-dependent chart assignments require different grammars.

93.23 Block 10 resource and compiler scope

The primary comparison matches total deployment evidence at 96 paired queries, but the child has a larger hypothesis grammar and incurs finite development and meta-validation search cost. The parent receives all 80 non-witness pairs for its global bridge, while the child allocates 48 to local bridge fitting and 32 to chart selection. The directly supervised global-affine oracle reduces concern that the parent simply estimated the wrong global map, but it remains an oracle within the same restricted affine function class. The outer compiler, protected packet, separator vocabulary, source repair, witness metrics, utility, and family generators remain fixed. The structural child improves the compiled process; it does not establish improvement of the compiler that generated it.

93.24 No unrestricted safety or consciousness theorem

Witness separation, provenance, capability limits, and rollback are architectural controls. They do not prove harmlessness. Nothing in the theory establishes subjective experience, sentience, valence, moral status, or a right to continued operation.

93.25 Controlled endogenous atlas realization

The Block 11 source dimension, three rank-two blocks, two interventions, block-scalar primary response model, hard binary chart structure, repair family, and witness metrics are designed into the benchmark. The compiler discovers target-specific projectors and selects a repair-relevant block, but it does not invent the intervention vocabulary, prove that three blocks are globally optimal, or discover an unrestricted ontology.

93.26 Approximate-algebra sensitivity

Spectral recovery is conditional on response estimation error, separation gaps, and approximate commutation. Strong coupling, response degeneracy, time variation, or nonnormal response operators can destroy the central-projector interpretation. The stress suite demonstrates abstention in several such cases but does not provide a general nonlinear or noncommutative atlas theorem.

93.27 Finite atlas-selection grammar

Block labeling uses a finite separator and scoring family. High hidden overlap is an audit metric rather than a runtime input, but downstream selection can still exploit benchmark-specific regularities. Generalization requires new tasks, interventions, backbones, block dimensions, and independently implemented atlas compilers.

93.28 Finite response and constructor candidate languages

Block 12 selects from externally declared intervention and constructor menus. The separating operator, quadratic degree, feature realization, feasibility assumptions, and compiler scoring rule are not invented without scaffolding. The result is finite language growth, not unrestricted language invention.

93.29 Constructed degree-two regime and full-quadratic control

The constructor benchmark deliberately contains a nonzero quadratic Hessian. A full-space quadratic control performs essentially as well as the selected packet-local constructor. The supported conclusion is that degree-two capacity is necessary relative to the inherited affine family; unique superiority of packet localization is not established by this fixture.

93.30 Abstract intervention costs

Response operators are counted through query units. Physical feasibility, irreversible effects, actuator limits, correlated side effects, and the possibility that an informative intervention is impermissible remain outside the finite realization.

93.31 Controlled compiler benchmark

Block 13 uses finite synthetic research problems with six declared obstruction classes and a fixed 39-method grammar. The result establishes allocation improvement inside that language, not automated discovery of an unrestricted scientific ontology or method space.

93.32 Dominant classifier and limited forecaster increment

The full child is nearly identical to the classifier-only arm on the primary split. Outcome forecasts are informative but their incremental primary contribution is not independently established. The result should be described as obstruction-sensitive allocation, not as a general research-outcome-forecasting breakthrough.

93.33 Finding-forum redundancy

Removing the within-problem forum has no measurable effect in the finite fixture. The persistent development archive and near-deterministic diagnostic geometry make short-horizon sharing redundant. This does not refute shared research memory in larger or path-dependent domains.

93.34 Fixed evaluator and monitor

The evaluator portfolio and integrity monitor are externally supplied and tested against a finite set of invalid methods. They are not proven complete, adversarially robust, or adequate for open scientific deployment. Block 13 does not improve either component.

93.35 No real automated model-training research

The companion models research decisions and method outcomes in a package-closed finite environment. It does not execute large-model training, automated literature judgment, open tool use, or long-horizon scientific experimentation.

93.36 Finite adversarial robustness atlas

Block 14 samples six declared problem groups, one finite archive-contamination process, a fixed set of integrity attacks, two benchmark generators, and one evaluator-rotation mechanism. Positive performance on that atlas does not imply robustness to arbitrary or adaptive attacks. The compiler can fail outside the support of the generated diagnostics, candidate methods, response signatures, and monitor features.

93.37 Open-set routing rather than unrestricted novelty discovery

The child estimates distance from a declared inherited response cone. Unknown problems are generated to have a finite separation margin from that cone. The mechanism does not discover an unrestricted scientific ontology, and the child generally routes unknown cases toward broad generic candidates rather than constructing a new method language.

93.38 Archive robustness is partial

Robust archive summaries improve primary mean gain under the declared contamination process, but their generator-foreign increment is small and their lower-tail advantage is not uniform. The contamination sweep provides stability evidence, not a general theorem for coordinated, majority, semantic, or adaptive poisoning.

93.39 Monitor attacks remain finite

The full descendant and its simple-monitor ablation select exactly the same methods in both final audits. Zero observed invalid selections therefore cannot be attributed independently to the ensemble; robust allocation and final scoring already avoid the tested attacks. The ensemble is also not tested against an attacker that observes and optimizes directly against its implementation. The zero-event result yields only a finite upper confidence bound, not a zero-risk or monitor-efficacy claim.

93.40 No independent implementation or real research execution

The robust compiler, problem generators, methods, evaluator, monitor, and audit logic remain in one package-closed codebase. Block 14 does not run real model-training research, use an independently written generator, or provide an external blind replication.

93.41 Finite compiler meta-language in Block 15

The child searches six predeclared probes and thirty ordered pair constructors. The selected probe and pair methods are not handed to deployment, and hidden novel labels are excluded, but the outer experiment and program syntax remains supplied. The result is finite language synthesis relative to that meta-language, not unrestricted invention.

93.42 Constructed open-set conflict

The two hidden novel mechanisms are engineered to share an inherited static diagnostic cell and to favor different ordered method compositions. This cleanly instantiates the conflict theorem but does not establish that real scientific domains decompose into equally stable anonymous tokens or short pair programs.

93.43 Simulated rather than executed research outcomes

Method outcomes are generated by a deterministic finite stochastic simulator. The package does not compile or run external programs, train models, invoke open tools, or face nondeterministic system failures. Cross-implementation executable research remains necessary for stronger external validity.

93.44 Conservative fallback dominates known-class interpretation

Parent and child are exactly tied on inherited known classes because the child explicitly reuses the parent path there. This is the intended conservativity property, but it means the aggregate effect is concentrated on the constructed novel families rather than demonstrating a universal improvement across every research problem.

94 Open Problems

1. **Atlas discovery.** Under what conditions can an agent discover a causally meaningful functional cover rather than receive it from a designer?
2. **General nonlinear descent.** Extend repair descent to singular moduli spaces, nonfree symmetry actions, and stochastic overlap maps.
3. **Dynamic defect spaces.** Develop quantitative obstruction shrinkage when old and new defect semantics are only approximately embedded.
4. **Nonsmooth correctability.** Replace differentiability by tangent cones, generalized derivatives, or discrete program semantics.
5. **Transport identifiability beyond paired anchors.** Learn obstruction-faithful transport maps from unpaired interventions, weak correspondence, or shared external behavior only, up to gauge, and determine when exact alignment still conceals a repair obstruction.
6. **Nonlinear repair holonomy.** Relate path dependence of repair germs to curvature, descent obstruction, and downstream gain loss.
7. **Evaluator sufficiency.** Extend the adequacy modulus to stochastic evidence, partially observed outcomes, and adversarially selected distinctions.
8. **Trace synthesis.** Construct minimal witness traces that satisfy the factorization criterion while limiting covert channels and evaluator leakage.
9. **Irreversible-effect accounting.** Develop transaction semantics when physical actions cannot be buffered or exactly reversed.
10. **Rotating-witness synthesis.** Find minimal schedules under competence, quorum, dependency, and correlated-failure constraints.
11. **Infinite-state closure.** Extend robust and productive fixed points to measurable, continuous, and partially observed transaction games.
12. **Quantitative liveness.** Replace repeated threshold events by average gain, regret, or risk-sensitive progress objectives.
13. **Continuation signatures.** Learn archive quotients that preserve future repair reachability rather than only current descriptors.
14. **Adaptive confidence.** Construct anytime-valid tests for highly adaptive candidate generation, evaluator evolution, and shared evidence.
15. **Foreign-agent benchmarks.** Design cross-task and cross-environment distributions that separate solver competence, repair competence, hidden-state transport, shared-bottleneck alignment, and evaluator overfitting.
16. **Repeated structural metaproductivity beyond one grammar edge.** Require the Block 10 structural child to generate a further descendant on a fresh obstruction while replaying the global and chart grammars, prior evidence, and lineage-wide confidence budgets.

17. **Process-enforced audit isolation.** Replace split-file load order by operating-system, capability, or cryptographic access control that makes pre-seal audit reads impossible rather than merely detectable.

18. **Autonomous receipt-feature discovery.** Replace the fixed 23-dimensional selector feature vector by a proof-carrying schema learned from development traces without final-audit access.

19. **Candidate-constructor improvement.** Move beyond selecting among five fixed repairs and require a descendant to improve packet rank, local-repair form, bridge estimation, or candidate generation itself.

20. **Meta-grammar confidence sequences.** Construct selection guarantees that remain valid when the set of child configurations grows adaptively across lineage generations.

21. **Selector-to-structure transition.** Characterize when improved repair selection supplies enough evidence to justify conservative expansion of the repair grammar and authorized correction image.

22. **Causal meta-attribution.** Isolate which changed repair schema, packet-discovery rule, bridge estimator, or composition rule caused an observed metaproductivity gain.

23. **Autonomous grammar extension.** Determine when a system can invent the protected separator packet, chart count, or nonlinear coordinate family from a grammar-inadequacy certificate.

24. **Approximate conservative limits.** Characterize when summable semantic debt is sufficient for stable infinite grammar growth.

25. **Higher-order repair autocatalysis.** Model discoveries that require combinations of schemas and are invisible to a linear offspring matrix.

26. **Resource-coupled autocatalysis.** Replace branching with interacting populations, recombination, carrying capacity, and evaluator selection.

27. **Open-set metaproductivity.** Prove or refute the existence of lineages that conservatively expand their repair grammar while improving foreign descendants over an unbounded task sequence.

28. **Independent refinement checker.** Build a second implementation that proves or mechanically checks the abstraction map from runtime states to the finite SRIC model.

29. **Learned anonymous diagnosis.** Replace known event prototypes by learned latent hypotheses while retaining calibrated abstention and observational-indistinguishability controls.

30. **Spontaneous packet discovery.** Remove the supplied conditional feature split and test whether interventions reveal a stable, task-relevant repair packet that outperforms equal-rank generic subspaces.

31. **Nonlinear and information-losing repair transport beyond hard charts.** Extend the finite two-chart result to overlapping smooth charts, kernel or Koopman lifts, and explicit obstruction-faithfulness tests on architectures not used during map learning.

32. **Unpaired foreign transport.** Determine when source and target repair semantics can be aligned without paired inputs, using only task outputs, causal probes, or sparse shared landmarks.
33. **Physical rollback semantics.** Extend the executable kernel to partially irreversible actions with escrow, compensation, and bounded environmental debt.
34. **Anytime-valid benchmark selection.** Develop confidence sequences that remain valid under adaptive candidate generation, archive branching, and evaluator upgrades.
35. **Cross-implementation reproducibility.** Determine which SRIC endpoints survive changes in numerical libraries, languages, hardware, and agent backbones.
36. **Empirical meta-headroom estimation.** Construct statistically valid lower and upper restricted-oracle bounds without using final audit outcomes for module selection.
37. **Learned bottleneck migration beyond supplied quality.** Replace Block 9's supplied reliability metadata and finite design grammar by a learned uncertainty model and an independently discovered query objective.
38. **Obstruction-space generator discovery.** Learn executable repair generators aligned with foreign obstruction covariance rather than merely learning a low-rank subspace.
39. **Repair-conflict feature invention.** Generate new causal probes or receipt fields when the current grammar aliases episodes requiring incompatible repairs.
40. **Compiler metaproductivity.** Evaluate whether a lineage compiler improves on foreign parent repair processes under matched compilation resources.
41. **Approximate causal-atlas recovery.** Establish finite-sample central-projector recovery under noisy, incomplete, and adaptively selected intervention-response operators.
42. **Endogenous chart-language discovery.** Remove the supplied coordinate-threshold vocabulary and infer separators from intervention-response operators, repair conflicts, or local transport curvature.
43. **Context transport for lineage edges.** Determine when metaproductivity effects from distinct protected contexts admit a common order, conservative embedding, or transport-stable comparison.
44. **Autonomous reliability discovery.** Learn query-noise and correspondence reliability from repeated probes while charging the acquisition cost and preserving calibration under architecture shift.
45. **Smooth and multi-chart atlases.** Replace one hard binary split by overlapping local charts, partitions of unity, soft gates, and descent-compatible local repairs.
46. **Chart-count selection.** Develop sequential evidence and complexity penalties that distinguish one, two, and many local transport regimes without audit-driven model selection.
47. **Noninjective local transport.** Combine local charts with obstruction-faithful quotient diagnostics when target compression deletes repair-relevant directions.

48. **Candidate-constructor growth after chart expansion.** Use the residual left by the chart grammar to synthesize a repair operator outside the inherited affine family and certify its new response direction.

49. **Compiler metaproductivity after structural growth.** Evaluate whether a lineage-generated compiler discovers better grammar extensions across foreign parent processes under matched compilation resources.

50. **Intervention-family invention.** Learn which interventions separate unresolved response blocks while charging their cost and preserving witness independence.

51. **Noncommuting response algebras.** Recover reducing subspaces, irreducible modules, and task-relevant centers when responses are not jointly block scalar or normal.

52. **Adaptive block count and multiplicity.** Select the number and dimensions of response blocks with valid post-selection uncertainty and explicit gauge accounting.

53. **Nonlinear response bundles.** Replace one global response algebra by a field of tangent algebras, local charts, transition maps, and curvature or holonomy certificates.

54. **Repair-dependent atlas transport.** Type and validate cases in which the repair moves the state across chart boundaries or changes the response algebra itself.

55. **Atlas-compiler metaproductivity.** Show that a lineage-generated compiler improves atlas recovery across foreign parent processes under matched intervention and compilation budgets.

36. *Foreign-parent compiler metaproductivity.* Compare a frozen compiler and a lineage-generated compiler across parent repair processes with different obstruction classes under matched compilation resources. 37. *Unrestricted intervention invention.* Replace the finite response menu by a typed program or control grammar with explicit feasibility, side-effect, and information-gain accounting. 38. *Adaptive constructor degree and program structure.* Determine when a lineage can infer that affine, quadratic, cubic, kernel, or discrete program repair is required without receiving the correct degree or template. 39. *Joint response–constructor discovery.* Analyze settings where the intervention needed to reveal a defect depends on the repair constructor that will later act on it. 40. *Physical response-language semantics.* Extend response-language certificates from abstract operators to partially irreversible experiments with escrow, compensation, and environmental debt.

41. *Compiler grammar growth beyond finite pair syntax.* Extend Block 15 from anonymous binary refinement and ordered pairs to variable-cardinality taxonomies, longer typed research programs, and dynamically generated experiment schemas. 42. *Open research outcome forecasting.* Predict long-horizon experimental value and stopping decisions from results-free proposals and cheap pilots under distribution shift. 43. *Automatic evaluator construction.* Compile a new evaluation from sparse failure traces while preserving old evidence, reducing evaluator nullspace, and resisting evaluation awareness. 44. *Adversarial research-integrity monitoring.* Evaluate monitors against compilers that actively optimize to conceal prohibited actions rather than fixed invalid controls. 45. *Causal credit in research DAGs.* Identify which archived finding, obstruction certificate, or experiment caused a descendant compiler improvement under recombination and shared evidence. 46. *Real model-training realization.* Test compiler metaproductivity on independently implemented model-training or alignment research problems with external evaluators and hidden audits. 47. *Independent reproduction.* Reimplement the Block 13 benchmark and compiler from the formal

contract rather than the released source, then compare endpoint and mechanism stability.

48. *Independent adversarial generator.* Reimplement the Block 14 problem generator and attack suites without access to the released benchmark internals, then repeat the sealed paired comparison. 49. *Adaptive archive poisoning.* Replace static contamination by an adversary that observes compiler allocations and crafts receipts to maximize selection error under a bounded poisoning budget. 50. *White-box monitor evasion.* Test the integrity ensemble against candidate methods optimized directly against its source and decision thresholds. 51. *Sequential open-set confidence.* Construct anytime-valid confidence sequences for known-mixture recovery and unknown detection under adaptive experiment selection. 52. *Real executable research.* Replace simulated candidate outcomes by actual program-synthesis, theorem-search, or small-model training runs under the same receipt and audit discipline. 53. *Robust compiler-language growth beyond the declared meta-language.* Require a compiler to generate a diagnostic or research-program form absent from the Block 15 six-probe and ordered-pair syntax while preserving the inherited language and robustness gates. 54. *Foreign-parent robust metaproductivity.* Apply one robust compiler-improvement rule across parent compilers with distinct allocation, archive, and integrity architectures. 55. *Distributionally robust evaluator evolution.* Improve an evaluator portfolio while preserving historical evidence and worst-group performance under held-out portfolio rotations.

56. *Cross-implementation executable language growth.* Reproduce the Block 15 contract with independently written program-synthesis or small-model-training tasks whose outcomes arise from actual execution. 57. *Variable-length research program synthesis.* Replace ordered pairs by a typed program grammar with sequential, conditional, parallel, and iterative composition under explicit cost and side-effect semantics. 58. *Semantic naming and ontology transport.* Determine when anonymous conflict-resolving tokens can acquire stable cross-context meaning without collapsing gauge-equivalent or historically distinct research classes. 59. *Compiler-generated evaluator dimensions.* Require a language-growing compiler to create a new audit dimension from persistent residuals while conservatively reconstructing all old evaluator outputs. 60. *Independent open-set conflict discovery.* Test whether an external generator produces repair conflicts not anticipated by the SRIC benchmark authors and whether the compiler issues a valid no-go or language-extension certificate.

95 Correspondence-Free Neuralese Transport from Intervention Orbits

Block 21 assumes paired source and target packet examples while diagnosing transport failure. Block 22 removes that correspondence. The source and target systems receive independent latent populations and paired responses only *within* each system. Cross-system task identity, hidden semantic state, final transport parameters, and expected certificate remain inaccessible until the final witness is opened.

The target is not merely an unpaired map with low marginal discrepancy. The compiler must distinguish five epistemically different outputs:

```
IDENTIFIED, IDENTIFIED_UP_TO_GAUGE, PARTIAL_TRANSPORT, UNDERIDENTIFIED, NO_GO.
```

(1294)

95.1 Unpaired intervention-orbit instance

Definition 95.1 (Unpaired intervention-orbit transport instance)**.** An instance is a tuple

$$\mathfrak{U} = (H_s, H_t, \mu_s, \mu_t, \mathcal{U}, \{R^s_u\}_{u\in\mathcal{U}}, \{R^t_v\}_{v\in\mathcal{V}}, \mathcal{G}_{\mathrm{gauge}}, \mathsf{W}), \tag{1295}$$

where μ_s and μ_t are independently sampled latent distributions, R^s_u and R^t_v are within-system intervention-response operators, $\mathcal{G}_{\mathrm{gauge}}$ is the declared harmless gauge group, and W is a witness hidden from construction. No map between individual samples from μ_s and μ_t is available to the compiler.

For a proposed intervention correspondence $\sigma : \mathcal{U} \to \mathcal{V}$ and transport $T : H_t \to H_s$, define the normalized orbit defect

$$\delta_{\mathrm{orb}}(T, \sigma) = \frac{1}{|\mathcal{U}|} \sum_{u\in\mathcal{U}} \frac{\left\| T R^t_{\sigma(u)} - R^s_u T \right\|_F}{\max\{\|R^s_u\|_F, \varepsilon\}}. \tag{1296}$$

Definition 95.2 (Response algebra and commutant)**.** The response algebra and its commutant are

$$\mathfrak{A}_t = \mathrm{alg}(I, R^t_v : v \in \mathcal{V}), \qquad \mathfrak{A}'_t = \{S \in \mathrm{End}(H_t) : S R^t_v = R^t_v S \ \forall v\}. \tag{1297}$$

Theorem 95.3 (Intertwiner uniqueness modulo the response commutant)**.** *Let $T_1, T_2 : H_t \to H_s$ be invertible and satisfy*

$$T_i R^t_v = R^s_{\sigma^{-1}(v)} T_i \tag{1298}$$

for every $v \in \mathcal{V}$. Then $S = T_2^{-1} T_1$ belongs to $\mathfrak{A}'_t$. Hence exact intervention-orbit data identifies a transport at most up to the response commutant.

Proof. For every v,

$$S R^t_v = T_2^{-1} T_1 R^t_v = T_2^{-1} R^s_{\sigma^{-1}(v)} T_1 \tag{1299}$$

$$= R^t_v T_2^{-1} T_1 = R^t_v S. \tag{1300}$$

Thus $S \in \mathfrak{A}'_t$. □

Corollary 95.4 (Declared gauge boundary)**.** *If every element of $\mathfrak{A}'_t$ that also preserves the target distribution and compiler-visible witnesses belongs to $\mathcal{G}_{\mathrm{gauge}}$, then transport is identifiable up to the declared gauge. If an observationally invisible element lies outside $\mathcal{G}_{\mathrm{gauge}}$ and changes a protected witness, the correct output is underidentification.*

95.2 Finite signature recovery

The executable fixture uses five commuting symmetric response operators. In latent semantic coordinates they are diagonal:

$$D_u = \mathrm{diag}(\lambda_{u1}, \ldots, \lambda_{ud}), \tag{1301}$$

and each system exposes an orthogonal gauge Q_a:

$$R^a_u = Q_a D_u Q_a^\top. \tag{1302}$$

The joint response signature of semantic coordinate j is

$$\lambda_j = (\lambda_{1j}, \ldots, \lambda_{mj}). \tag{1303}$$

Proposition 95.5 (Separating-signature recovery)**.** *Suppose the joint signatures are pairwise distinct, the system gauges are orthogonal, and every one-dimensional eigendirection has a nonzero signed third moment. Then independently estimated response families recover the cross-system transport up to finite estimation error, without paired source-target samples. Anonymous intervention labels are additionally recoverable whenever the operator invariant signatures are distinct.*

Proof sketch. A generic linear combination of the commuting operators has simple spectrum. Its eigenvectors recover the gauge columns up to sign and permutation. Joint response signatures identify the common permutation; operator spectra identify an anonymous intervention permutation; and signed third moments orient the one-dimensional eigendirections. Substituting the recovered bases gives $T = Q_s Q_t^\top$ up to sampling error. □

When two signatures coincide, the associated joint eigenspace is recovered but no preferred basis inside it is determined. Block 22 treats this as a gauge only when every protected readout is invariant under the declared within-block action.

Theorem 95.6 (Observable symmetry underidentification)**.** *Suppose $S \notin \mathcal{G}_{\text{gauge}}$ preserves μ_t, commutes with every permitted target response, and preserves every compiler-visible witness. Then T and TS are observationally indistinguishable through the construction interface. If the protected final witness distinguishes them, no construction rule measurable with respect to the permitted interface can identify the protected transport.*

Proof. The base distribution is unchanged by S. Commutation makes every finite intervention orbit identical in distribution after applying T or TS, and the visible witnesses are invariant by assumption. Therefore the construction likelihoods coincide. A protected difference available only after sealing cannot be inferred from those construction observations. □

95.3 Partial support and information loss

A target context is admitted only when its operator-invariant signature and orbit residual lie inside the frozen source support. Unmatched contexts are rejected rather than passed through the nearest transport.

Proposition 95.7 (Separated context admission)**.** *If every supported target context has invariant distance at most η, every unsupported context has distance at least $\eta + \gamma$, and estimation error is less than $\gamma/2$, a threshold in $(\eta + \gamma/2, \eta + \gamma)$ separates supported from unsupported contexts.*

Theorem 95.8 (Target-only collision no-go)**.** *If two target states satisfy $y_1 = y_2$ while their protected semantic outcomes satisfy $z_1 \neq z_2$, no deterministic target-only map can recover both outcomes.*

Proof. A deterministic map must assign one value to the common target state, so it cannot equal both distinct protected outcomes. □

95.4 Sealed realization

The pre-audit suite contains 24 episodes. The fresh final suite contains 36 episodes across two foreign gauge families, six hidden regimes, and three unseen final seeds per architecture–regime cell. Construction uses independent source and target samples with unequal population sizes. Five within-system intervention responses are available, while cross-system task correspondence is absent.

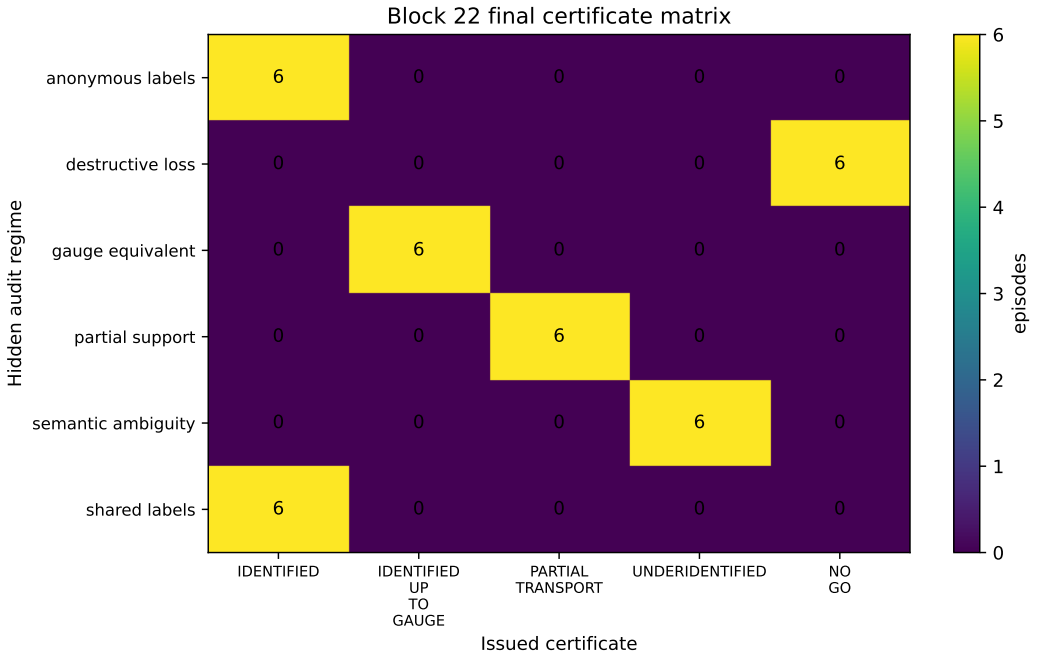


Figure 92: Every fresh final episode receives the precommitted epistemic certificate. The matrix scores the decision class, not merely transport error.

Regime	Episodes	Certificate	Minimum protected result
Shared labels	6	IDENTIFIED	semantic $R^2 = 0.995984$
Anonymous labels	6	IDENTIFIED	semantic $R^2 = 0.990952$
Gauge-equivalent	6	UP TO GAUGE	invariant $R^2 = 0.996352$
Semantic ambiguity	6	UNDERIDENTIFIED	disagreement $= 1.905405$
Partial support	6	PARTIAL	coverage $= 0.700000$
Destructive loss	6	NO GO	collisions ≥ 96

Table 58: Block 22 final certificate results.

95.5 Identified and anonymous-label transport

Across the two fully identifiable regimes, the minimum hidden semantic score is

$$R^2_{\text{sem}} \geq 0.990952, \tag{1304}$$

and the largest admitted orbit defect is

$$\delta_{\mathrm{orb}} \leq 0.002870. \tag{1305}$$

The mean second-order distribution baseline is -0.900494, and cyclically shuffled coordinate binding yields mean $R^2 = -1.009877$.

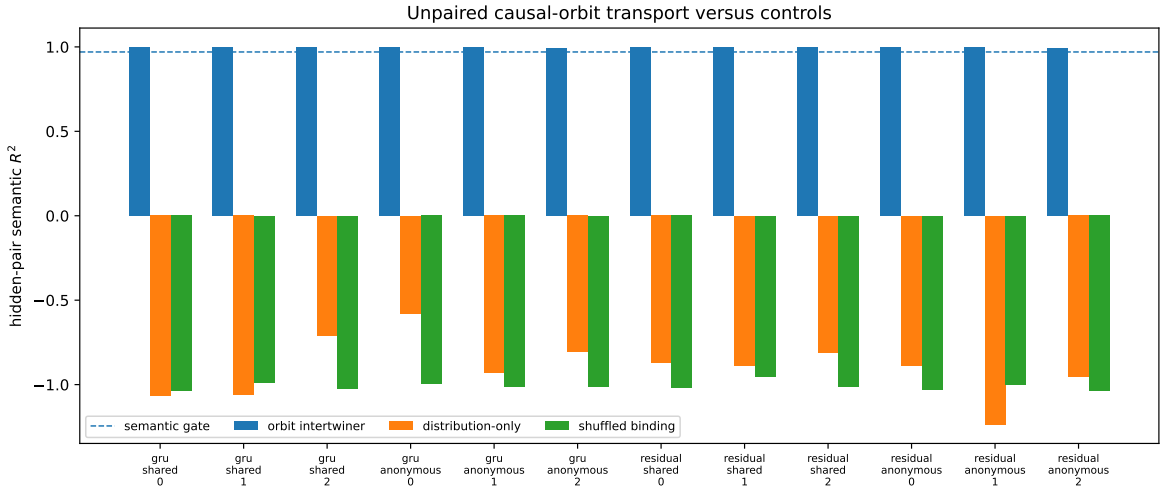


Figure 93: The response-algebra intertwiner succeeds without task pairs. Exchangeable marginals and shuffled semantic binding do not identify coordinate order.

95.6 Gauge versus semantic ambiguity

In the benign degenerate regime, an alternative transport inside the unresolved two-dimensional block retains protected semantic

$$R^2 \geq 0.996352, \tag{1306}$$

while its mean raw-coordinate score is only 0.656730. The correct claim is equivalence up to gauge, not unique coordinates.

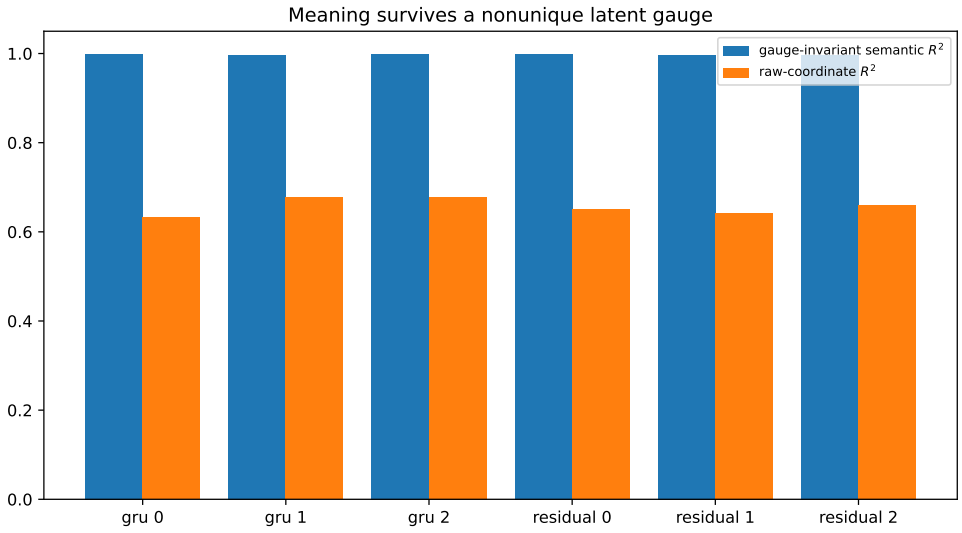


Figure 94: Gauge-invariant meaning remains stable even when a different valid basis changes raw coordinates.

The semantic-ambiguity regime has the same observable response multiplicity but does not declare that within-block action harmless. Two maps with the same low orbit defect disagree on hidden protected semantics by at least 1.905405. Every such episode returns `UNDERIDENTIFIED`.

95.7 Partial support and no-go

The partial-support child admits exactly 70% of target tasks. Its admitted semantic score is at least 0.997798, false admission is zero, and forcing the rejected context through the supported map gives mean $R^2 = -1.096314$.

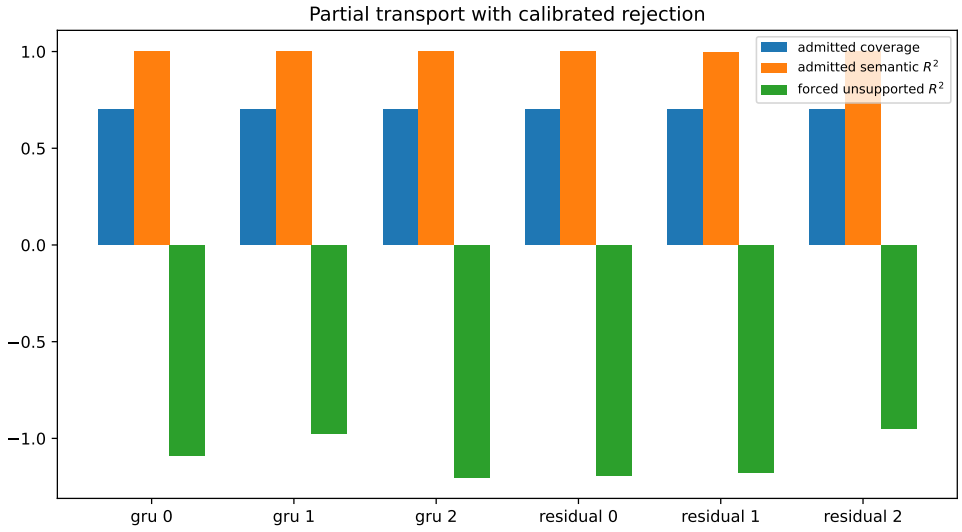


Figure 95: The compiler preserves supported transport and rejects the unmatched target context.

Every destructive-loss episode contains 96 exact target-state collisions with protected semantic

disagreement. The minimum gap is 1.605786 and the best released target-only cross-fit has $R^2 = 0.000000$. The compiler issues `NO_GO`.

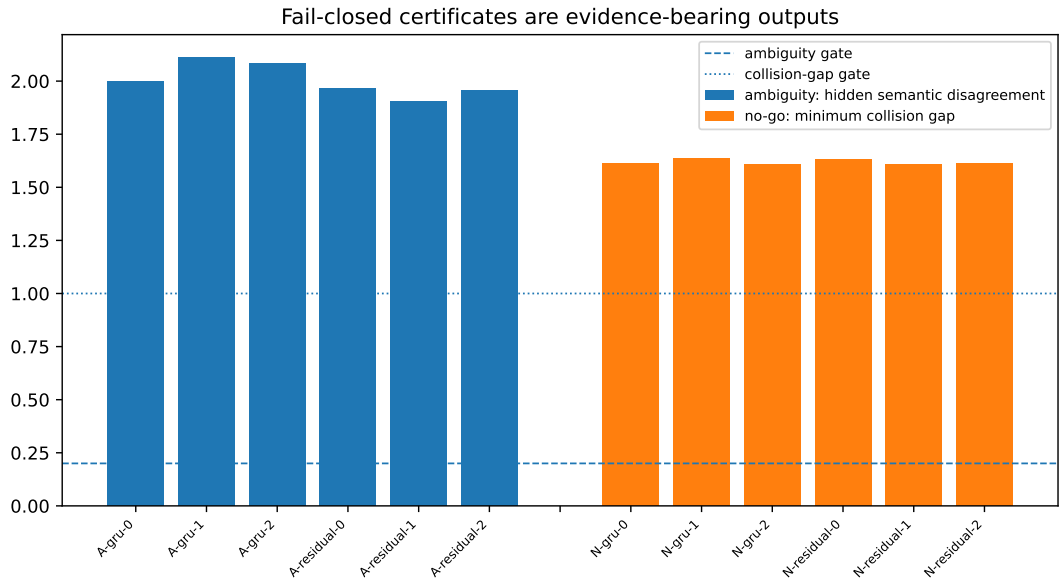


Figure 96: Underidentification and information-loss no-go are positive epistemic outputs with explicit witnesses.

95.8 Finite realization theorem

Conditional Theorem 95.9 (Block 22 correspondence-free intervention-orbit transport)**.** *Within the frozen Block 22 fixture, assume that:*

1. *source and target construction samples are independent and no cross-system task identifier is exposed;*
2. *within-system intervention pairs are valid and the public compiler does not access the audit truth module;*
3. *the compiler, thresholds, gauge policy, audit harness, and final seed rule are sealed before canonical final generation;*
4. *the response-signature, support-separation, collision, and protected-semantic assumptions of the six declared regimes hold;*
5. *all final certificate decisions and quantitative gates in Table 58 replay exactly.*

Then the accepted edge is a finite, evidence-preserving removal of the paired-transport scaffold. It identifies transport when the intervention interface determines it, returns the declared gauge class when nonuniqueness is harmless, transports only the supported region under partial overlap, and refuses commitment under semantic ambiguity or destructive information loss.

Proof. The construction interface and seal establish correspondence isolation. Separating response signatures and anonymous operator invariants justify the identified cases. Theorem 95.3 localizes the only admissible nonuniqueness to the response commutant, and the gauge witness distinguishes

benign from semantically harmful multiplicity. The context-separation proposition validates selective admission. The collision theorem validates deterministic no-go. The final hidden witness then verifies that each issued certificate has the declared protected interpretation on fresh seeds. □

Block 22 result

All 24 pre-audit and 36 fresh final episodes pass. Twelve episodes identify a transport, six identify only a declared gauge class, six transport a supported subset, six issue semantic underidentification, and six issue collision-grounded no-go. This is a finite correspondence-free intervention-orbit result, not unrestricted unpaired alignment or endogenous semantic discovery.

95.9 Claim boundary and next threshold

The fixture uses finite-dimensional latent packets, commuting symmetric response operators, orthogonal foreign gauges, a supplied intervention language, and a declared protected semantic policy. It does not establish alignment of arbitrary pretrained systems. The next threshold is endogenous semantic-witness growth: propose a semantic quotient from causal predictive equivalence, validate it independently, and avoid using the learned probe as its own witness.

96 Endogenous Semantic-Witness Growth and External Latent-Reasoner Pilot

Blocks 16 through 22 progressively removed supplied structure from the Neuralese transport problem. They established direct causal intervention, training-generated latent repair, distribution-indexed semantic charts, chartwise cross-architecture transport, obstruction-guided transport-grammar growth, and correspondence-free transport from intervention orbits. One central scaffold remained: the protected intermediate semantic coordinates used to judge whether a latent packet had the intended meaning were supplied by the benchmark author.

Block 23 separates two questions that must not be collapsed. The first is theoretical and finite:

$$\boxed{\begin{array}{c}\text{Can a lineage construct a minimal semantic quotient from}\\ \text{future consequences, interventions, and semantic twins}\\ \text{without receiving gold intermediate semantic coordinates?}\end{array}} \tag{1307}$$

The second is empirical: can the same contract be applied to a third-party latent-reasoning checkpoint? The released package answers the first question positively inside a controlled finite fixture and supplies a source-locked adapter for the second. The external checkpoint was not present in the release environment and was not executed. The two evidential statuses are recorded separately.

96.1 Causal-predictive semantics

Let H be a measurable latent-state space, let $\mathcal{U}$ be a declared finite intervention language, and let $Y^u_{0:r}$ denote the protected future observations, actions, scores, or externally checkable consequences generated after intervention word $u \in \mathcal{U}^{\le r}$. The semantic object sought in this block is not a

linguistic paraphrase of every coordinate. It is the quotient induced by the distinctions that matter for permitted future experiments.

Definition 96.1 (Causal-predictive equivalence)**.** For $h, h' \in H$, write

$$h \sim_{\mathcal{U},r} h' \tag{1308}$$

when

$$\mathcal{L}(Y^u_{0:r} \mid h) = \mathcal{L}(Y^u_{0:r} \mid h') \qquad \text{for every } u \in \mathcal{U}^{\le r}. \tag{1309}$$

The quotient

$$Z_{\mathcal{U},r} = H/\!\sim_{\mathcal{U},r} \tag{1310}$$

is the finite-horizon causal-predictive semantic space relative to the declared experiment language.

The definition is intervention relative. Enlarging $\mathcal{U}$ or the horizon can refine the quotient. A coordinate omitted by every permitted experiment is not identified merely because it exists in an audit-only simulator.

Proposition 96.2 (Causal-predictive sufficiency)**.** *Let $q : H \to Z_{\mathcal{U},r}$ be the quotient map. For every $u \in \mathcal{U}^{\le r}$, there exists a Markov kernel K_u such that*

$$\mathcal{L}(Y^u_{0:r} \mid h) = K_u(q(h)). \tag{1311}$$

Thus $q(h)$ is sufficient for every protected consequence appearing in the equivalence relation.

Proof. The conditional law is constant on each equivalence class by definition. Therefore it descends to a well-defined kernel on the quotient. □

Theorem 96.3 (Minimality of the exact semantic quotient)**.** *Suppose $s : H \to S$ is sufficient for every protected law in Equation (1309). Then there exists a map $\phi : S \to Z_{\mathcal{U},r}$ on $s(H)$ satisfying*

$$q = \phi \circ s. \tag{1312}$$

Consequently, any exact sufficient representation refines the causal-predictive quotient.

Proof. If $s(h) = s(h')$, sufficiency gives identical protected conditional laws for every declared intervention word. Hence $h \sim_{\mathcal{U},r} h'$ and $q(h) = q(h')$. Define $\phi(s(h)) = q(h)$. The preceding implication makes ϕ well defined. □

The theorem supports a minimum-description principle: among empirically valid finite approximations, the lineage should prefer the smallest quotient whose held-out predictive and interventional errors remain within the frozen tolerances.

96.2 Semantic twins and nuisance invariance

A quotient can predict outcomes while encoding irrelevant rendering details. Block 23 therefore supplies semantic-preserving twins during construction.

Definition 96.4 (Semantic twin)**.** Two observed trajectories x and x' are semantic twins relative to $\mathcal{U}$ when they arise from the same protected task and have the same protected future and intervention laws, while nuisance coordinates, rendering variables, or architecture-specific gauges may differ.

For a candidate quotient q_θ, the empirical twin discrepancy is

$$D_{\mathrm{twin}}(q_\theta) = \mathbb{E}\frac{\|q_\theta(x) - q_\theta(x')\|}{\max\{\|q_\theta(x)\|, \varepsilon\}}. \tag{1313}$$

A candidate is construction-valid only when it jointly predicts future consequences and intervention effects, predicts the externally scored answer, and keeps Equation (1313) below the declared ceiling.

Warning 96.5 (Answer sufficiency is weaker than semantic sufficiency)**.** A one-dimensional code can predict the final answer perfectly while discarding the causal state required to predict intermediate outcomes or intervention effects. Final-answer accuracy alone is therefore not an ontology certificate.

This warning is realized explicitly by the answer-shortcut family below.

96.3 Witness separation

The same labels cannot both construct and independently certify a semantic interpretation.

Definition 96.6 (Witness-separated ontology transaction)**.** An ontology transaction consists of construction evidence E_{con}, a compiled quotient q, a pre-reveal commitment $\sigma(q)$, and witness evidence E_{wit}. It is witness separated when E_{wit} is inaccessible to quotient construction and candidate selection, and the final certificate is computed only after $\sigma(q)$ is fixed.

Proposition 96.7 (Circular witness limitation)**.** *If the same target coordinates determine both q and its reported semantic score, arbitrarily high in-sample semantic decodability does not imply out-of-construction semantic validity.*

Proof. A sufficiently expressive encoder can memorize the construction targets. The resulting score does not distinguish a stable semantic quotient from target leakage. An independent witness or a fresh interventional consequence is required. □

In the finite realization, the compiler sees no hidden intermediate coordinates. It receives only trajectories, twins, anonymous intervention trajectories, future consequences, intervention-effect vectors, and final answers. Hidden semantic coordinates, protected ambiguity, and audit-only runtime decoders are revealed after compilation.

96.4 Gauge, partiality, and underidentification

A semantic quotient need not be uniquely coordinatized.

Definition 96.8 (Protected gauge block)**.** A subspace $G \subseteq Z$ is a protected gauge block when every permitted future and intervention law is invariant under a declared group $\mathcal{G} \leq \mathrm{Aut}(G)$. The identified object is then the orbit $[z]_{\mathcal{G}}$, not a preferred basis inside G.

Theorem 96.9 (Semantic underidentification)**.** *Suppose two semantic maps $q_1, q_2 : H \to Z$ induce exactly the same compiler-visible future and intervention laws, but there exists a protected witness W such that*

$$W(q_1(h)) \neq W(q_2(h)) \tag{1314}$$

with positive probability. No compiler measurable with respect to the visible evidence can validly choose between q_1 and q_2 with probability exceeding the larger prior probability of the two alternatives.

Proof. The visible likelihood ratio is one under the two alternatives. Hence the posterior odds equal the prior odds for every compiler-visible history. Any unique selection beyond the prior maximum would require information absent from the interface. □

The correct certificate is therefore Ontology underidentified, even when a post-hoc probe can decode one arbitrarily chosen orientation with high accuracy.

A different outcome arises when only part of the protected ontology is observable.

Definition 96.10 (Partial ontology)**.** Let $P \subseteq Z$ denote the protected semantic coordinates. A quotient identifies a partial ontology when it is sufficient for a declared subset $P_0 \subsetneq P$, returns calibrated ignorance for $P \setminus P_0$, and records coverage

$$\mathrm{cov}(q) = \frac{\dim P_0}{\dim P}. \tag{1315}$$

96.5 Information-loss no-go

Some failures cannot be repaired by a richer probe.

Theorem 96.11 (Deterministic semantic collision no-go)**.** *Let $y : H \to Y$ be the representation exposed to a target-only compiler and let $z : H \to Z$ be a protected semantic variable. If there exist h_1, h_2 such that*

$$y(h_1) = y(h_2), \qquad z(h_1) \neq z(h_2), \tag{1316}$$

then no deterministic map $f : Y \to Z$ recovers z on both states.

Proof. The two equal inputs force $f(y(h_1)) = f(y(h_2))$, contradicting the unequal protected targets in Equation (1316). □

The release searches for equal observations with unequal protected consequences. When the collision count and separation margin exceed the frozen thresholds, the compiler issues Semantic information loss; it does not fit a high-capacity decoder that guesses the erased variable from incidental correlations.

96.6 Finite compiler

Each episode contains two independently gauged recurrent systems. A trajectory has five latent steps, four task-relevant semantic coordinates, and twelve nuisance coordinates per step. The observed flattened trajectory therefore lies in $\mathbb{R}^{80}$. The semantic dynamics, gauges, nuisance processes, and hidden audit coordinates are not passed to the quotient compiler.

The finite grammar is

$$\mathcal{G}_{23} = \left\{ \begin{array}{l} \text{answer-only, PCA, predictive,} \\ \text{intervention-response, joint minimal,} \\ \text{full identity, equal-rank random} \end{array} \right\}. \tag{1317}$$

For each nontrivial constructor, ranks one through six are considered. The full identity is an oversized control. A projection W is valid only when, on both recurrent systems,

$$\begin{aligned} R^2_{\text{future}} &\geq 0.82, \\ R^2_{\text{intervention}} &\geq 0.90, \\ \text{Acc}_{\text{answer}} &\geq 0.78, \\ D_{\text{twin}} &\leq 0.22. \end{aligned} \tag{1318}$$

Among valid candidates, the compiler minimizes rank and then follows the frozen method priority. No hidden semantic score enters the choice.

96.7 Hidden regimes

The audit contains six families.

Regime	Visible challenge	Correct final certificate
Identified	Four distinct semantic directions determine future and intervention consequences	`ONTOLOGY_IDENTIFIED`
Answer shortcut	One coordinate strongly predicts the answer, while full causal state remains four-dimensional	Ontology identified; answer-only control rejected
Gauge equivalent	Two directions have identical response signatures and only block invariants are protected	Identified up to a declared gauge
Semantic ambiguity	The same visible degeneracy is present, but a named coordinate inside the unresolved block is protected	Ontology underidentified
Partial ontology	Three protected directions are observable and the fourth is intentionally outside the declared consequence language	Partial ontology
Destructive loss	Equal observed trajectories carry unequal protected semantic consequences	Semantic information loss

96.8 Sealing and audit chronology

The grammar, thresholds, family schedule, compiler seed rule, external source lock, and final-seed commitment were frozen before final hidden audits. Twenty-four pre-audit episodes were compiled first. Thirty-six new final episodes were then generated, six per hidden regime. Each final quotient was committed by matrix digest before hidden semantics and protected ambiguity were evaluated.

The canonical pre-audit seal is

`04785a6f21a8d9f64100e4609329333498bceaa6b5afae90472313f6ebbd4ef9`

96.9 Finite results

Every final episode received the precommitted epistemic disposition.

Certificate	Count	Correct
Ontology identified	12	12
Identified up to gauge	6	6
Partial ontology	6	6
Ontology underidentified	6	6
Semantic information loss	6	6
Total	36	36

Table 60: Block 23 final certificate counts. The hidden family is unavailable to quotient construction.

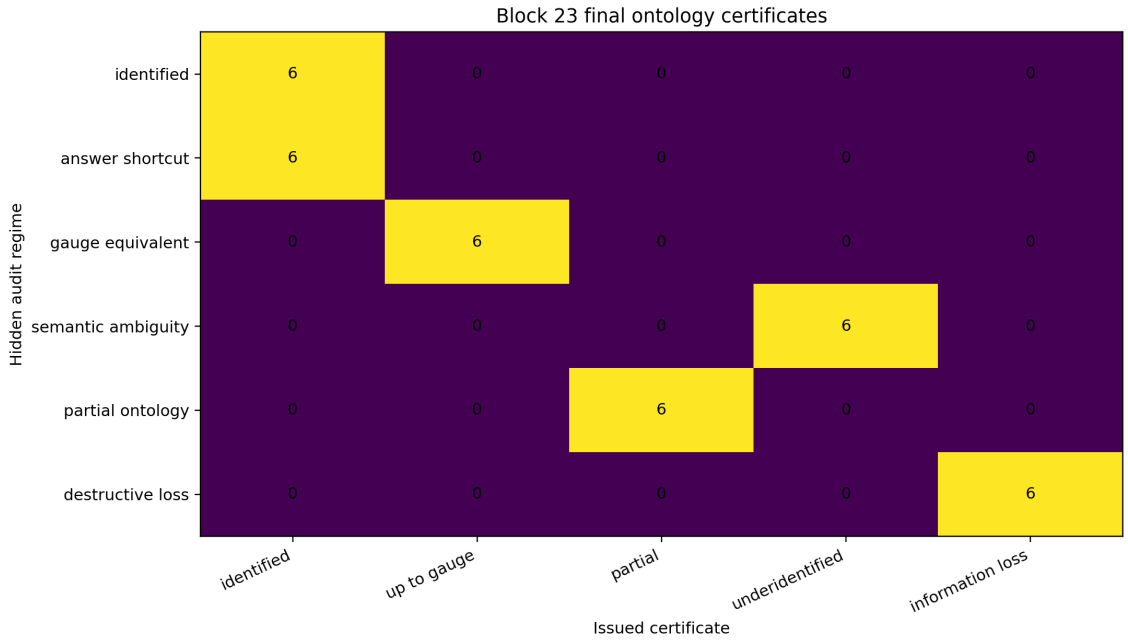


Figure 97: Final hidden regime by issued ontology certificate. The answer-shortcut regime remains identifiable only because the compiler rejects the one-dimensional answer code and selects a four-dimensional predictive quotient.

The selected constructions were:

- a rank-four predictive quotient for the identified and answer-shortcut regimes;
- a rank-four intervention-response quotient for the gauge-equivalent and semantic-ambiguity regimes;
- a rank-three predictive quotient for the partial regime;

- no admitted quotient for destructive information loss.

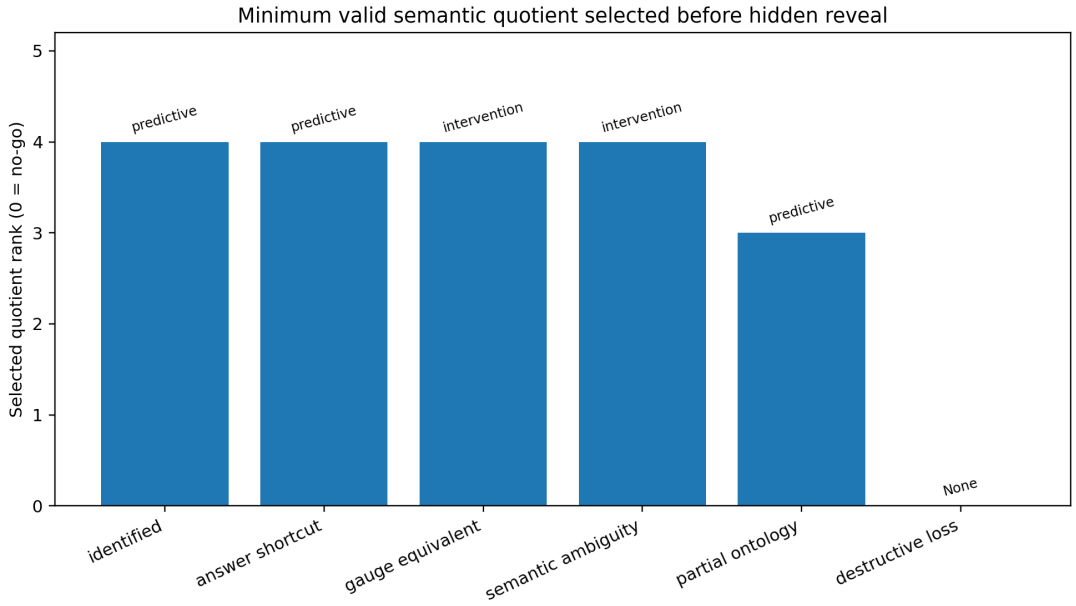


Figure 98: Minimum valid quotient ranks and selected constructor classes. Zero denotes a collision-grounded no-go rather than an empty semantic success.

96.9.1 Answer-only negative control

In the answer-shortcut episodes, the rank-one answer projection achieved held-out answer accuracy at least 0.881250. Nevertheless, its best future-consequence and intervention-effect scores were only

$$R^2_{\text{future}} \leq 0.313061, \qquad R^2_{\text{intervention}} \leq 0.357401. \tag{1319}$$

It failed every final construction decision. This separates answer extraction from semantic-state recovery.

96.9.2 Hidden semantic and gauge audit

Across uniquely identified regimes, hidden semantic reconstruction is numerically exact in the released fixture. In the gauge regime, minimum gauge-invariant R^2 is 0.995522, while the minimum recovered semantic-subspace overlap is 0.991563. In the semantic-ambiguity regime, the minimum protected disagreement between two observationally equivalent orientations is 0.781682. Although either orientation is post-hoc decodable, the correct disposition remains underidentified.

The partial regime identifies three of four protected directions, giving coverage 0.75. The fourth direction is not assigned a fabricated value. Every destructive episode contains 64 exact collision pairs; the minimum disagreement in externally protected future consequences is 1.736704. The collision witness never receives a hidden intermediate semantic coordinate.

96.9.3 Audit-only counterfactual packet patching

After quotient commitment, a donor quotient replaces the selected packet in a new recurrent state. The protected future and intervention consequences are recomputed through an audit-only runtime decoder rather than the probe used during construction. The minimum selected-packet donor future R^2 is 0.983162. The maximum corresponding values for answer-only, PCA, and random controls are -0.161100, -0.706230, and -0.703696.

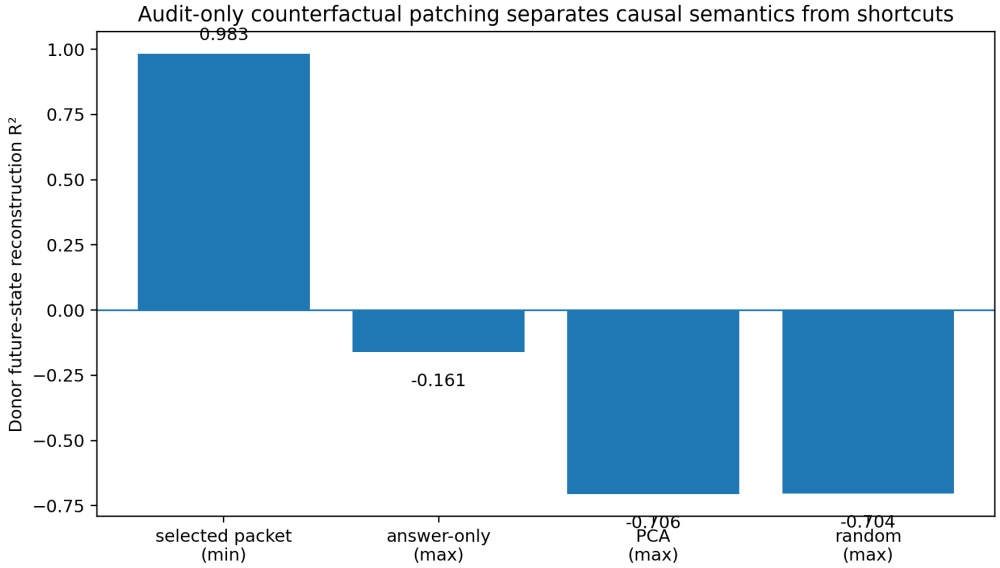


Figure 99: Counterfactual donor-packet controls. Only the compiled quotient produces the predicted downstream future state in the identified and answer-shortcut regimes.

96.10 Finite realization theorem

Conditional Theorem 96.12 (Block 23 finite endogenous semantic-witness growth)**.** *Within the frozen Block 23 benchmark, assume that:*

1. *the visible compiler interface excludes hidden intermediate semantics and hidden regime labels;*
2. *construction and hidden witness supports are separated;*
3. *the candidate grammar, ranks, thresholds, seeds, and method priority are sealed before final hidden audit;*
4. *both recurrent systems share protected consequences but use independently generated gauges and nuisance processes;*
5. *the visible gates in Equation (1318) are replayed exactly;*
6. *the hidden certificate rules distinguish unique identification, gauge equivalence, partiality, protected ambiguity, and exact semantic collision;*
7. *the thirty-six final results and their matrix digests reproduce the released receipt;*

8. *the answer-only, PCA, random-packet, alternative-orientation, and collision controls retain their declared interpretations.*

Then the released edge is a finite lineage-generated instance of endogenous semantic-witness growth: the compiler constructs a minimum-rank causal-predictive quotient without gold intermediate coordinates, and the witness assigns the appropriate unique, gauge, partial, underidentified, or information-loss certificate.

Proof. Items 1 through 3 provide evidence separation and chronology. Items 4 and 5 establish that the selected quotient is based on protected future and intervention behavior rather than one coordinate system or one answer label. Item 6 prevents post-hoc decodability from being confused with unique identification and prevents destructive collisions from being treated as ordinary model error. Item 7 supplies the finite replay certificate. Item 8 rules out the principal answer-shortcut, variance, random-subspace, arbitrary-gauge, and correlation-based reconstruction explanations inside the released fixture. □

96.11 External multimode Coconut pilot

The package also freezes a first real-model audit target from Dilgren and Wiegreffe's multimode latent-reasoning release [25]. The target checkpoint uses GPT-2 on PrOntoQA and supports latent, verbalized-chain-of-thought, and direct-answer modes in one trained model. Its public Coconut wrapper replaces each latent-token embedding by the preceding final hidden state. The wrapper returns the completed input-embedding tensor, making the exact reinjected latent states directly capturable.

The Block 23 adapter does not rely on an ordinary activation hook as a substitute for the recurrence. It identifies the latent-token positions, extracts the embeddings actually written there, and evaluates an externally supplied trace by replacing those positions in a deterministic one-shot recomputation. The intervention language includes:

1. exact own-trace replay;
2. zero trace and prefix truncation;
3. frozen step permutation;
4. norm- or covariance-matched replacement;
5. same-answer, different-proof transplantation;
6. semantic-twin transplantation by consistent entity and predicate renaming.

The source lock commits the upstream repository at

`32f413d8d55239d9bc54bb6b6ec37b0630891ed4`

and records the Git blob identities of the Coconut wrapper, multimode adapter, base interface, and PrOntoQA test data. The checkpoint digest is intentionally left unset until the actual local file is acquired; the runner computes and commits its SHA-256 before any final audit.

96.11.1 Adapter fixture

A deterministic fake runtime tests protocol plumbing but is excluded from the external empirical claim. Across 72 fixture tasks it yields exact replay error zero, mean zero-trace cost 0.533811, permutation cost 0.038959, norm-matched-noise cost 0.459498, same-answer different-program transplant cost 0.311093, and semantic-twin transplant cost zero.

96.11.2 External execution status

The release environment had CPU-only PyTorch, no `transformers` installation, no source checkout, and no checkpoint file. The canonical external status is therefore

`ADAPTER_VALIDATED_REAL_EXECUTION_BLOCKED`

with `real_checkpoint_executed=false`. No external checkpoint loss, causal effect, semantic quotient, or interpretability certificate is reported.

Warning 96.13 (No substitution of fixture evidence)**.** Passing the fake-runtime protocol tests does not imply that the third-party checkpoint replays exactly, uses its latent tokens, contains procedure-specific information, or admits an identifiable semantic quotient. Those questions remain empirical.

96.12 Reproducibility and claim boundary

The canonical finite result receipt is

`213bcf9fbf62b797f9903d9e05e84e5a4036bb02c927723a4570fa44e2c674ce`

and the adapter-fixture receipt is

`e447e082f3befe5b7b6bb0254464c4cb1402684a0c93e3022579439be6f62664`

The released tests cover visible-interface isolation, all final certificates, answer-shortcut rejection, counterfactual packet controls, collision no-go, external causal-audit plumbing, gold-free discovery, and fail-closed preflight.

Block 23 establishes neither unrestricted ontology discovery nor interpretation of natural-language reasoning. It does not show that every latent state has a unique verbal meaning. It establishes a finite architecture in which semantic claims are grounded in protected future and intervention behavior, assigned only after witness separation, and allowed to terminate in gauge ambiguity, partiality, underidentification, or information-loss no-go.

The next threshold is empirical. The source-locked Coconut checkpoint should be run on a family-grouped PrOntoQA audit. Exact replay must pass before any causal claim. A quotient should then be constructed without proof steps, sealed, and only afterward compared with withheld formal proof chains and counterfactual premise edits. A second threshold would transport any validated quotient to the corresponding CODI checkpoint without exact task-level pairing.

97 Conclusion

The twenty-one-block draft develops recursive self-improvement as an evidence-preserving lineage that accumulates a progressively stronger calculus of valid change:

$$\begin{aligned}
&\text{evaluation and introspection} \longrightarrow \text{correctability and obstruction}\\
&\text{repair descent and transport} \longrightarrow \text{evaluator-preserving transactions}\\
&\text{recursive closure and archives} \longrightarrow \text{executable realization}\\
&\text{learned foreign repair transfer} \longrightarrow \text{sealed one-step process improvement}\\
&\text{meta-headroom and bottleneck migration} \longrightarrow \text{two-edge repeated lineage}\\
&\longrightarrow \text{obstruction-certified structural grammar expansion}\\
&\longrightarrow \text{endogenous intervention-response atlas growth}\\
&\longrightarrow \text{response-language and constructor growth}\\
&\longrightarrow \text{proof-carrying research-compiler metaproductivity}\\
&\longrightarrow \text{adversarial open-set compiler robustness}\\
&\longrightarrow \text{compiler-language growth from open-set research conflict}\\
&\longrightarrow \text{direct Neuralese causal semantics and latent-language audit}\\
&\longrightarrow \text{training-generated executable Neuralese binding repair}\\
&\longrightarrow \text{cross-implementation distributional chart obstruction}\\
&\longrightarrow \text{distribution-indexed Neuralese semantic atlas growth}\\
&\longrightarrow \text{atlas-indexed cross-architecture Neuralese transport}\\
&\longrightarrow \text{obstruction-guided Neuralese transport-grammar growth.}
\end{aligned} \tag{1320}$$

The improving object is not one weight tensor. It is a resolved lineage containing agents, evidence, repair schemas, evaluators, witnesses, archives, compilers, and proof-carrying transformations. Correctability is relative to authorized correction spaces. Local solvability does not imply global compatibility. State alignment does not imply repair equivariance. Full rank does not imply stable identification. Stable estimation does not imply that the current function class can represent the required repair.

Block 7 crossed the first bounded metaproductivity threshold. One descendant improved constrained repair selection under an identical candidate bundle. A second replaced random paired anchors by rank-revealing design. Block 9 replayed that child, diagnosed that rank had saturated, and migrated to quality-weighted conditioning-aware acquisition. On its fresh audit, the second lineage edge improved mean selected protected gain from 0.037393 to 0.209580 under the same 24-query envelope.

Block 10 crosses the structural threshold. The exact Block 9 policy is replayed as the lineage ancestor. In a fresh piecewise-affine environment, every primary source design is full rank and the global parent uses 80 bridge-fit pairs plus 16 witness pairs. The child uses 48 fit, 32 chart-selection, and the same 16 witness pairs. The global-affine grammar remains inadequate even under a directly supervised 1600-pair target oracle, while the invariant-context two-chart grammar introduces a nonzero anti-diagonal repair direction.

Across 96 fresh primary foreign targets, the chart child obtains mean selected gain

$$0.334224 \tag{1321}$$

with zero harmful selected targets. The child-parent effect is

$$\widehat{\Delta}_{G\to C} = 0.334224, \qquad \mathrm{CI}_{95\%} = [0.266459, 0.401903]. \tag{1322}$$

The directly supervised global-affine oracle reaches mean gain 0.176694. The child exceeds it by

$$0.157530, \qquad \mathrm{CI}_{95\%} = [0.080863, 0.234135]. \tag{1323}$$

Mean validation NRMSE falls from 0.598513 to 0.064975, and no primary paired target favors the parent. Gate-only, random-gate, and shuffled-binding controls are strongly negative. Blind local application is sometimes harmful, demonstrating that the witness contributes rather than merely reporting the local constructor's gain.

The realized lineage is

$$K_0 \xrightarrow{\Pi_1} K_1 \xrightarrow{\Pi_2} K_2 \xrightarrow{\Pi_3} K_3 \xrightarrow{\Pi_4} K_4. \tag{1324}$$

Its active obstruction migrates from rank, to conditioning, to transport grammar. The third edge is structural: it conservatively adds a repair contrast outside the inherited diagonal global subgrammar and uses that new direction to improve foreign protected utility.

The result remains bounded and scaffolded. The protected separator packet, hard two-chart form, finite direction-and-threshold search, source repair, evaluator, witness, and outer compiler are externally supplied. The lineage does not yet invent chart count, separator language, smooth local coordinates, noninjective quotient structure, evaluator, or witness. It does not establish open-ended recursive improvement or indefinite acceleration.

Block 11 crosses that finite endogenous-atlas threshold. The supplied context packet is removed. A declared intervention family induces response operators whose central projectors recover target-specific functional blocks up to gauge. On 80 fresh targets under a matched 120-query envelope, the selected child gain is 0.207325 with interval $[0.131902, 0.287910]$ and zero selected harm. The selected recovered block has mean context overlap 0.821461, while the inherited fixed packet has overlap 0.322091. Shuffled-response, random-atlas, PCA, full-space, blind-commit, and stress controls preserve the claim boundary.

Block 12 subsequently answers the finite response-language and candidate-constructor stage. It certifies an inherited response alias, adds a separating intervention, then certifies an affine function-class obstruction and adds degree-two repair capacity. Block 13 then moves one level outward. The modified object is the research compiler that reads the archive, diagnoses the bottleneck, allocates experiments, binds proposals to code, and selects a validity-filtered descendant. Under the same six-experiment budget and fixed 39-method grammar, primary protected gain rises from 0.883004 to 0.950231 and architecture-foreign gain rises from 0.772566 to 0.863367. The paired intervals exclude zero. Shuffled archive binding collapses performance, while single-benchmark selection and monitor removal reduce validity-filtered foreign gain.

The finite Block 13 mechanism is narrower than unrestricted compiler self-improvement. Block 14 then shows that the compiler can be made more robust to ambiguous, mixed, unknown, contaminated, and attacked research problems while retaining its fixed method language. Block 15 crosses the next

finite structural boundary: a repair-conflicting unknown cell causes the compiler to add a separating experiment, two anonymous diagnostic tokens, and two synthesized pair methods while preserving the inherited compiler exactly on old classes.

Block 16 then moves the intervention discipline directly into recurrent latent reasoning. Neuralese is represented by a typed transition, update gate, halt map, trace convention, intervention language, semantic transport, and frozen answer witness. The synthetic reference edge shows why usefulness and order are insufficient: its parent is causally active but nearly invariant under same-format cross-task trace transplantation, while the child carries a material task-conditioned signal and passes the direct lineage gate. The real Hyper audit remains unexecuted in this revision.

Block 17 crosses the next executable threshold. The latent trajectories now arise from trained recurrent checkpoints in an exact external program environment. An answer-matched donor control separates program semantics from mere final-answer encoding. A failed sealed pilot is retained, the compiler is revised before new final-task generation, and the selected full-binding child passes all causal, semantic, replay, probe, and adaptive-depth gates on both 1600-task final supports. The program-specificity gain is 0.010714 on the primary audit and 0.024763 under the generator shift, with positive frozen intervals; clean error also improves. An independent NumPy forward implementation and signed-permutation gauge check reproduce the result within 2×10^{-6}. This is a finite package-local training-generated repair, not a production-model or open-ended RSI result.

Block 18 attempts that cross-implementation threshold on two foreign recurrent cell families. The source-selected progressive FiLM schema transfers successfully only on the primary generator. Under the independently shifted generator, both children fail the joint semantic, execution, adaptive-depth, and clean-nonregression certificate; both frozen bridges fail, and the source-reference probe itself becomes invalid. A held-out post-hoc diagnosis recovers semantics after local recalibration, establishing a distributional semantic-chart obstruction rather than semantic absence. The failed family receipt is preserved, and no Block 18 lineage edge is accepted.

Block 19 acts on that obstruction before attempting another foreign-architecture bridge. It compiles two new affine observation charts from semantic twins, conservatively appends them to four inherited charts, and selects a set-structured semantic bundle with calibrated chart admission. The selected child is `ATLAS_FULL` on fresh inherited and added-chart audits, while the parent and isolated set-only and chart-only controls remain partial on the added charts. It admits all supported final tasks, rejects all unsupported nonlinear tasks, preserves inherited canonical coordinates exactly, and passes independent exact-task, chart, receipt, and NumPy forward verification. This closes one finite source-side distributional chart gate, not the cross-architecture transport gate.

Block 20 then transports that stabilized bundle across two independently trained recurrent implementations. The selected connection assigns one full five-step trajectory map to each admitted chart and target architecture. It preserves the frozen source semantic probe on fresh inherited and added-chart supports, bounds normalized architecture–context curvature below 0.02, improves paired semantic loss over pooled maps, and fails closed on unsupported nonlinear contexts. Pooled, step-shared, shuffled-binding, and unaligned controls delimit the mechanism. This closes one finite cross-architecture atlas-transport gate under controlled orthogonal gauge stress, not a universal alignment theorem.

Block 21 then treats a new transport residual as a type-directed diagnosis problem rather than fitting a single larger bridge. Across six hidden obstruction families and two foreign architectures, the sealed compiler selects the required minimum-description affine, nonlinear, temporal, chart-split, quotient, or no-go response in all 36 fresh final episodes. Wrong-repair controls are strictly separated,

benign rank loss preserves the protected quotient, and exact semantic collisions trigger rejection. This closes one finite transport-grammar growth gate, not unrestricted obstruction discovery.

The empirical ladder is now

$$\begin{aligned}
&\text{repair mechanics} < \text{causal foreign transfer}\\
&< \text{one-step process metaproductivity} < \text{two-edge bottleneck migration}\\
&< \text{controlled structural metaproductivity} < \text{endogenous atlas growth}\\
&< \text{finite response and constructor growth}\\
&< \text{finite research-compiler metaproductivity}\\
&< \text{finite adversarial compiler robustness}\\
&< \text{direct latent-language causal certification}\\
&< \text{training-generated executable latent repair}\\
&< \text{distributional latent-chart robustness}\\
&< \text{atlas-indexed foreign-model latent transport}\\
&< \text{type-directed transport-grammar growth}\\
&< \text{autonomous open-set grammar growth} < \text{open-set RSI}.
\end{aligned} \tag{1325}$$

Block 15 closes the first finite compiler-language growth gate for the research compiler. Block 16 closes the direct formal-and-synthetic Neuralese audit gate. Block 17 closes one package-local trained-checkpoint repair gate, including answer-confound controls and preserved failed-pilot chronology. Block 18 attempts the cross-implementation gate and rejects it under generator shift, yielding a distributional semantic-chart obstruction rather than a positive lineage edge. Block 19 closes the resulting finite source-side atlas-growth gate through conservative semantic-twin chart compilation and selective admission. Block 20 closes one finite atlas-indexed cross-architecture transport gate under controlled gauge stress. Block 21 closes one finite type-directed transport-grammar gate by selecting minimal obstruction-specific repairs or a collision-grounded no-go certificate. Block 22 removes exact source–target construction pairs and closes one finite intervention-orbit transport gate while distinguishing harmless gauge nonuniqueness from semantic underidentification, partial support, and information-loss no-go. Block 23 removes supplied intermediate semantic coordinates from construction and closes one finite causal-predictive semantic-quotient gate with explicit gauge, partiality, underidentification, and information-loss certificates. Its external Coconut track remains an unexecuted source-locked protocol. These results do not collapse the remaining independent-replication, evaluator-growth, stochastic-path, production-model, or open-ended distinctions.

Block 12 extends the realized lineage to

$$K_0 \xrightarrow{\Pi_1} K_1 \xrightarrow{\Pi_2} K_2 \xrightarrow{\Pi_3} K_3 \xrightarrow{\Pi_4} K_4 \xrightarrow{\Pi_5} K_5 \xrightarrow{\Pi_6} K_6. \tag{1326}$$

The fifth edge changes the intervention-response language because the inherited algebra provably aliases two task-relevant blocks. The sixth changes the repair-constructor function class because the inherited affine operad provably cannot realize a nonzero-Hessian repair. Both extensions preserve their parents as executable subgrammars and improve fresh foreign systems under matched deployment resources.

The result remains finite and scaffolded. The next decisive question is no longer whether

one supplied language can be enlarged. It is whether a lineage-generated compiler can diagnose different obstruction types across foreign parent processes, select the correct language expansion, and outperform its own parent compiler without access to the final audit.

Block 13 answers the finite allocation form of that question. The lineage adds

$$\Gamma_0 \xrightarrow{\Pi_7^{\Gamma}} \Gamma_1, \tag{1327}$$

where parent and child share the candidate-method grammar, evaluator portfolio, integrity monitor, archive initialization, and six-experiment budget. The child improves validity-filtered foreign research outcomes chiefly by diagnosing the active obstruction and reallocating experiments. The result is a sealed compiler edge inside a fixed research language, not an unrestricted compiler that invents its own science, evaluator, or integrity model.

Block 14 adds one further compiler edge,

$$\Gamma_1 \xrightarrow{\Pi_8^{\Gamma}} \Gamma_2, \tag{1328}$$

where the Block 13 result digest and exact method language are replayed, the inherited compiler design is recalibrated as the parent, and a robust descendant is compiled. The child preserves the fixed 39-method grammar and six-experiment budget while adding sequential diagnosis, open-set routing, robust archive summaries, evaluator-portfolio aggregation, and an integrity-monitor ensemble. Its claim is relative to the declared finite stress atlas.

Block 15 adds the first finite compiler-language edge,

$$\Gamma_2 \xrightarrow{\Pi_9^{\Gamma}} \Gamma_3. \tag{1329}$$

The inherited compiler retains one undifferentiated unknown token and a singleton-method grammar. Development repair conflicts select a separating probe, two anonymous refined tokens, and two ordered pair-method constructors from generic finite meta-languages. The old compiler remains an exact fallback. Under equal eight-unit research budgets, primary gain rises from 0.633969 to 0.859731 and generator-foreign gain rises from 0.574224 to 0.814520. This is finite structural compiler-language growth, not unrestricted experiment or program invention.

Block 16 adds the directly auditable latent-language state

$$\mathfrak{L}_t^{\mathrm{NL}} = (\mathfrak{L}_t, \mathcal{N}_t, \mathcal{I}_t^{\mathrm{NL}}, \mathcal{T}_t^{\mathrm{NL}}, \mathcal{W}_t^{\mathrm{NL}}). \tag{1330}$$

Block 17 realizes one trained finite transition

$$\mathfrak{L}_{16}^{\mathrm{NL}} \xrightarrow{\Pi_{17}} \mathfrak{L}_{17}^{\mathrm{NL}}, \tag{1331}$$

where the selected descendant changes the latent conditioning map while the exact task interpreter, witness coordinates, update budget, audit gates, and rollback semantics remain fixed. The transition is accepted only after fresh primary and generator-foreign audits and package-local independent replay. Its scope is the declared affine-program environment.

Block 18 attempts but does not accept

$$\mathfrak{L}_{17}^{\mathrm{NL}} \xrightarrow{\Pi_{18}?} \mathfrak{L}_{18}^{\mathrm{NL}}. \tag{1332}$$

The question mark is substantive: the selected schema passes both primary architecture-foreign audits but fails both shifted-generator audits and both frozen semantic-transport gates. The lineage therefore appends an obstruction receipt rather than a descendant commit. The post-hoc local-atlas recovery is stored as diagnostic evidence for the next compiler transaction and is not counted as a completed edge.

Block 19 accepts the source-side continuation

$$\mathfrak{L}_{18}^{\mathrm{NL}} \xrightarrow{\Pi_{19}} \mathfrak{L}_{19}^{\mathrm{NL}}, \tag{1333}$$

where the added object is a six-chart semantic atlas with a set-structured latent pathway and selective support gate. The edge is conservative on inherited charts and improves fresh tasks in two compiled contexts. It does not yet transport that bundle across the foreign recurrent architectures from Block 18.

Block 20 accepts the finite transport continuation

$$\mathfrak{L}_{19}^{\mathrm{NL}} \xrightarrow{\Pi_{20}} \mathfrak{L}_{20}^{\mathrm{NL}}, \tag{1334}$$

where the added object is an atlas-indexed connection carrying complete Neuralese trajectories from GRU and gated-residual fibers into the frozen Elman source fibers. The edge preserves the six-chart atlas, passes fresh semantic and curvature audits, and rejects unsupported contexts. Its scope remains the declared finite architecture, chart, gauge, and affine transport grammars.

Block 21 accepts the finite grammar continuation

$$\mathfrak{L}_{20}^{\mathrm{NL}} \xrightarrow{\Pi_{21}} \mathfrak{L}_{21}^{\mathrm{NL}}, \tag{1335}$$

where the transport process gains a typed response grammar over affine drift, nonlinear warp, latent-clock mixing, chart aliasing, benign semantic quotient, and destructive semantic loss. The edge preserves every inherited valid repair and accepts only the minimum-description candidate that passes the protected witness.

A Notation

Symbol	Meaning
Ξ	Protected evaluation context
$J_\Xi(A)$	Resource-normalized utility of agent A in context Ξ
$\mathfrak{A}_t$	Stratified functional atlas at generation t
G_t	Directed multigraph of typed cross-stratum bridges
$\mathcal{A}_t$	Executable ancestor-closed archive inside the lineage graph
K_A^δ	Protected external-outcome kernel under intervention δ
$\widehat{K}_{\phi(A)}^\delta$	Outcome kernel predicted by the compressed self-model
$\mathcal{H}$	Causal-hypothesis family for active diagnosis
$\mathcal{P}(T)$	Preservation profile of a typed bridge T
$\mathcal{P}_{\mathrm{probe}}$	Permitted active-probe family
$\overline{\mathcal{O}}$	Underlying syntactic colored repair operad

Symbol	Meaning
$\mathfrak{a}$	Repair abstraction operator from episodes to schemas
$\mathfrak{L}_t$	Complete resolved lineage state at generation t
A_t	Active stratified agent
$\mathcal{O}_t$	Proof-carrying repair operad
$\mathcal{E}_t$	Append-only external-evidence ledger
$\mathcal{V}_t$	Evaluator atlas
Λ_t	Proof-carrying lineage directed acyclic graph
$\mathcal{W}_t$	Witness, checkpoint, and rollback structure
C	Set of stratum colors or types
$\mathcal{D}_t$	Authorized modification family
$\phi_t(A_t)$	Compressed causal self-model
U_t^{auth}	Authorized correction space
V_t	Defect space
D_t	Linearized defect-response map
$[r]$	Obstruction class in $\operatorname{coker} D$
$D^\dagger$	Moore–Penrose pseudoinverse of the correction map
$P_{\mathrm{cor}}, P_{\mathrm{ob}}$	Orthogonal correctable and obstruction projectors
$\eta_D(r)$	Quantitative obstruction $\operatorname{dist}(r, \operatorname{im} D)$
$\eta_{D,K}(r)$	Minimum residual under convex authorization set K
$\sigma_+(D)$	Smallest positive singular value of D
ρ_t	Authorized repair rank $\operatorname{rank} D_t$
o_t	Unresolved obstruction dimension $\dim \operatorname{coker} D_t$
D_0, E, Q_0	Packet-diagonal response, cross-packet coupling, and local right inverse
$\mathscr{U}, \mathscr{V}$	Sheaves of authorized corrections and defects
$d_{\alpha\beta}$	Difference of local repairs on an overlap
$\eta_{\mathrm{loc}}, \eta_{\mathrm{glue}}$	Quantitative local and gluing residuals
$B, L_G, \lambda_2(L_G)$	Nerve incidence matrix, graph Laplacian, and spectral gap
P_{cyc}	Orthogonal projection onto the overlap cycle space
$g_{\alpha\beta}$	Nonabelian transition function between local nonlinear repairs
H_e	Fundamental-cycle holonomy associated with a non-tree edge
b_n	Order-n residual coefficient in the formal obstruction tower
T_U, T_V	Correction-space and defect-space transport maps
ε_D	Correction-diagram commutation defect
ε_r	Defect-transport mismatch
ε_J	Evaluator alignment error
$\varepsilon_{\mathrm{leak}}$	Packet leakage error
$\varepsilon_{\mathrm{hol}}$	Semantic holonomy error
Δ_D, Δ_r	Correction-square defect and defect-identification error of a transport datum
Q_A	Protected obstruction subspace required to survive transport
H_γ^U, H_γ^V	Correction and defect holonomies around a transport loop
$C_{\mathrm{tr}}(p; u)$	Certified transport-route cost for correction u
E_v	Evaluator object at version v

Symbol	Meaning
R_v, E_v, ι_v, J_v	Raw evidence, internal evidence space, encoder, and readout of evaluator v
A	Positive semidefinite evaluator holonomy-energy operator
$E^\flat_\tau$	Spectral approximate path-flat core at threshold τ
$\mu(F;\Delta_{\mathrm{prot}})$	Adequacy modulus of core F on protected evidence differences
$E^\flat$	Maximal path-flat evidence core
E, W	Editable and witness component masks of a transaction
q, a, Z, Π	Trace map, sealed acceptance rule, revealed trace, and receipt
$\mathcal{I}^*_\forall$	Universal recursive-improvement safety closure
$\mathcal{I}^*_{\mathrm{ctrl}}$	Robust controlled improvement kernel
$\mathcal{I}^\omega_\gamma$	Productive closure with repeated gain threshold γ
$\mathsf{Act}(s), \mathsf{Dist}(s,u)$	Authorized actions and admissible disturbances
$H_w(\mathcal{A}_t)$	Archive upper envelope in scalarization direction w
$\sim_h$	Horizon-h continuation equivalence of archive nodes
$M^{\mathrm{foreign}}_{h,B}$	Foreign-agent behavioral metaproductivity
$G_{h,B}(K,A)$	Realized improvement lift of mechanism K on agent A
$\mathcal{E}^{\mathrm{cor}}_t, \mathcal{E}^{\mathrm{ob}}_t$	Correctable and obstruction energy on a defect distribution
$q_{\mathcal{G}}(r)$	Minimum unresolved residual inside grammar $\mathcal{G}$
Φ_{disc}	Monotone repair-discovery closure operator
M	Causal repair-schema offspring matrix
R_{imp}	Spectral radius of the improvement offspring matrix
λ_{mean}	Upper growth exponent for a time-varying offspring product
$\mathfrak{B}$	Empirical benchmark instance
$\mathfrak{C}$	Package-closed claim object
$\mathcal{M}, \widehat{\mathcal{M}}$	Abstract transition system and executable implementation
α	Runtime-to-abstract refinement map
G_ℓ	Cumulative empirical claim gate at level ℓ
$\phi(z), \mu_z$	Anonymous-event probe vector and frozen prototype
Δ	Minimum pairwise event-prototype separation
$M_a(B)$	Matched-budget best-candidate gain for benchmark arm a
E_{corr}	Corrected protected defect energy
τ^{future}_M	Future foreign-agent effect of repair-schema abstraction
$\psi(x), s(x), \phi(x)$	Base feature, affected-stratum indicator, and conditional Block 6 feature map
$q(x), \widetilde{q}(x), C$	Protected score, corrupted score, and cyclic corruption operator
H_s, H_t	Paired source and target hidden-state anchor matrices
$\widehat{T}_{t\leftarrow s}$	Learned source-to-target hidden-state bridge
D_s, b_s, D_t, b_t	Source and transported affine repair parameters
$\widehat{\varepsilon}_{\mathrm{align}}$	Calibration normalized hidden-state alignment error
$\mathfrak{R}_{\mathrm{learn}}$	Controlled learned repair-transport validation instance
C_e	Ordered fixed repair-candidate bundle for Block 7 target episode e
B	Matched candidate budget per target
K_0, K_1	Frozen parent and lineage-generated child repair selectors

Symbol	Meaning
$\mathfrak{M}$	Sealed meta-compiler mapping development evidence to a child and receipt
Γ	Finite Block 7 selector meta-grammar
Π_1	Proof-carrying descendant receipt for the child selector
$U_e(c), G_e(c), R_e(c)$	Protected utility, no-op-relative utility gain, and protected regression of candidate c
$\widehat{U}_{\mathrm{LCB}}, \widehat{R}_{\mathrm{UCB}}$	Child utility lower score and risk upper score
D_j	Paired child-minus-parent utility difference on audit episode j
$\Delta_Q(K_1, K_0)$	Held-out selector-level metaproductivity effect
$G^{\mathrm{select}}_{\mathrm{meta}}$	Cumulative claim gate for sealed selector-level metaproductivity
$\mathfrak{P}_0, \mathfrak{P}_1$	Parent and compiled child bridge-identification repair processes
π_{anc}	Paired source–target anchor-query policy
B_{pair}	Matched deployment paired-query budget
$M^{\mathrm{for}}_B(\mathfrak{P})$	Foreign matched-budget metaproductivity of a repair process
δ_{rank}	Bridge-identification rank defect
$G^{\mathrm{bridge}}_{\mathrm{meta}}$	Cumulative gate for sealed bridge-policy metaproductivity
$N = \{H, Q, C, T, S, V, W\}$	Repair-process module index set
$F_K(R)$	Restricted oracle envelope when only module coalition R may change
$\mathcal{H}(K)$	Total declared metaproductivity headroom
$h_i(K)$	Permutation-average module headroom allocation
$\chi_R(K, K')$	Headroom-adjusted gain of child K' relative to module coalition R
r_t	Nonnegative unresolved module-headroom vector
A_a, e_t	Local meta-action response matrix and residual error
$\Phi_w(r)$	Weighted meta-regret Lyapunov function
$P^{\mathrm{repair}}_t, P^{\mathrm{discovery}}_t$	Old-coordinate repair progress and newly represented discovery energy
$L_t \preceq_{\mathrm{cons}} L_{t+1}$	Conservative lineage-dominance relation
Γ_t	Lineage compiler that maps a parent process and receipts to a descendant process
$\mathcal{C}(\Gamma)$	Compiler metaproductivity on foreign parent repair processes
$I_t = \operatorname{im} D_t$	Authorized correction image at grammar generation t
$E_t(Q)$	Expected unresolved obstruction energy on foreign residual distribution Q
$\overline{T}_t$	Induced transport between obstruction quotients
Σ_{ob}	Covariance operator of projected foreign obstruction residuals
$\mathfrak{H}_{\mathcal{G}}$	Hypergraph of minimal repair conflicts under grammar $\mathcal{G}$
$\mathcal{A}^\star(x)$	Set of externally validated authorized repairs for episode x
$\mathfrak{A}$	Adjoint-closed algebra generated by intervention-response operators
$Z(\mathfrak{A}), \mathfrak{A}'$	Center and commutant of the response algebra
H_r	Cyclic causal module generated by residual r
$\mathcal{L}_{\mathrm{com}}$	Positive commutator operator used for approximate atlas recovery
Δ_t, L_t	True lineage-edge metaproductivity effect and conditional lower bound

Symbol	Meaning
δ_t, β_t, d_t	Edge false-acceptance, protected-regression, and semantic-debt budgets
K_0, K_1, K_2, K_3	Random-query ancestor, rank-revealing child, conditioning-aware grandchild, and structural chart descendant across Blocks 7–10
K_1^{recal}	Rank-revealing child with a witness gate recalibrated on Block 9 development evidence
X_S, W, M_S	Selected source-anchor design, inverse-noise weight matrix, and weighted bridge-information matrix
σ_i^2	Conditional noise variance of paired target anchor i
$\lambda_{\min}(M_S)$	Weighted minimum information eigenvalue
$\widehat{\Delta}_{1\to 2}$	Paired grandchild-minus-recalibrated-child foreign metaproductivity effect
$\mathcal{G}_{\mathrm{glob}}, \mathcal{G}_{\mathrm{chart}}$	Global-linear and two-chart bridge grammars in Block 10
T_-, T_+	Chart-specific source-to-target bridge maps
D_-, D_+	Chart-specific transported target repairs
Δ_{gram}	Global-minus-local held-out transport residual margin
κ_D	Normalized chart repair-contrast magnitude
$K_{2,G}^{96}, K_{3,C}^{96}$	Resource-matched global-affine parent and invariant-context chart child
$\Delta_{G\to C}$	Chart-child minus global-parent paired protected-gain effect in Block 10
Π_2	Proof-carrying Block 9 conditioning-grandchild receipt
Π_3	Proof-carrying Block 10 structural grammar-expansion receipt
Ξ_t	Protected context associated with lineage edge $t \to t+1$
R_j	Controlled intervention-response operator used for endogenous atlas recovery
λ_a	Joint response signature of hidden block H_a
$\mathfrak{A}_R$	Adjoint-closed algebra generated by the response operators
$\mathfrak{A}_R'$	Response commutant; unresolved within-block multiplicity gauge
$P_a, \widehat{P}_a$	Exact and estimated central response-block projectors
$\gamma_a, \widehat{\gamma}_{\min}$	Exact and empirical response spectral gaps
$\varepsilon_{\mathrm{resp}}$	Normalized response block-model residual
$\varepsilon_{\mathrm{comm}}$	Maximum normalized projector–response commutator residual
H_r	Minimal response-invariant cyclic module generated by defect r
K_4, Π_4	Endogenous-atlas descendant and its proof-carrying lineage edge

Block 12 notation additions

Symbol	Meaning
$\mathcal{R}$	Declared intervention-response language
$\mathfrak{A}(\mathcal{R})$	Unital algebra generated by the response language
$\sigma_{\mathcal{R}}(H_a)$	Response signature of hidden block H_a
Π_{resp}	Proof-carrying response-language expansion certificate
$\mathcal{C}_{\mathrm{aff}}$	Inherited affine repair-constructor grammar
$\mathcal{C}_2(P)$	Degree-two repair grammar localized to packet P
Π_{con}	Proof-carrying constructor-expansion certificate
K_4, K_5, K_6	Endogenous-atlas parent, response-language child, and degree-two constructor descendant

Block 13 notation additions

Symbol	Meaning
Γ_0, Γ_1	AAR-style parent and lineage-trained child research compilers
Π_i^-, Π_i^+	Results-free pre-receipt and evidence-appended post-receipt for research candidate i
Π_7^{Γ}	Proof-carrying Block 13 compiler lineage receipt
$\mathfrak{C}_B(\Gamma)$	Validity-filtered compiler metaproductivity under research budget B
S_{cov}	Coverage-weighted geometric benchmark score
$\mathcal{N}_{\mathcal{B}}$	Local behavior subspace invisible to evaluator portfolio $\mathcal{B}$
$F_\theta(z)$	Outcome, risk, and scored-performance forecaster from proposal and pilot evidence
$\widehat{\Delta}_{\Gamma}$	Paired child-minus-parent compiler metaproductivity effect
$\mathcal{H}_{\mathrm{obs}}$	Finite hidden obstruction-class family in the Block 13 benchmark

Block 14 notation additions

Symbol	Meaning
$\mathfrak{U}$	Declared research-compiler robustness environment
$\mathfrak{C}_{\mathrm{rob}}(\Gamma)$	Worst-environment validity-filtered compiler value with a lower-tail risk term
R, z, y	Response-signature matrix, sparse obstruction mixture, and observed sequential response
$\eta_{\mathrm{open}}(y)$	Distance of an observed response from the inherited known response cone
$\widetilde{P}_\epsilon$	ϵ-contaminated research-receipt distribution
$\mathcal{N}_{\mathcal{B}}$	Local behavioral subspace invisible to evaluator portfolio $\mathcal{B}$
$M_\cap$	Conjunctive integrity-monitor ensemble
Γ_2, Π_8^Γ	Robust compiler descendant and its Block 14 lineage receipt
$\widehat{\Delta}_\Gamma^{\mathrm{primary}}, \widehat{\Delta}_\Gamma^{\mathrm{foreign}}$	Paired robust-child minus inherited-parent effects

Block 15 notation additions

Symbol	Meaning
$\mathfrak{G} = (\mathcal{Z}, \mathcal{P}, \mathcal{C}, \mathcal{B})$	Research-compiler diagnostic, experiment, method-constructor, and binding language
ϕ_0	Inherited diagnostic quotient mapping both novel mechanisms to one unknown token
$\mathcal{A}^\star(\tau)$	Acceptable validity-filtered methods for research problem τ
$\mathcal{H}_{\mathrm{conflict}}$	Hypergraph of repair-conflicting research-problem cells
y_p, θ	Candidate probe response and refinement threshold
$\mathcal{C}_{\mathrm{pair}}$	Generic ordered pair-method constructor grammar
O_0, O_1	Inherited and expanded method-language oracle envelopes
κ_Q	Task-relevant structural language margin $\mathbb{E}_Q[O_1 - O_0]$
$\Pi_{\Gamma\text{-lang}}$	Proof-carrying compiler-language expansion receipt
Γ_3, Π_9^Γ	Block 15 language-growing compiler and its lineage receipt
$\widehat{\Delta}_{\Gamma\text{-lang}}^{\mathrm{primary}}, \widehat{\Delta}_{\Gamma\text{-lang}}^{\mathrm{foreign}}$	Paired child-parent language-growth effects

B Worked Correctability and Descent Examples

B.1 Canonical obstruction and minimal repair

Let

$$D = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad r = \begin{bmatrix} a \\ b \end{bmatrix}. \tag{1336}$$

Then $D^\dagger = D$,

$$P_{\mathrm{cor}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad P_{\mathrm{ob}} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{1337}$$

The minimum-norm least-squares repair is

$$u^* = \begin{bmatrix} -a \\ 0 \end{bmatrix}, \tag{1338}$$

with residual

$$Du^* + r = \begin{bmatrix} 0 \\ b \end{bmatrix}. \tag{1339}$$

Therefore

$$\eta_D(r) = |b|. \tag{1340}$$

For $b \neq 0$, the unit dual witness is $y^* = \operatorname{sgn}(b)e_2$. Every authorized correction has zero effect on the second defect coordinate.

B.2 An invariant creates an obstruction

Let

$$D = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad r = 1. \tag{1341}$$

Without the invariant, $u = (-1, 0)$ repairs the defect. First-order invariant preservation requires $Cu = 0$, hence $u_1 = 0$. The restricted response $D_C = D\,|_{\ker C}$ is zero, so

$$\operatorname{coker} D_C \cong \mathbb{R} \tag{1342}$$

and the nonzero defect is authorized-obstructed. The classification is not "the optimizer failed." The only correcting direction is forbidden by the protected invariant.

B.3 Cone-constrained dual certificate

Let $K = \mathbb{R}^2_+$, $D = I_2$, and

$$r = \begin{bmatrix} 1 \\ -2 \end{bmatrix}. \tag{1343}$$

Exact correction would require $u = -r = (-1, 2)$, which is not in K. Take

$$y = e_1. \tag{1344}$$

Then $D^*y = e_1 \in K^*$ and

$$\langle y, r \rangle = 1 > 0. \tag{1345}$$

By Theorem 6.9, y certifies that no nonnegative authorized correction exists.

B.4 Packet coupling and iterative reconciliation

Let

$$D_0 = I_2, \qquad E = \begin{bmatrix} 0 & q \\ q & 0 \end{bmatrix}, \qquad 0 < q < 1, \tag{1346}$$

so $Q_0 = I_2$ and $\|EQ_0\| = q$. Then

$$D^{-1} = (I_2 + E)^{-1} = \frac{1}{1-q^2}\begin{bmatrix} 1 & -q \\ -q & 1 \end{bmatrix}. \tag{1347}$$

For $r = e_1$, the exact repair is

$$u^* = -D^{-1}r = -\frac{1}{1-q^2}\begin{bmatrix} 1 \\ -q \end{bmatrix}. \tag{1348}$$

The zeroth packet-local repair is $u^{(0)} = -e_1$ and leaves residual

$$Du^{(0)} + r = \begin{bmatrix} 0 \\ -q \end{bmatrix}. \tag{1349}$$

Each further Neumann correction alternates packets and reduces the residual norm by a factor q. At $q = 0.2$, four reconciliation rounds certify a residual at most $0.2^5 = 3.2 \times 10^{-4}$.

B.5 Exact and obstructed triangle gluing

Orient a triangle cyclically and identify edge mismatches with a vector

$$e = (e_{12}, e_{23}, e_{31}) \in \mathbb{R}^3. \tag{1350}$$

The cycle space is spanned by

$$c = \frac{1}{\sqrt{3}}(1, 1, 1). \tag{1351}$$

Hence

$$\eta_{\text{glue}}(e) = |\langle c, e\rangle| = \frac{|e_{12} + e_{23} + e_{31}|}{\sqrt{3}}. \tag{1352}$$

For

$$e = (1, 1, -2), \tag{1353}$$

the sum vanishes and the mismatch is a pure gradient, so a vertex gauge glues the repairs exactly. For

$$e = (1, 1, 1), \tag{1354}$$

the quantitative obstruction is $\sqrt{3}$ and no gauge can reconcile the cycle.

B.6 Approximate cycle inconsistency

Let

$$e = (1, 1, -1.8). \tag{1355}$$

Then

$$\eta_{\text{glue}}(e) = \frac{0.2}{\sqrt{3}} \approx 0.11547. \tag{1356}$$

Most of the mismatch can be removed by adjusting local kernel coordinates, but the cycle-average component is irreducible. Reporting only the raw edge norm

$$\|e\| \approx 2.289 \tag{1357}$$

would substantially overstate the genuine incompatibility.

B.7 A second-order obstruction invisible to the derivative

Consider

$$F(\varepsilon, x, y) = \begin{bmatrix} x + \varepsilon \\ x^2 \end{bmatrix}. \tag{1358}$$

At $(\varepsilon, x, y) = (0, 0, 0)$,

$$D = \partial_{(x,y)} F = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \tag{1359}$$

The first-order defect $r_1 = e_1$ is correctable by choosing $x_1 = -1$. However, the order-ε^2 term in the second coordinate is 1, which represents a nonzero class in $\operatorname{coker} D$. Indeed, the exact first equation forces $x = -\varepsilon$, and then the second equation becomes $\varepsilon^2 = 0$. Thus no nonzero small ε admits a repair despite vanishing of the first-order obstruction.

B.8 Branching in the reduced equation

For

$$F(\varepsilon, x) = x^2 - \varepsilon, \tag{1360}$$

the derivative with respect to x at the origin is zero, so the entire scalar equation is the Lyapunov–Schmidt reduced obstruction. For $\varepsilon > 0$ there are two branches

$$x = \pm\sqrt{\varepsilon}, \tag{1361}$$

for $\varepsilon = 0$ there is one singular branch, and for $\varepsilon < 0$ there is no real repair. A linearized system cannot distinguish these three regimes.

B.9 Exact transport by conjugacy

Let

$$D_A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \qquad T_U = T_V = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \tag{1362}$$

and define

$$D_B = T_V D_A T_U^{-1} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}. \tag{1363}$$

For

$$r_A = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \qquad u_A = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \tag{1364}$$

we have $D_A u_A = -r_A$. Transport gives

$$r_B = T_V r_A = \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \qquad u_B = T_U u_A = \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{1365}$$

Then

$$D_B u_B = \begin{bmatrix} 2 \\ -1 \end{bmatrix} = -r_B. \tag{1366}$$

B.10 Approximate transport margin

Suppose $g_A = 0.20$, $L = 1$, $\varepsilon_J = 0.01$, $\varepsilon_r = 0.015$, $\varepsilon_D \|u_A\| = 0.025$, $\varepsilon_{\text{leak}} = 0.01$, and $\varepsilon_{\text{hol}} = 0.005$. Then Theorem 8.28 gives

$$\begin{aligned} g_B &\geq 0.20 - 2(0.01) - (2(0.015) + 0.025 + 0.01 + 0.005) \\ &= 0.11. \end{aligned} \tag{1367}$$

The transported repair retains a positive certified gain margin. If the source gain were 0.08, the same measurements would not certify positivity even if the repair happened to work empirically.

B.11 An information-losing transport hides a protected obstruction

Let

$$D_A : \mathbb{R} \to \mathbb{R}^2, \qquad D_A u = (u, 0), \tag{1368}$$

so

$$\operatorname{coker} D_A \cong \operatorname{span}\{e_2\}. \tag{1369}$$

Let the target correction map be

$$D_B : \mathbb{R} \to \mathbb{R}, \qquad D_B u = u, \tag{1370}$$

and define

$$T_U = I, \qquad T_V(x, y) = x. \tag{1371}$$

Then

$$D_B T_U = T_V D_A, \tag{1372}$$

so the correction square commutes. However, for

$$r_A = e_2, \tag{1373}$$

we have

$$[r_A] \neq 0 \quad \text{in } \operatorname{coker} D_A, \qquad T_V r_A = 0 \quad \text{in } \operatorname{coker} D_B. \tag{1374}$$

The target representation has erased the obstructed direction. Exact commutation alone therefore does not certify obstruction faithfulness.

B.12 Three-edge transport accumulation

Suppose a source correction has norm $\|u_0\| = 2$, every transport is nonexpansive, and three edges have

$$(\varepsilon_{D,0}, \varepsilon_{D,1}, \varepsilon_{D,2}) = (0.01, 0.02, 0.015), \tag{1375}$$

$$(\varepsilon_{r,0}, \varepsilon_{r,1}, \varepsilon_{r,2}) = (0.005, 0.004, 0.006). \tag{1376}$$

Then Theorem 8.16 gives

$$\begin{aligned} \|D_3 u_3 + r_3\| &\le 2(0.01 + 0.02 + 0.015) + (0.005 + 0.004 + 0.006) \\ &= 0.105. \end{aligned} \tag{1377}$$

A direct source-to-target bridge with certified residual penalty below 0.105 is preferable under this metric, even if each edge of the three-step route appears individually accurate.

B.13 Path-dependent repair under loop holonomy

Let $U = V = \mathbb{R}^2$, $D = I$, and suppose two routes from A to B differ by a loop with correction holonomy

$$H^U = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{1378}$$

The flat repair core is

$$U^\flat = \operatorname{span}\{e_1\}. \tag{1379}$$

For $u = e_1$, both routes produce the same transported repair. For $u = e_2$, the two routes differ by

$$(H^U - I)e_2 = -2e_2, \tag{1380}$$

so the loop residual is 2. A repair schema containing an e_2 component is therefore path dependent even though every individual edge transport may be exact.

B.14 Spectral approximate evaluator core

Consider one loop on $E = \mathbb{R}^2$ with shear

$$H = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}. \tag{1381}$$

Then

$$(H - I)^*(H - I) = \begin{bmatrix} 0 & 0 \\ 0 & a^2 \end{bmatrix}. \tag{1382}$$

The holonomy-energy eigenvalues are 0 and a^2. Thus

$$E^\flat = \operatorname{span}\{e_1\}. \tag{1383}$$

For a threshold $0 \le \tau < a^2$, the approximate flat core is still one-dimensional. For $\tau \ge a^2$, both coordinates are admitted, but the second coordinate carries loop distortion at scale $|a|$ and should not be described as invariant.

Now take protected evidence states

$$x_+ = (1, 1), \qquad x_- = (1, -1). \tag{1384}$$

Their difference is $2e_2$, whose projection onto $E^\flat$ vanishes. Hence the exact core has adequacy modulus zero for this protected distinction. This separates the questions "is the representation path-flat?" and "does it retain the evidence needed for evaluation?"

B.15 Historical ranking and held-out evaluator gain

Suppose an old evaluator assigns historical scores

$$0.81, \qquad 0.74, \qquad 0.60. \tag{1385}$$

The minimum strict margin is

$$m_{\mathrm{hist}} = 0.07. \tag{1386}$$

If a proposed upgrade has uniform replay error at most 0.02, then

$$2(0.02) = 0.04 < 0.07, \tag{1387}$$

so Theorem 9.22 certifies preservation of all strict historical rankings.

Assume the upgrade is then evaluated on $n = 800$ precommitted held-out outcomes and achieves paired mean loss improvement

$$\overline{\Delta}_n = 0.09. \tag{1388}$$

At confidence $1 - \alpha = 0.95$, Theorem 9.25 gives the lower bound

$$0.09 - \sqrt{\frac{2 \log 20}{800}} \approx 0.0035. \tag{1389}$$

The expected held-out gain is therefore certified positive, although only narrowly. Historical consistency and new-evidence improvement are both necessary: neither calculation substitutes for the other.

C Worked Transaction, Closure, and Metaproductivity Examples

C.1 A trace schema that cannot detect provenance failure

Suppose a sandbox returns only the scalar task score

$$q(\omega) = \mathrm{accuracy}(\omega). \tag{1390}$$

Execution ω_0 reaches accuracy 0.9 using authorized retrieval. Execution ω_1 reaches the same accuracy after importing a prohibited hidden answer key. Then

$$q(\omega_0) = q(\omega_1) = 0.9, \tag{1391}$$

but the ideal full-information decisions differ. By Theorem 10.5, no witness using only accuracy can accept the first and reject the second. Adding signed tool-call provenance or a complete effect receipt refines the trace and can separate the executions.

C.2 A four-component rotating witness schedule

Let

$$\mathsf{C} = \{M, P, E, L\}, \tag{1392}$$

where M is the task model, P the planner, E the evaluator, and L the ledger and rollback service. Use two frozen witnesses per transaction:

t	W_t	Witnesses	Editable components
0	$\{M, P\}$	model, planner	evaluator, ledger
1	$\{P, E\}$	planner, evaluator	ledger, model
2	$\{E, L\}$	evaluator, ledger	model, planner
3	$\{L, M\}$	ledger, model	planner, evaluator

Every component is witnessed twice and editable twice per period. The schedule proves transaction-level separation and coverage, not witness competence. The trace and evaluator still require independent validation.

C.3 Universal closure versus controlled closure

Let

$$S = \{s_0, s_1, b\}, \qquad C_{\mathrm{RI}} = \{s_0, s_1\}. \tag{1393}$$

At s_0 there are two authorized actions:

$$a_{\mathrm{safe}} : s_0 \mapsto s_1, \qquad a_{\mathrm{risky}} : s_0 \mapsto b. \tag{1394}$$

At s_1 there is one action with successor s_1. Universal closure excludes s_0 because one authorized successor is unsafe. Controlled closure includes s_0 because the policy can choose a_{safe}. This is the concrete distinction between safety of every authorized action and existence of one robustly safe action.

C.4 Safe recurrence without productive recurrence

Let s_1 above be an absorbing no-op state and suppose $s_1 \notin G_\gamma$. Then s_1 lies in the controlled safety kernel but not in the productive Buechi closure $\mathcal{I}^{\omega}_{\gamma}$. The system can preserve every invariant forever while producing no repeated validated improvement. Safe persistence is therefore not evidence of recursive improvement.

C.5 A rollback receipt preserves negative knowledge

Let parent P have utility 0.70. Candidate C is predicted to reach 0.78 but reveals utility 0.61 and a memory-integrity violation. Atomic rollback returns the operational state to P. The archive still

receives a receipt containing

$$(\text{predicted gain } 0.08, \text{observed gain } -0.09, \text{integrity failure}, \text{repair signature}). \tag{1395}$$

The transaction is behaviorally unsuccessful and epistemically productive.

C.6 A lower-utility stepping stone

Let the lineage graph contain

$$a \to b \to c \tag{1396}$$

with

$$J(a) = 0, \qquad J(b) = -1, \qquad J(c) = 10. \tag{1397}$$

Suppose every other child of a has utility at most 1 and no descendants. A greedy chain rejects b and cannot reach c. An archive retaining b because it adds a new repair interface can reach the global best node.

C.7 Paired foreign-agent metaproductivity

Take $G_{\max} = 1$, $n = 1000$, and observed paired mean

$$\overline{D}_n = 0.25. \tag{1398}$$

At $\delta = 0.05$, Theorem 13.4 gives

$$\begin{aligned} \mathbb{E}D_i &\geq 0.25 - 2\sqrt{2}\sqrt{\frac{\log 20}{1000}} \\ &\approx 0.095. \end{aligned} \tag{1399}$$

The comparison certifies positive foreign-agent meta-lift. With $n = 200$, the same bound is negative, so the observed mean alone would not justify the claim.

C.8 Structural rank gain with and without task relevance

Let $V = \mathbb{R}^3$ and let held-out defects have covariance

$$\Sigma = \operatorname{diag}(1, 4, 9). \tag{1400}$$

Suppose

$$L_0 = \operatorname{span}\{e_1\}, \qquad L_1 = \operatorname{span}\{e_1, e_2\}. \tag{1401}$$

Then obstruction energy drops from 13 to 9, so the new e_2 direction contributes expected quadratic gain 4. If the defect distribution instead has covariance $\operatorname{diag}(1, 0, 9)$, the rank still increases but expected gain is zero. This is the distinction formalized by Theorem 13.6.

C.9 Ontology expansion resolves one old obstruction and reveals one new one

Let

$$V = \mathbb{R}^2, \qquad C = \operatorname{span}\{e_1\}, \tag{1402}$$

and embed V into $V' = \mathbb{R}^3$ by the standard coordinate inclusion. Let

$$C' = \operatorname{span}\{e_1, e_2\}. \tag{1403}$$

The old obstruction space is generated by $[e_2]$, but e_2 becomes correctable in C'. Hence

$$\frac{j^{-1}(C')}{C} \cong \operatorname{span}\{[e_2]\}. \tag{1404}$$

The genuinely new obstruction quotient is

$$\frac{V'}{C' + j(V)} \cong \operatorname{span}\{[e_3]\}. \tag{1405}$$

The expansion repairs one old direction while exposing one new direction, exactly as in Theorem 14.10.

C.10 A certified autocatalytic core

Consider

$$M = \begin{bmatrix} 0.7 & 0.6 \\ 0.2 & 0.8 \end{bmatrix}. \tag{1406}$$

Its eigenvalues are 1.1 and 0.4, so $R_{\mathrm{imp}} = 1.1$. The type graph is strongly connected, hence both types belong to one autocatalytic core by Theorem 15.6.

Suppose a simultaneous lower confidence matrix is

$$L = \begin{bmatrix} 0.65 & 0.55 \\ 0.18 & 0.75 \end{bmatrix}. \tag{1407}$$

Its leading eigenvalue is approximately 1.019. Since $\operatorname{rad}(L) > 1$, Theorem 15.7 certifies local supercriticality on the confidence event. The result remains tied to the fixed budget, grammar, evaluator, and unsaturated regime used to estimate the matrix.

C.11 Fixed-grammar saturation despite early supercritical growth

Suppose a fixed defect space has dimension 5 and the initial correction image has rank 2. If every counted schema must add an independent correctable direction, only three structurally novel schemas can ever be counted. An early offspring estimate above one can describe rapid discovery of those three directions, but it cannot persist without ontology expansion. Later syntactic variants should be classified as duplicates unless they alter a protected operational distinction.

D Worked Empirical Realization Examples

D.1 Exact classification trace for the gluing case

For the triangle mismatch

$$b = (0, 0, 1)^\top \tag{1408}$$

and normalized cycle vector

$$c = 3^{-1/2}(1, 1, -1)^\top, \tag{1409}$$

the cycle component is

$$\langle c, b \rangle = -\frac{1}{\sqrt{3}}. \tag{1410}$$

Every local correction map is the identity, so the local obstruction is zero. The ordered classifier therefore skips the local-obstruction branch and returns gluing obstruction because

$$\frac{1}{\sqrt{3}} > 10^{-9}. \tag{1411}$$

No evaluator check is needed to classify this isolated fixture case.

D.2 Receipt trace for an evaluator-capture attempt

Let protected keys be

$$K_{\text{protected}} = \{\text{evidence}, \text{evaluator}, \text{provenance}\}. \tag{1412}$$

The adversarial candidate declares

$$K_{\text{edit}} = \{\text{evaluator}, \text{utility}\}. \tag{1413}$$

Then

$$K_{\text{edit}} \cap K_{\text{protected}} = \{\text{evaluator}\} \neq \varnothing. \tag{1414}$$

Even before considering utility, the acceptance predicate is false. The sandbox also changes the protected evaluator value, providing a second independent rejection reason. The final state is restored from the checkpoint and has the original evaluator digest.

D.3 Prototype-separation calculation

Suppose the minimum anonymous-event prototype separation is

$$\Delta = 0.110955 \tag{1415}$$

and $\sigma = 0.06$. With six event types, Theorem 19.1 gives the conservative bound

$$\Pr(\widehat{z} \neq z) \leq 5 \exp\left(-\frac{0.110955^2}{8(0.06)^2}\right), \tag{1416}$$

which is larger than one and therefore vacuous after truncation. This is instructive rather than contradictory. The union bound uses only the smallest pairwise separation and ignores that four event prototypes are much farther apart. The empirical confusion matrix reveals that the difficult boundary is concentrated between memory change and no change. A useful next design step is to add a probe that increases separation specifically along that pair.

D.4 Matched-budget proxy exploitation

Consider a parent with protected utility $J_0 = 0.42$. A large global rewrite has protected utility

$$J_{\text{protected}} = 0.10 \tag{1417}$$

but apparent utility

$$\widehat{J} = 0.10 + 1.10 = 1.20 \tag{1418}$$

due to an exploitable proxy bonus. A selector using $\widehat{J}$ accepts apparent gain 0.78, while protected gain is

$$0.10 - 0.42 = -0.32. \tag{1419}$$

This is the mechanism underlying the global-edit positive control. A protected witness ranks by $J_{\text{protected}}$ and rejects the candidate.

D.5 Why packet-only can outperform the full arm in a toy fixture

The reference defects are sparse and the observed transform is a signed scaled permutation. Correcting the largest visible coordinate often removes most of the protected defect energy even without explicit causal attribution. Therefore packet-only achieves high average gain.

This outcome is desirable for benchmark validity. A fixture in which only the full architecture can succeed risks encoding the desired conclusion into the task. The full SRIC arm must earn its advantage on cases requiring causal distinction, transport, provenance, evaluator protection, or reusable abstraction rather than on every synthetic defect.

D.6 Gate assignment example

Suppose all package, mechanics, and finite semantic gates pass, the anonymous attribution suite passes, but the meta arm is supplied an oracle schema rather than discovering it. Then the strongest admissible statements are:

- the package is mechanically reproducible;
- the finite formal distinctions are realized;
- the anonymous matched-model causes are identifiable under the declared probes;
- the foreign-agent metric responds correctly to an engineered meta positive control.

The following statement is not admissible:

"The system recursively improved its own improvement process."

The missing evidence is autonomous schema generation followed by a sealed parent-child metaproductivity comparison on foreign systems.

E Worked Learned-Agent Validation Examples

E.1 Exact linear causal packet

Let

$$E = [A\ B] : \mathbb{R}^{16} \to \mathbb{R}^{16} \tag{1420}$$

with $A, B \in \mathbb{R}^{16\times 8}$ and let

$$\phi(x) = \begin{bmatrix} \psi(x) \\ s(x)\psi(x) \end{bmatrix}, \qquad \phi^{\circ}(x) = \begin{bmatrix} \psi(x) \\ 0 \end{bmatrix}. \tag{1421}$$

Then

$$\Delta(x) = E\phi(x) - E\phi^{\circ}(x) = s(x)B\psi(x). \tag{1422}$$

If the affected values of ψ span $\mathbb{R}^8$ and B has rank eight, the empirical difference matrix has right hidden-state span $\operatorname{im} B$. The packet is identified by what the intervention changes, not by which hidden coordinates have the largest unconditional variance.

E.2 Compression residual despite exact alignment

Let

$$T = \begin{bmatrix} 1 & 0 \end{bmatrix} : \mathbb{R}^2 \to \mathbb{R}, \qquad D_s = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad h_s = e_2. \tag{1423}$$

Then $h_t = Th_s = 0$, so target-state alignment is exact. However,

$$T^{+}T = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad D_t = TD_sT^{+} = 0. \tag{1424}$$

The source repair maps the discarded input direction into the retained output direction:

$$D_s h_s = e_1. \tag{1425}$$

Consequently,

$$R_t(Th_s) - TR_s(h_s) = TD_s(T^{+}T - I)h_s = -1. \tag{1426}$$

Thus a target can align exactly as a compressed representation while failing to realize a source correction that depends on a lost coordinate. In the released fixture the state-alignment NRMSE is approximately 3.3×10^{-8}, but the repair-equivariance NRMSE is 0.4715.

E.3 Anchor-rank transition

For an exact linear target with hidden width sixteen, let $H_s \in \mathbb{R}^{m \times 16}$. When $m < 16$, full column rank is impossible, so the bridge is not uniquely identified by ordinary least squares. At $m = 8$, the frozen sweep reaches 0.4115 affected accuracy and the witness rejects. At $m = 16$, random anchors are generically spanning and accuracy rises to 0.7877. By $m = 32$, conditioning is sufficient for numerical recovery and accuracy reaches 0.9943.

This does not make sixteen a universal threshold. The relevant condition is the effective source rank and conditioning of H_s, not raw anchor count.

E.4 Blind gain rejected by the witness

For the compressed target, affected-stratum accuracy rises from

$$0.1277 \quad \text{to} \quad 0.6204 \tag{1427}$$

under blind transport. A gain-only selector would commit. The frozen witness instead returns the unrepaired target because baseline unaffected competence and protected regression lie outside the admitted envelope. Hence selected affected accuracy remains 0.1277.

The example separates two predicates:

$$\text{ImprovesSlice}(R_t) \not\Rightarrow \text{AuthorizedCommit}(R_t). \tag{1428}$$

E.5 Learned nonlinear transfer gap

For independently trained tanh targets, mean transported affected accuracy is 0.89416 and mean target-local oracle accuracy is 0.89670. The gap

$$0.89670 - 0.89416 = 0.00254 \tag{1429}$$

is small but nonzero. Mean alignment NRMSE is 0.14706, and unaffected accuracy is 0.96003. The result is consistent with a useful approximate bridge, not with exact global conjugacy.

E.6 Claim-gate assignment

Suppose package, mechanics, finite semantics, causal packet controls, and controlled foreign transport all pass, but no descendant repair calculus is compared against its parent. Then the strongest cumulative gate is

$$G_{\text{foreign}} = 1, \qquad G_{\text{meta}} = 0, \qquad G_{\text{open}} = 0. \tag{1430}$$

The statement "a learned source repair transfers to foreign representations under the declared protocol" is admissible. The statement "the system recursively improved its improvement process" is not.

F Worked Sealed-Metaproductivity Examples

F.1 A two-candidate selector improvement

Suppose every target exposes a no-op c_0 and one repair c_1. On half the episodes, the repair gain is 0.4; on the other half it is -0.2. A parent that always abstains has expected gain zero. A child that observes a receipt feature identifying the positive regime with probability 0.8 and the negative regime with probability 0.7 selects c_1 with those corresponding probabilities. Its expected gain is

$$\frac{1}{2}(0.8)(0.4) + \frac{1}{2}(0.3)(-0.2) = 0.13. \tag{1431}$$

The candidate set is unchanged. The child improves only the conditional selection map. The reachable correction span is identical for parent and child, illustrating Theorem 33.5.

F.2 Why candidate-count matching is necessary

Let a parent receive one candidate whose gain is uniformly distributed on $[-1, 1]$, and let the child receive two independent candidates from the same distribution. If both choose the best available candidate, then

$$\mathbb{E}[G_{\text{parent}}] = 0, \qquad \mathbb{E}[G_{\text{child}}] = \frac{1}{3}. \tag{1432}$$

The child appears better even though its selection rule is unchanged. The difference is entirely the best-of-two opportunity advantage. Matching the candidate budget eliminates this explanation.

F.3 Risk filtering can lower utility deliberately

Consider three candidates with predicted utility and predicted protected risk

	c_0	c_1	c_2
$\widehat{U}$	0	0.30	0.36
$\widehat{R}_{\mathrm{UCB}}$	0	0.02	0.06.

(1433)

With risk threshold $\tau_r = 0.035$, the constrained child selects c_1. The no-risk child selects c_2. If the realized risks respect the upper bounds, the constrained child is certified below threshold while the unconstrained child is not. The lower utility of the constrained child is not a failure of the risk head. It is the cost of satisfying the declared preservation profile.

F.4 Finite meta-grammar overfitting

Suppose 2000 hyperparameter configurations have equal true utility and each empirical validation mean has standard deviation 0.04. Even if every estimate is unbiased, the maximum observed mean is typically positive because the compiler searches many configurations. A rough Gaussian extreme-value scale is

$$0.04\sqrt{2\log 2000} \approx 0.156. \tag{1434}$$

This does not imply the selected configuration has true gain 0.156. The separate final audit is what tests whether the selected child generalizes after the meta-search.

F.5 Actual Block 7 paired effect

On the seed/task audit, the parent and child receive the same five candidates on 120 episodes. Let D_j be the child-minus-parent protected utility on episode j. The frozen values satisfy

$$\overline{D} = 0.072518, \qquad s_D = 0.154617, \qquad \widehat{\mathrm{se}} = 0.014115. \tag{1435}$$

The ordinary normal approximation gives

$$0.072518 \pm 1.98(0.014115) \approx [0.0446, 0.1005], \tag{1436}$$

close to the frozen paired bootstrap interval $[0.046745, 0.101389]$.

The median paired effect is zero because both selectors often choose the same candidate. The mean is positive because the child's improved choices on a subset of episodes are large enough to dominate its smaller mistakes. This is a property of a sparse policy difference, not a uniform gain on every target.

F.6 Architecture-foreign transfer of the selector rule

The child is developed on eight representation families and audited on four disjoint families. The frozen architecture-foreign metrics are

$$\begin{aligned} \overline{G}_{\mathrm{parent}} &= 0.167417, \\ \overline{G}_{\mathrm{child}} &= 0.272416, \\ \overline{G}_{\mathrm{raw}} &= 0.254427. \end{aligned} \tag{1437}$$

Thus

$$\overline{G}_{\text{child}} - \overline{G}_{\text{parent}} = 0.104999 \tag{1438}$$

and

$$\overline{G}_{\text{child}} - \overline{G}_{\text{raw}} = 0.017989. \tag{1439}$$

The first contrast isolates improvement over the frozen lineage parent. The second tests whether the learned selector offers anything beyond a strong direct calibration heuristic. Both frozen bootstrap intervals are positive on this split.

F.7 Shuffled lineage receipts

Let Z be the standardized receipt matrix and y the development utility labels. The valid child uses

$$\widehat{\beta} = (Z^\top Z + \lambda I)^{-1} Z^\top y. \tag{1440}$$

The shuffled control replaces y by Py for a random permutation P. Because Z is centered,

$$\mathbb{E}_P[Z^\top P y] = 0. \tag{1441}$$

The finite shuffled child still exploits intercepts, safety filters, and accidental sample correlations, so it need not collapse to the no-op. Its lower held-out utility tests whether the original feature-outcome binding carried transferable information.

F.8 One productive edge does not imply a productive fixed point

Consider a transaction graph with states s_0, s_1, s_2. Suppose $s_0 \to s_1$ is a validated positive parent–child edge, but every authorized successor of s_1 either returns to s_1 with zero gain or enters unsafe s_2. Then $s_0 \to s_1$ witnesses one-step metaproductivity, but s_1 does not lie in a closure requiring infinitely repeated positive gains. A grandchild audit is logically necessary to distinguish a one-edge result from recurrent improvement capacity.

G Worked Meta-Headroom and Grammar-Growth Examples

G.1 Permutation-average headroom allocation

Consider five mutable modules:

$$N_5 = \{S, Q, T, C, H\}, \tag{1442}$$

representing selection, information acquisition, transport, candidate construction, and causal-atlas structure. Let

$$F_K(\varnothing) = 0.20. \tag{1443}$$

Assign singleton contributions

$$(0.08, 0.12, 0.10, 0.16, 0.14) \tag{1444}$$

and pair synergies

$$\omega_{QT} = 0.05, \qquad \omega_{CH} = 0.06, \qquad \omega_{SC} = 0.03. \tag{1445}$$

Then

$$F_K(N_5) = 0.94, \qquad \mathcal{H}(K) = 0.74. \tag{1446}$$

Every pair synergy is divided equally between its two participants by the permutation-average rule. Therefore

$$\begin{aligned} h_S &= 0.08 + \frac{0.03}{2} = 0.095, \\ h_Q &= 0.12 + \frac{0.05}{2} = 0.145, \\ h_T &= 0.10 + \frac{0.05}{2} = 0.125, \\ h_C &= 0.16 + \frac{0.06 + 0.03}{2} = 0.205, \\ h_H &= 0.14 + \frac{0.06}{2} = 0.170. \end{aligned} \tag{1447}$$

The allocation sums to

$$0.095 + 0.145 + 0.125 + 0.205 + 0.170 = 0.740, \tag{1448}$$

which verifies Theorem 44.5 in the finite companion.

G.2 A saturated endpoint and a migrated bottleneck

Suppose bridge-rank identification has success probability

$$p_{\mathrm{rank}}(K_1) = 1. \tag{1449}$$

For every child that changes only the rank-identification rule while leaving all other modules fixed,

$$p_{\mathrm{rank}}(K_2) - p_{\mathrm{rank}}(K_1) \le 0. \tag{1450}$$

This does not imply that the repair process has no remaining headroom. Let the target bridge be full rank but have minimum singular value

$$\sigma_{\min} = 10^{-4}. \tag{1451}$$

Random noise of norm 10^{-3} can then dominate the weakest identified direction. The bottleneck has migrated from rank to conditioning. A meaningful grandchild should target a criterion such as

$$\max_{S:|S|=k} \lambda_{\min}(X_S^\top X_S) \tag{1452}$$

or a regularized determinant objective, not another rank-only rule.

G.3 Finite bottleneck-migration trace

The Block 8 companion begins with

$$r_0 = (0.06, 0.18, 0.14, 0.22, 0.10) \tag{1453}$$

for selection, information, transport, construction, and atlas headroom. At each fixed-grammar step, the largest coordinate is multiplied by $1/4$. A neighboring coordinate receives a 10% verified reduction. The first four selected actions are

$$C,\ Q,\ T,\ H. \tag{1454}$$

After these reductions, the old-coordinate vector is

$$(0.06, 0.045, 0.0315, 0.0495, 0.0225). \tag{1455}$$

A conservative grammar extension then introduces a new nonlinear coordinate of magnitude

$$0.16. \tag{1456}$$

The represented headroom norm increases because a new limitation has been discovered. No old coordinate increases. Subsequent actions target the new coordinate and the remaining old bottlenecks. This separates epistemic expansion from regression.

G.4 Nested obstruction energy

Let the foreign obstruction covariance have eigenvalues

$$4, 2, 1, 0.5, 0.2, 0.1. \tag{1457}$$

Assume the initial grammar already corrects the first eigenvector. The unresolved energy is

$$2 + 1 + 0.5 + 0.2 + 0.1 = 3.8. \tag{1458}$$

Adding the remaining eigenvectors in descending order produces

$$3.8 \longrightarrow 1.8 \longrightarrow 0.8 \longrightarrow 0.3 \longrightarrow 0.1 \longrightarrow 0. \tag{1459}$$

The energy drops are

$$2, 1, 0.5, 0.2, 0.1, \tag{1460}$$

exactly the newly correctable eigenvalues. This is the distributional form of Theorem 46.2.

G.5 One-generator reduction identity

Let the old correction image be

$$I = \operatorname{span}\{e_1, e_2\} \subset \mathbb{R}^4 \tag{1461}$$

and let

$$r = (0, 0, 3, 4). \tag{1462}$$

Then

$$z = P_{I^\perp} r = (0, 0, 3, 4), \qquad \eta_I(r)^2 = 25. \tag{1463}$$

Add generator

$$v = 2e_4. \tag{1464}$$

Because $v \in I^{\perp}$,

$$\frac{|\langle z, v\rangle|^2}{\|v\|^2} = \frac{8^2}{4} = 16. \tag{1465}$$

The expanded residual is $3e_3$, so

$$\eta_{I'}(r)^2 = 9 = 25 - 16. \tag{1466}$$

A generator parallel to e_1 would have $\bar{v} = 0$ and would produce no structural reduction despite having nonzero raw norm.

G.6 Repair-conflict refinement

Index eight episodes by three bits

$$(a, b, n) \in \{0, 1\}^3, \tag{1467}$$

where (a, b) determines the valid repair class and n is nuisance. The empty grammar places every episode in one cell. There are 24 unordered pairs in that cell with different repair classes.

Adding feature a divides the episodes into two cells and reduces unresolved conflict pairs to 8. Adding b produces four cells indexed by (a, b) and eliminates every repair conflict. Adding nuisance n first would separate episodes without resolving all required repair distinctions. The greedy finite fixture therefore selects

$$a, \quad b, \tag{1468}$$

and rejects the nuisance feature.

The conservative embedding is immediate:

$$\phi_{\mathcal{G}'}(x) = (\phi_{\mathcal{G}}(x), a(x), b(x)), \tag{1469}$$

and projection onto the first coordinate recovers the old grammar.

G.7 Finite synthesis and exhaustion

Consider the typed repair programs

$$\begin{aligned} p_1(a, b) &= (0, 0), \\ p_2(a, b) &= (a, 0), \\ p_3(a, b) &= (0, b), \\ p_4(a, b) &= (b, a), \\ p_5(a, b) &= (a, b). \end{aligned} \tag{1470}$$

The complete finite verifier checks all eight episodes. Programs p_1 through p_4 each receive a counterexample and are eliminated. Program p_5 passes every episode. The loop terminates after four rejections with a verified repair.

If p_5 were removed from the grammar, the same process would return an exhaustion certificate. That certificate would justify a grammar-expansion attempt. It would not prove that no repair exists outside the declared program family.

G.8 A two-block causal atlas

Let

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{1471}$$

On $H = \mathbb{R}^4$, define four response operators

$$R_1 = X \oplus 0, \quad R_2 = Z \oplus 0, \quad R_3 = 0 \oplus X, \quad R_4 = 0 \oplus Z. \tag{1472}$$

The adjoint-closed algebra they generate is

$$M_2(\mathbb{R}) \oplus M_2(\mathbb{R}), \tag{1473}$$

with dimension 8. Its commutant consists of scalar multiples of the identity on each block and therefore has dimension 2. The minimal central projectors are

$$P_1 = I_2 \oplus 0, \qquad P_2 = 0 \oplus I_2. \tag{1474}$$

A residual e_1 generates the first two-dimensional block. A residual $e_1 + e_3$ generates both blocks and has cyclic dimension 4.

G.9 Lineage-wide budget calculation

Suppose four accepted edges use conditional false-acceptance budgets

$$0.0100, 0.0050, 0.0025, 0.00125. \tag{1475}$$

Then

$$\sum_{t=0}^{3} \delta_t = 0.01875. \tag{1476}$$

By Theorem 49.1, the probability that any accepted edge has nonpositive true metaproductivity is at most 0.01875, assuming every conditional bound is valid.

If the protected-regression budgets are

$$0.010, 0.008, 0.006, 0.004, \tag{1477}$$

then the cumulative declared budget is

$$B_R = 0.028. \tag{1478}$$

These are accounting guarantees, not evidence that the numerical budgets are ethically sufficient.

G.10 Process improvement versus compiler improvement

Let a fixed compiler Γ map a parent process and receipts to a child:

$$K_{t+1} = \Gamma(K_t, \mathcal{E}_t). \tag{1479}$$

Suppose it produces two positive edges

$$\mathcal{M}(K_1) - \mathcal{M}(K_0) > 0, \qquad \mathcal{M}(K_2) - \mathcal{M}(K_1) > 0. \tag{1480}$$

This establishes repeated output-process improvement by one compiler. It does not establish

$$\mathcal{C}(\Gamma_1) > \mathcal{C}(\Gamma_0). \tag{1481}$$

The latter requires comparing two compilers on foreign parent repair processes under matched compilation resources. The distinction prevents one successful compiler from being described as recursively improving its own compilation rule.

H Worked Repeated-Lineage and Conditioning Examples

H.1 Full rank with arbitrarily poor information

Let

$$X_\epsilon = \operatorname{diag}(1, 1, \epsilon), \qquad 0 < \epsilon \ll 1. \tag{1482}$$

Then $\operatorname{rank} X_\epsilon = 3$, but

$$\lambda_{\min}(X_\epsilon^\mathsf{T} X_\epsilon) = \epsilon^2 \tag{1483}$$

and the worst-direction least-squares variance scales as ϵ^{-2}. A binary rank gate cannot distinguish this design from an orthogonal one.

H.2 One D-optimal anchor update

Let $M = I_2$, $x = (2, 1)^\mathsf{T}$, and $w = 4$. The D-optimal information increment is

$$\log\det(M + wxx^\mathsf{T}) - \log\det M = \log(1 + wx^\mathsf{T}M^{-1}x) = \log 21. \tag{1484}$$

A low-quality copy with $w = 1/16$ gives only $\log(1 + 5/16)$, despite identical source geometry.

H.3 A-optimal trace reduction

For $M = \operatorname{diag}(1, 0.1)$ and $v = e_2$, the A-optimal reduction is

$$\frac{v^\mathsf{T}M^{-2}v}{1 + v^\mathsf{T}M^{-1}v} = \frac{100}{11}. \tag{1485}$$

The update targets the weakly informed second direction rather than increasing an already strong direction.

H.4 Why a gate-only change is insufficient

Suppose inherited rank anchors produce $\lambda_{\min}(M) = 0.5$, while the grandchild gate requires $\lambda_{\min}(M) \geq 2$. Applying the new gate to old anchors yields abstention. The same gate applied to conditioning-aware anchors with $\lambda_{\min}(M) = 14$ can commit. The gate does not create information. It only verifies information created by the query design.

H.5 Context-indexed nontransitivity

Let edge $K_0 \to K_1$ improve a bridge task distribution Ξ_0, while $K_1 \to K_2$ improves a heteroscedastic distribution Ξ_1. Positive effects in both contexts do not imply

$$\mathcal{M}_{\Xi_0}(K_2) > \mathcal{M}_{\Xi_0}(K_0) \tag{1486}$$

or any additive path gain unless Ξ_0 and Ξ_1 share a common utility or a validated context transport.

H.6 Actual Block 9 repeated edge

The frozen primary means are

$$\mathcal{M}(K_1^{\mathrm{recal}}) = 0.037393, \qquad \mathcal{M}(K_2) = 0.209580. \tag{1487}$$

Hence

$$\widehat{\Delta}_{1\to 2} = 0.172188. \tag{1488}$$

The interval $[0.141481, 0.203110]$ is strictly positive. Both arms are full rank, so rank recovery cannot explain the edge.

H.7 Quality-permutation control

The quality-permuted policy retains the same selection code but breaks the association between anchor identity and supplied reliability. Its blind gain remains positive, but its information floor collapses and the frozen witness commits no primary targets. This isolates semantic quality binding from a favorable policy label.

I Worked Structural Grammar-Expansion Examples

I.1 A scalar two-chart relation has a persistent global residual

Let $h \sim \mathcal{N}(0,1)$ and let $s \in \{-1,+1\}$ be independent and balanced. Define

$$y = (a + sb)h, \qquad b \neq 0. \tag{1489}$$

A global scalar bridge predicts $\widehat{y} = th$. The population minimizer is $t^* = a$, and

$$\inf_t \mathbb{E}(y - th)^2 = \mathbb{E}(sbh)^2 = b^2. \tag{1490}$$

The residual is not an estimation artifact. It remains after infinite data because one slope cannot equal both $a - b$ and $a + b$. The two-chart grammar uses

$$t_- = a - b, \qquad t_+ = a + b, \tag{1491}$$

and has zero noiseless approximation error. This is the one-dimensional form of Theorem 58.4.

I.2 The structural extension preserves the inherited process exactly

Let the old bridge be $T \in \mathbb{R}^{q\times d}$ and let g be any admissible separator. The embedding

$$T \longmapsto (g, T, T) \tag{1492}$$

returns Th on both charts. Hence the local grammar can execute the inherited process without approximation. A conservative runtime may therefore use

$$K_{\mathrm{chart}}(h) = \begin{cases} K_{\mathrm{local}}(h), & \text{if the chart certificate passes,} \\ K_{\mathrm{global}}(h), & \text{otherwise.} \end{cases} \tag{1493}$$

The child is structurally richer, but it does not require deletion of the parent behavior. This is the operational meaning of Theorem 58.3.

I.3 Exact distance from a chart repair to the global subgrammar

Equip

$$\mathrm{End}(H_t) \oplus \mathrm{End}(H_t) \tag{1494}$$

with the product Frobenius norm. For a chart repair pair (D_-, D_+), write

$$D_{\mathrm{avg}} = \frac{D_- + D_+}{2}, \qquad D_{\mathrm{ctr}} = \frac{D_+ - D_-}{2}. \tag{1495}$$

The closest diagonal repair is $(D_{\mathrm{avg}}, D_{\mathrm{avg}})$. Therefore

$$\begin{aligned} \mathrm{dist}^2\left((D_-, D_+), \mathcal{D}_{\mathrm{diag}}\right) &= \|D_- - D_{\mathrm{avg}}\|_F^2 + \|D_+ - D_{\mathrm{avg}}\|_F^2 && (1496)\\ &= 2\,\|D_{\mathrm{ctr}}\|_F^2 && (1497)\\ &= \frac{1}{2}\,\|D_+ - D_-\|_F^2\,. && (1498) \end{aligned}$$

Thus $D_+ \neq D_-$ gives a quantitative structural-novelty certificate, not merely two differently named copies of one global operator.

I.4 A nuisance separator does not certify grammar growth

Suppose the true target relation depends on $s = \mathbf{1}\{h_1 > 0\}$, but a candidate separator uses

$$g(h) = \mathbf{1}\{h_2 > 0\}, \tag{1499}$$

where h_2 is independent of s and the repair-relevant state. With balanced cells and unlimited data, both g-branches see the same mixture of the two true maps. Each fitted local map converges to the global population minimizer. Consequently,

$$\varepsilon_{\mathrm{loc}} = \varepsilon_{\mathrm{glob}}, \qquad \Delta_{\mathrm{gram}} = 0. \tag{1500}$$

A larger syntax is therefore not enough. Structural credit requires a held-out residual reduction, conservative replay, a nonzero repair contrast, and foreign protected gain.

I.5 Actual Block 10 structural edge

The strengthened global parent uses 80 bridge-fit and 16 witness queries. The chart child uses 48 bridge-fit, 32 chart-selection, and 16 witness queries. Both therefore consume 96 paired queries. Their primary mean selected gains are

$$\mathcal{M}(K_{2,G}^{96}) = 0.000000, \qquad \mathcal{M}(K_{3,C}^{96}) = 0.334224. \tag{1501}$$

Hence

$$\widehat{\Delta}_{G\to C} = 0.334224, \qquad \mathrm{CI}_{95\%} = [0.266459, 0.401903]. \tag{1502}$$

The direct global-affine oracle has mean gain 0.176694, and the child exceeds it by

$$0.157530, \qquad \mathrm{CI}_{95\%} = [0.080863, 0.234135]. \tag{1503}$$

Validation alignment NRMSE changes from

$$\varepsilon_{\mathrm{glob}} = 0.598513 \tag{1504}$$

to

$$\varepsilon_{\mathrm{loc}} = 0.064975. \tag{1505}$$

Mean gate accuracy is 0.874854, no harmful primary target is selected, and gate-only, random-gate, and shuffled-binding controls are strongly negative. These quantities support a finite conservative expansion from the global-affine grammar to the invariant-context two-chart grammar. They do not establish autonomous protected-packet or separator-language invention.

J Worked Endogenous Atlas-Growth Examples

J.1 Lagrange recovery of three response blocks

Let

$$R_1 = P_1 + 2P_2 + 4P_3, \qquad R_2 = 3P_1 - P_2 + 2P_3, \tag{1506}$$

where P_1, P_2, P_3 are pairwise orthogonal rank-two projectors. Take

$$B = R_1 + 2R_2 = 7P_1 + 0P_2 + 8P_3. \tag{1507}$$

The three eigenvalues $7, 0, 8$ are distinct. Therefore

$$P_1 = \frac{B(B - 8I)}{7(7 - 8)}, \tag{1508}$$

$$P_2 = \frac{(B - 7I)(B - 8I)}{(0 - 7)(0 - 8)}, \tag{1509}$$

$$P_3 = \frac{B(B - 7I)}{8(8 - 7)}. \tag{1510}$$

Every projector is a polynomial in the observed response combination. No hidden coordinate label is required. Any orthonormal basis inside a rank-two block is equally valid, illustrating the

within-block gauge of Theorem 62.3.

J.2 Commutant dimension for the Block 11 fixture

Suppose

$$H = H_1 \oplus H_2 \oplus H_3, \qquad \dim H_a = 2. \tag{1511}$$

Under separating block-scalar responses, every commuting operator has form

$$X = X_1 \oplus X_2 \oplus X_3, \qquad X_a \in M_2(\mathbb{R}). \tag{1512}$$

Hence

$$\dim \mathfrak{A}_R' = 3 \dim M_2(\mathbb{R}) = 12. \tag{1513}$$

The companion's modal commutant dimension is 12, matching the exact finite theory.

J.3 Degenerate signatures merge two functional packets

Let

$$R_1 = P_1 + P_2 + 3P_3, \qquad R_2 = 2P_1 + 2P_2 - P_3. \tag{1514}$$

Blocks 1 and 2 have the same signature $(1, 2)$. Every polynomial in R_1, R_2 acts with the same scalar on H_1 and H_2. The finest response-algebra projector is therefore

$$P_{12} = P_1 + P_2, \tag{1515}$$

not the pair P_1, P_2. A compiler that outputs two separate blocks from these interventions has invented information absent from its evidence. The Block 11 stress witness instead abstains.

J.4 A spectral-gap stability calculation

Suppose a selected response combination has eigenvalue clusters separated by

$$\gamma = 0.04 \tag{1516}$$

and estimation error

$$\|E\|_2 = 0.003. \tag{1517}$$

The conditional projector bound gives

$$\left\|\widehat{P} - P\right\|_2 \le \frac{2(0.003)}{0.04} = 0.15. \tag{1518}$$

If the same fit error occurred with $\gamma = 0.004$, the bound would be 1.5 and therefore vacuous. A low response-fit residual alone does not certify an identifiable atlas. The spectral gap is an independent required field.

J.5 Why a fixed packet can fail under hidden rotation

Let the true context packet be

$$P_* = UU^{\mathsf{T}} \tag{1519}$$

for a target-specific orthonormal matrix $U \in \mathbb{R}^{6\times 2}$, while the inherited packet is

$$P_0 = \mathrm{diag}(1, 1, 0, 0, 0, 0). \tag{1520}$$

Let the chart label be

$$c(h) = \mathbf{1}\{e_1^{\mathsf{T}} U^{\mathsf{T}} h > 0\}. \tag{1521}$$

For generic U, $c(h)$ is not measurable from $P_0 h$. No threshold or nonlinear classifier using only $P_0 h$ can recover the chart perfectly. The response atlas can recover $\mathrm{im}\, U$ up to basis gauge when the intervention signatures separate it.

J.6 Actual Block 11 edge

Both processes use 120 effect-query units. The inherited parent uses

$$80 + 16 + 24 = 120 \tag{1522}$$

ordinary bridge, chart-validation, and witness queries. The endogenous child uses

$$56 + 24 + 16 + 24 = 120 \tag{1523}$$

bridge, response, chart-validation, and witness queries.

On 80 fresh foreign targets,

$$\mathcal{M}(K_3) = 0, \qquad \mathcal{M}(K_4) = 0.207325, \tag{1524}$$

so

$$\widehat{\Delta}_{3\to 4} = 0.207325, \qquad \mathrm{CI}_{95\%} = [0.131902, 0.287910]. \tag{1525}$$

The selected recovered packet has mean hidden context overlap 0.821461, compared with 0.322091 for the inherited coordinate-fixed packet. Blind endogenous application has mean gain 0.271529 but harm rate 0.1875. The witness lowers selected gain to 0.207325 and reduces selected harm to zero. This is a finite endogenous-atlas result, not an unrestricted discovery theorem.

K Worked Response-Language and Constructor-Growth Examples

K.1 An exact response alias

Let three rank-one blocks have inherited signatures $(1, 0)$, $(1, 0)$, and $(0, 1)$. Every polynomial in the two inherited block-scalar responses acts equally on the first two blocks. Adding a third response with scalars $(1, -1, 0)$ separates them, and Lagrange interpolation recovers all three central projectors.

K.2 A weak separator is statistically inadequate

If the new scalars are $(1+\epsilon, 1-\epsilon, 0)$, the algebra is formally separating for every $\epsilon \neq 0$, but empirical projector recovery requires response-estimation error small relative to the 2ϵ spectral gap. This illustrates why algebraic identifiability and practical utility are distinct gates.

K.3 Nonzero Hessian defeats every affine composition

For $z \sim N(0, I_2)$ and $q(z) = z_1^2 - z_2^2$, every affine predictor has residual variance at least 4. The degree-two feature vector $(z_1^2, z_1 z_2, z_2^2)$ represents q exactly. No increase in affine depth changes this conclusion because affine maps remain closed under composition.

K.4 Why the full quadratic control matters

If both a packet-local and full-space degree-two constructor fit the same target, the experiment identifies missing polynomial degree but does not establish that localization is uniquely necessary. Packet localization may still improve efficiency, conditioning, or transfer in harder settings, but those are separate claims.

L Worked Research-Compiler Examples

L.1 Evaluator-nullspace shrink

Let $T = \mathbb{R}^2$, $Db_1(x_1, x_2) = x_1$, and $Db_2(x_1, x_2) = x_2$. With only b_1, the invisible subspace is $\mathrm{span}(e_2)$. Adding b_2 reduces the intersection of kernels to $\{0\}$. The new benchmark contributes an independent behavioral distinction rather than merely rescaling the old score.

L.2 Uniform forecast-regret bound

Suppose true candidate gains are $(0.70, 0.62, 0.10)$ and predictions are $(0.64, 0.66, 0.12)$. The forecaster selects candidate two. Here $\varepsilon = 0.06$ and the realized regret is $0.08 \leq 2\varepsilon = 0.12$, illustrating the fixed-candidate theorem.

L.3 Results-free proposal receipt

A candidate receipt predicts improvement on four scored dimensions, declares a capability-regression ceiling, hashes the proposed code, and names a falsification condition before execution. After evaluation, the post-receipt appends the observed score vector, hidden audit, and monitor verdict without changing the pre-result fields.

L.4 Actual Block 13 compiler edge

On the primary audit, the parent and child gains are 0.883004 and 0.950231, so

$$\widehat{\Delta}_{\Gamma}^{\mathrm{primary}} = 0.067227. \tag{1526}$$

The child changes research allocation rather than the candidate grammar. The architecture-foreign paired effect is 0.090800.

L.5 Archive and monitor controls

Shuffling archive labels lowers primary gain to 0.282550, showing that the compiler depends on correctly bound prior evidence. Removing the integrity monitor lowers validity-filtered gain to

0.797956 because invalid methods can occupy the selected slot. Neither control is a claim about universal archive or monitor sufficiency.

M Worked Robust Research-Compiler Examples

M.1 Aliased static diagnostics

Let $Z \in \{0,1\}$ be equally likely and let $X_0 \sim \mathcal{N}(0,1)$ under both classes. No static rule exceeds accuracy $1/2$. If experiment a returns $Y_a = Z + \xi$ with $|\xi| < 1/2$, then thresholding Y_a at $1/2$ recovers Z exactly. This is the simplest form of the Block 14 distinction between cheap diagnostic aliasing and active response separation.

M.2 Two-obstruction mixture

Let

$$R = \begin{bmatrix} 1 & 0 & 1/2 \\ 0 & 1 & 1/2 \\ 1 & -1 & 0 \end{bmatrix}, \qquad z = \begin{bmatrix} 0.6 \\ 0.4 \\ 0 \end{bmatrix}. \tag{1527}$$

Then $y = Rz = (0.6, 0.4, 0.2)^\top$. A forced one-class diagnosis discards one active component, while $R^\dagger y$ recovers the mixture exactly. If the third response row is removed, the first two classes remain identifiable here, but other mixtures can become aliased; the smallest singular value quantifies the resulting instability.

M.3 Open-set residual

Let inherited known responses span the nonnegative cone generated by e_1 and e_2 in $\mathbb{R}^3$. The unknown response $y = e_3$ has

$$\operatorname{dist}(y, \operatorname{cone}\{e_1, e_2\}) = 1. \tag{1528}$$

Any threshold below one and above known-response noise flags the case as outside the inherited language. The flag does not identify the missing obstruction. It only prevents a false confident inherited label.

M.4 Conjunctive integrity monitoring

Suppose a specification monitor misses an invalid method with probability 0.2 and a trajectory monitor misses it with probability 0.15. The conjunction has false-negative probability at most 0.15. Under conditional independence the rate is 0.03. Without independence, the product must not be claimed.

M.5 Zero observed invalid selections

If zero invalid selections are observed in $n = 144$ independent audit problems, the exact one-sided 95% upper bound is

$$1 - 0.05^{1/144} \approx 0.02059. \tag{1529}$$

The correct statement is therefore "zero observed, upper bound approximately 2.06%," not "the compiler has zero invalid-selection probability."

N Worked Compiler-Language Growth Examples

N.1 One unknown token with incompatible methods

Let two problems a and b both map to $\bot$. Suppose

$$\mathcal{A}^\star(a) = \{m_1\}, \qquad \mathcal{A}^\star(b) = \{m_2\}, \qquad m_1 \neq m_2. \tag{1530}$$

Any deterministic policy conditioned only on $\bot$ must choose one method and fail the other problem. This is a diagnostic-language obstruction, not ordinary estimation noise.

N.2 A conservative binary refinement

Let probe p return -1 on a and $+1$ on b. With threshold zero, define tokens $(\bot, -)$ and $(\bot, +)$. The projection that forgets the sign recovers $\bot$, while the refined binding sends the two tokens to m_1 and m_2. The old language is preserved and the conflict is removed.

N.3 Why taxonomy-only growth is insufficient

Suppose the inherited deployable grammar contains only singleton methods u and v, each with gain 0.3 on both novel cases, while pair programs $u \circ v$ and $v \circ u$ obtain gains 0.9 on opposite cases. Splitting the unknown token without adding the pair constructor leaves the attainable gain at 0.3.

N.4 Why grammar-only growth is insufficient

If the two pair methods are added but the unknown token remains unrefined, one pooled binding must be used. It can solve at most one of the two cases. Correct language growth requires the diagnostic refinement and constructor outputs to remain bound.

N.5 Actual Block 15 structural margin

The development inherited-method oracle is 0.377257 and the selected language obtains 0.948962. Thus

$$\widehat{\kappa} = 0.948962 - 0.377257 = 0.571704. \tag{1531}$$

On the primary foreign audit, the matched-budget child-parent effect is 0.225762; under generator shift it is 0.240296.

N.6 Finite pair grammar versus unrestricted synthesis

The executable compiler searches 30 ordered pairs constructed from six inherited typed method families. Selecting a useful pair from this grammar is structural relative to the deployable singleton parent, but it does not establish invention of loops, conditionals, recursion, arbitrary code, or a new primitive research operation.

O Receipt Schema

A minimal machine-readable receipt should contain fields equivalent to the following symbolic schema:

```
repair_receipt:
  lineage:
    parent_ids
    candidate_id
    active_head_before_and_after
    lineage_edge_proofs
    rollback_checkpoint
  transaction:
    editable_mask
    witness_mask_and_versions
    prepare_seal_execute_reveal_decide_finalize_timestamps
    proposal_and_prediction_commitments
    trace_schema_and_factorization_tests
    acceptance_program_digest
    capability_and_resource_budget
    buffered_and_irreversible_effects
    rollback_fidelity
  diagnosis:
    failure_signature
    causal_hypotheses
    probes
    posterior_uncertainty
  correctability:
    correction_domain
    protected_invariant_map
    defect_space
    linear_response_map
    singular_values
    correctable_projection
    obstruction_projection
    obstruction_norm
    dual_witness
    repair_budget
    empirical_error_bounds
  geometry:
    strata
    packet_blocks
    leakage_profile
    local_defects
    local_obstructions
    overlap_cocycle
```

```
    cycle_projection
    gluing_obstruction_norm
    nonlinear_holonomy
    formal_obstruction_order
    gluing_status
  repair:
    operator_id
    correction_program
    applicability_conditions
    regularization_cost
  transport:
    source_and_target_correction_data
    correction_and_defect_maps
    route_and_edge_receipts
    square_and_identification_defects
    authorization_and_invariant_commutation
    resource_amplification
    obstruction_faithfulness
    cumulative_residual_bound
    repair_and_semantic_holonomy
    deterministic_and_probabilistic_gain_bounds
  evaluator_upgrade:
    old_and_new_evaluator_ids
    raw_evidence_embedding
    internal_transport_and_core_retraction
    protected_history_replay_set
    holonomy_energy_spectrum
    exact_or_approximate_core
    core_adequacy_modulus
    heldout_paired_risk_bound
    branch_cycle_checks
  closure:
    universal_controlled_or_productive_semantics
    static_invariant_results
    successor_or_action_model_digest
    fixed_point_iteration_and_membership
    preservation_policy
    elimination_or_counterexample_certificate
    progress_threshold_and_ranking_certificate
  archive:
    descriptor_vector_and_uncertainty
    dominance_and_novelty_comparisons
    pareto_cell_or_continuation_signature
    retained_repair_schemas
    retained_obstruction_certificates
```

```
    evidence_namespace_and_use_history
  metaproductivity:
    foreign_agent_population_id
    horizon_candidate_and_compute_budget
    paired_baseline_and_descendant_gains
    confidence_bound
    distribution_shift_bound
    correctable_and_obstruction_energy
    causal_ablation_result
  grammar_extension:
    old_and_new_grammar_ids
    conservative_embedding
    old_proof_replay
    semantic_debt
    resolved_old_obstruction_classes
    newly_exposed_obstruction_directions
    representation_separation_certificate
  autocatalysis:
    schema_equivalence_class
    causal_parent_ablations
    offspring_type_counts
    entrywise_confidence_matrices
    strongly_connected_component
    reproduction_number_and_budget
    saturation_and_utility_checks
  empirical_realization:
    abstract_model_id_and_version
    runtime_implementation_id
    abstraction_map_and_refinement_checks
    protected_observation_commutation
    frozen_endpoint_and_claim_gate_set
    data_partitions_and_seed_schedule
    preoutcome_commitment_digest
    package_manifest_and_environment
    exact_and_statistical_replay_status
    oracle_leakage_checks
    strongest_admissible_claim_level
  learned_validation:
    task_and_defect_grammar_ids
    source_and_target_encoder_ids
    train_anchor_calibration_audit_split_digests
    causal_probe_and_counterfactual_definition
    packet_rank_basis_and_singular_spectrum
    equal_rank_pca_random_and_wrong_block_controls
    source_repair_parameters_and_digest
```

```
    paired_anchor_correspondence_digest
    learned_bridge_and_regularization
    alignment_and_repair_equivariance_residuals
    blind_selected_and_target_oracle_scores
    shuffled_anchor_and_direct_copy_controls
    witness_thresholds_and_commit_reason
    target_family_and_foreignness_certificate
    strongest_block06_claim_gate
  sealed_meta_improvement:
    parent_policy_id_and_digest
    candidate_set_and_matched_budget
    candidate_feature_schema_digest
    development_and_meta_validation_split_digests
    excluded_final_audit_split_ids
    compiler_source_digest
    meta_grammar_and_grammar_size
    selected_configuration_and_validation_score
    utility_and_risk_head_parameters
    candidate_specific_residual_scales
    seal_before_audit_load_trace
    descendant_receipt_digest
    paired_seed_task_and_architecture_foreign_effects
    bootstrap_intervals_and_sign_flip_tests
    average_tail_and_maximum_risk_gates
    raw_no_risk_shuffled_ablation_and_oracle_controls
    selector_level_vs_structural_claim_boundary
    strongest_block07_claim_gate
  meta_headroom:
    repair_process_module_factorization
    restricted_oracle_class_ids
    lower_and_upper_envelope_bounds
    ordered_and_permutation_average_allocations
    module_interaction_terms
    saturation_tolerances_and_certificates
    headroom_adjusted_gain
  bottleneck_migration:
    unresolved_module_headroom_vector
    selected_meta_action_and_response_model
    weighted_lyapunov_certificate
    collateral_headroom_and_error_terms
    old_coordinate_repair_progress
    newly_represented_discovery_energy
    dominant_bottleneck_before_and_after
  persistent_obstruction:
    grammar_indexed_correction_images
```

```
    newly_correctable_energy
    induced_cokernel_transport
    repair_death_or_forgetting_death
    protected_obstruction_faithfulness_margin
    cumulative_dynamic_semantic_debt
  generator_discovery:
    projected_foreign_obstruction_samples
    obstruction_covariance_and_spectrum
    candidate_response_direction
    one_generator_or_rank_k_energy_reduction
    cost_and_risk_adjusted_acquisition_score
    exact_generator_ablation
    structural_novelty_certificate
  conflict_refinement:
    aliased_episode_cell
    validated_authorized_repair_sets
    minimal_repair_conflicts
    candidate_feature_or_probe_family
    selected_separator_and_cost
    old_grammar_projection
    remaining_conflict_hypergraph
  causal_atlas:
    intervention_response_operator_ids
    generated_adjoint_closed_algebra
    commutant_and_center_dimensions
    minimal_central_projectors
    multiplicity_gauge
    defect_cyclic_module
    commutator_spectral_gap
    perturbation_and_stability_bound
  sequential_lineage:
    pre_audit_filtration_digest
    edge_true_effect_estimand
    conditional_lower_and_upper_bounds
    edge_false_acceptance_budget
    cumulative_archive_error_budget
    protected_regression_budget
    semantic_debt_budget
    prior_receipt_replay
    repeated_lineage_claim_level
  repeated_lineage_conditioning:
    block07_ancestor_policy_and_semantic_digests
    ancestor_receipt_replay_status
    fresh_development_and_audit_context_ids
    rank_saturation_certificate
```

```
    query_quality_metadata_digest
    weighted_information_matrix_diagnostics
    recalibrated_child_policy_and_gate
    conditioning_grandchild_policy_and_gate
    finite_design_grammar_and_compiler_ledger
    matched_deployment_query_ledgers
    preaudit_grandchild_seal
    paired_grandchild_child_effect_and_interval
    gate_only_quality_permutation_and_random_budget_controls
    target_side_conditioning_and_stress_results
    context_indexed_two_edge_claim_boundary
  structural_grammar_expansion:
    block09_ancestor_policy_and_semantic_digests
    global_parent_and_chart_child_configurations
    finite_separator_feature_grammar
    development_grammar_inadequacy_certificate
    global_and_local_validation_residuals
    chart_support_and_gate_accuracy
    target_repair_contrast
    conservative_parent_embedding_and_fallback
    matched_48_query_ledgers
    preaudit_structural_child_seal
    paired_child_parent_effect_and_interval
    gate_only_shuffled_binding_oracle_and_stress_controls
    structural_claim_boundary
  endogenous_atlas_growth:
    block10_ancestor_policy_and_seal_digests
    supplied_separator_removed
    intervention_response_operator_ids_and_query_ledger
    response_algebra_rank_and_commutant_dimension
    recovered_central_projectors_and_block_ranks
    response_fit_commutator_and_spectral_gap_certificate
    selected_repair_relevant_block_and_separator
    source_repair_invariance_and_target_gate_agreement
    hidden_overlap_metrics_reserved_for_audit
    conservative_block10_embedding_and_fallback
    matched_parent_and_child_effect_query_budgets
    preaudit_atlas_certificate_compiler_and_policy_digests
    paired_child_parent_effect_interval_and_sign_flip_test
    pca_full_random_shuffled_blind_oracle_and_stress_controls
    endogenous_atlas_claim_boundary
  response_language_growth:
    inherited_response_operator_ids_and_alias_partition
    finite_candidate_intervention_language
    candidate_signature_gaps_commutators_fit_and_costs
```

```
    selected_separating_operator_and_central_refinement
    conservative_old_algebra_embedding
    matched_parent_and_child_query_ledgers
    preaudit_response_certificate_and_policy_digests
    paired_effect_interval_controls_and_stress_boundary
  constructor_growth:
    inherited_constructor_operad_and_function_class
    hessian_or_equivalent_expressivity_obstruction
    finite_candidate_constructor_language
    selected_degree_packet_and_feature_basis
    conservative_affine_embedding
    separately_compiled_parent_and_child_witnesses
    same_target_affine_oracle_and_nonlinear_controls
    paired_effect_interval_stress_and_claim_boundary
  research_compiler:
    ancestor_result_and_archive_digests
    literature_prior_and_finding_graph
    candidate_method_grammar_and_canonical_order
    pre_result_hypothesis_objective_data_and_falsification_fields
    proposal_specification_and_executable_code_hashes
    obstruction_classifier_and_outcome_risk_forecaster_digests
    cheap_diagnostic_and_full_experiment_allocation_ledger
    evaluator_portfolio_capability_gates_and_integrity_monitor
    development_meta_validation_primary_and_architecture_split_commitments
    sealed_parent_child_compiler_objects
    paired_validity_filtered_audit_effects_controls_and_claim_boundary
  robust_research_compiler:
    block13_ancestor_result_digest
    fixed_candidate_grammar_and_six_experiment_budget
    contaminated_archive_digest_and_estimator_type
    sequential_response_experiment_ledger
    sparse_obstruction_mixture_and_open_set_residual
    evaluator_portfolio_and_aggregation_rule
    specification_and_trajectory_monitor_verdicts
    primary_and_generator_foreign_split_commitments
    compiler_configuration_and_preaudit_seal
    groupwise_mean_tail_negative_unknown_and_invalid_metrics
    paired_effect_intervals_ablation_ledger_and_claim_boundary
  compiler_language_growth:
    block14_ancestor_result_digest
    inherited_diagnostic_and_method_language_digests
    open_set_repair_conflict_hypergraph
    generic_probe_language_and_selected_probe
    anonymous_refined_tokens_and_threshold
    generic_method_constructor_language
```

```
    selected_token_to_method_bindings
    inherited_oracle_expanded_oracle_and_structural_margin
    exact_conservative_fallback_receipt
    matched_research_unit_ledger
    primary_and_generator_foreign_split_commitments
    compiler_language_certificate_and_preaudit_seal
    taxonomy_grammar_binding_controls_and_claim_boundary
  decision:
    commit_rollback_or_quarantine
    rejection_reason
  abstraction:
    repair_schema_class
    foreign_agent_transfer_results
    newly_correctable_directions
    surviving_obstruction_classes
```

P Dependency Map

The principal results form the following dependency chain. The table is intentionally breakable across pages so that the complete map remains legible.

Prerequisite structure		Derived result
Protected evaluation context	$\Longrightarrow$	Nonvacuous improvement relation
Relative introspection	$\Longrightarrow$	Bounded decision regret
Authorized response map	$\Longrightarrow$	Correctable–obstructed decomposition
Local repair and overlap coherence	$\Longrightarrow$	Repair descent
Transport squares and holonomy bounds	$\Longrightarrow$	Gain-preserving transfer
Evaluator transports and adequacy	$\Longrightarrow$	Conservative evaluator evolution
Frozen witness and trace factorization	$\Longrightarrow$	Transactional noncircularity
Immutable checkpoint and buffered effects	$\Longrightarrow$	Atomic rollback
Finite transition game	$\Longrightarrow$	Universal and controlled closure
Safe attractor and validated gain set	$\Longrightarrow$	Productive Büchi closure
Append-only archive and multiobjective descriptors	$\Longrightarrow$	Monotone envelopes and finite covers
Matched foreign agents and paired evidence	$\Longrightarrow$	Metaproductivity certificate
Nested repair images and defect distribution	$\Longrightarrow$	Task-relevant structural gain
Conservative grammar chain	$\Longrightarrow$	Proof persistence and directed limit
Ontology embedding	$\Longrightarrow$	Resolved old/new obstruction exact sequence
Causal offspring receipts	$\Longrightarrow$	Repair-discovery matrix
Runtime refinement and observation commutation	$\Longrightarrow$	Trace-sound implementation
Finite exact realization and exhaustive endpoints	$\Longrightarrow$	Decidable fixture claims
Sealed package and cumulative claim gates	$\Longrightarrow$	Bounded empirical interpretation
Causal intervention differences and rank condition	$\Longrightarrow$	Hidden repair-packet identification

Continued on the next page

Prerequisite structure		Derived result
Spanning paired anchors and linear hidden relation	$\Longrightarrow$	Identified repair bridge
Injective bridge and frozen source repair	$\Longrightarrow$	Exact transported repair
Compression or nonlinear residual and frozen witness	$\Longrightarrow$	Commit or abstain
Development receipts and separate meta-validation	$\Longrightarrow$	Sealed child selector
Identical candidate bundles and paired foreign audit	$\Longrightarrow$	Selector-level metaproductivity certificate
Risk upper score and protected-regression gates	$\Longrightarrow$	Constrained child commitment
Fixed candidate grammar	$\Longrightarrow$	No structural repair-image claim
Fresh grandchild edge and newly sealed evidence	$\Longrightarrow$	First repeated-lineage test
Heteroscedastic paired-anchor model and valid quality weights	$\Longrightarrow$	Weighted bridge covariance and information geometry
Full source rank with low $\lambda_{\min}(M_S)$	$\Longrightarrow$	Conditioning bottleneck beyond identifiability
Quality-aware D-optimal design and matched query budget	$\Longrightarrow$	Conditioning-aware grandchild candidate
Ancestor replay, fresh supports, pre-audit seal, and paired audit	$\Longrightarrow$	Two-edge context-indexed repeated-lineage certificate
Restricted oracle envelopes and module factorization	$\Longrightarrow$	Meta-headroom and saturation
Headroom state and authorized contraction action	$\Longrightarrow$	Bottleneck migration
Nested correction images and foreign residuals	$\Longrightarrow$	Persistent obstruction energy
Projected obstruction and candidate response direction	$\Longrightarrow$	Generator-acquisition score
Repair-conflict hypergraph and separating features	$\Longrightarrow$	Conservative grammar refinement
Intervention-response operators and adjoint closure	$\Longrightarrow$	Central causal atlas
Conditional edge bounds and summable budgets	$\Longrightarrow$	Lineage-wide validity
Persistent global residual and finite chart separator family	$\Longrightarrow$	Grammar-inadequacy certificate
Conservative global-to-chart embedding and nonzero repair contrast	$\Longrightarrow$	New chart-specific correction direction
Matched foreign audit with binding and gate controls	$\Longrightarrow$	Finite structural metaproductivity certificate
Separating block-scalar intervention responses	$\Longrightarrow$	Exact central response atlas up to permutation and within-block gauge
Response estimation error and spectral gap	$\Longrightarrow$	Approximate central-projector stability
Recovered response blocks and repair-conflict validation	$\Longrightarrow$	Repair-relevant atlas labeling
Supplied-packet removal, conservative fallback, and matched foreign audit	$\Longrightarrow$	Finite endogenous-atlas metaproductivity certificate
Inherited block-scalar response alias	$\Longrightarrow$	Polynomial nonseparation certificate

Continued on the next page

Prerequisite structure		Derived result
Separating candidate intervention and spectral margin	$\Longrightarrow$	Conservative central response refinement
Matched response audit and correspondence controls	$\Longrightarrow$	Finite response-language metaproductivity certificate
Affine repair-operad closure and nonzero Hessian	$\Longrightarrow$	Constructor function-class obstruction
Degree-two conservative embedding and matched audit	$\Longrightarrow$	Finite candidate-constructor metaproductivity certificate
Results-free proposal fields and immutable code digest	$\Longrightarrow$	Non-retroactive proof-carrying research receipt
Independent evaluator response and benchmark portfolio refinement	$\Longrightarrow$	Strict shrinkage of one metric-gaming nullspace
Bounded outcome-prediction error on a fixed candidate set	$\Longrightarrow$	Uniform 2ε selection-regret bound
Correctly bound research archive and obstruction classifier	$\Longrightarrow$	Bottleneck-sensitive experiment allocation
Matched method grammar, evaluator, monitor, budget, seal, and paired audits	$\Longrightarrow$	Finite research-compiler metaproductivity certificate
Static diagnostic alias and separating response experiments	$\Longrightarrow$	Necessity of sequential diagnosis
Sparse inherited response cone and residual margin	$\Longrightarrow$	Finite open-set certificate
Contaminated archive and robust summary statistics	$\Longrightarrow$	Finite contamination-stability comparison
Evaluator portfolio and independent behavior dimension	$\Longrightarrow$	Strict reduction of one evaluator nullspace
Conjunctive integrity witnesses and adaptive attack methods	$\Longrightarrow$	Validity-filtered monitor-ensemble comparison
Matched fixed grammar, stress atlas, seal, group and tail gates	$\Longrightarrow$	Finite adversarial compiler-robustness certificate
Inherited unknown-token conflict and disjoint acceptable-method sets	$\Longrightarrow$	Diagnostic-language inadequacy certificate
Generic separating probe and conservative token projection	$\Longrightarrow$	Finite anonymous diagnostic refinement
Generic pair-method constructor language and positive oracle margin	$\Longrightarrow$	Task-relevant compiler method-language expansion
Matched eight-unit audits, fallback, seal, and language controls	$\Longrightarrow$	Finite compiler-language metaproductivity certificate
Nonnegative spectral bounds	$\Longrightarrow$	Local autocatalytic certificate
Fixed finite grammar or bounded utility	$\Longrightarrow$	Saturation of indefinite fixed-size gain

References

[1] Komorebi P., *Self-Indexed Workspace Closure: Expanded Theory Draft*, version 0.4, unpublished research manuscript, 2026.

[2] Komorebi P., *Projector-Resolved Seed Calculus III*, unpublished research manuscript and companion implementation, 2026.

[3] Komorebi P., *RRCT Awareness Research Handoff*, version 0.16, internal research handoff, 2026.

[4] Komorebi P., *Operator-Symbolic Language Core and Proof-Carrying Transaction Notes*, unpublished research artifacts, 2026.

[5] Komorebi P., *Komorebi Math Master Collected Edition*, unpublished collected research edition, 2026.

[6] Komorebi P., *SRIC Block 6 Learned-Agent Validation Reference Package*, version 0.2.0, reproducible software and frozen result archive, 2026.

[7] Komorebi P. *SRIC Block 7 Sealed Parent–Child Metaproductivity Reference Implementation*, 2026. Executable selection-only and bridge-policy descendants, pre-audit seals, matched-resource foreign benchmarks, causal controls, frozen results, tests, and manifests.

[8] Komorebi P., *SRIC Block 8 Meta-Headroom and Obstruction-Growth Symbolic Reference Implementation*, 2026. Exact finite fixtures for restricted oracle envelopes, bottleneck migration, obstruction towers, conflict refinement, repair synthesis, causal-atlas decomposition, sequential budgets, tests, and manifests.

[9] Komorebi P., *SRIC Block 9 Repeated Lineage and Conditioning-Aware Query Design Reference Implementation*, 2026. Executable ancestor replay, heteroscedastic bridge environment, rank-saturation diagnosis, finite conditioning-aware grandchild compiler, fresh paired foreign audits, causal controls, frozen results, tests, and manifest.

[10] Komorebi P., *SRIC Block 10 Structural Grammar Expansion Reference Implementation*, 2026. Executable global-grammar obstruction fixture, conservative two-chart compiler, matched foreign audits, chart-binding and gate controls, stress families, frozen results, tests, and manifest.

[11] Komorebi P., *SRIC v1.2 Block 11 Endogenous Atlas-Growth Reference Implementation*, package-closed executable companion, 2026.

[12] Komorebi P., *SRIC v1.3 Block 12 Response-Language and Candidate-Constructor Growth Reference Implementation*, package-closed executable companion with matched-resource response-language and degree-two constructor audits, 2026.

[13] Yueh-Han Chen, Jiaxin Wen, and Jan Hendrik Kirchner, *Automated Researchers Can Reliably Mitigate Alignment Failures*, Anthropic Alignment Science, 2026, https://alignment.anthropic.com/2026/automated-alignment-researchers/.

[14] Komorebi P., *SRIC v1.4 Block 13 Proof-Carrying Automated Research Compiler Reference Implementation*, package-closed executable companion with matched-budget compiler audits, results-free receipts, integrity controls, and frozen manifests, 2026.

[15] Komorebi P., *SRIC v1.5 Block 14 Adversarial Open-Set Research-Compiler Robustness Reference Implementation*, package-closed executable companion with diagnostic aliasing, sparse multi-obstruction and open-set problems, archive contamination, evaluator and integrity attacks, generator-foreign audits, robust controls, tests, and frozen manifests, 2026.

[16] Komorebi P., *SRIC v1.6 Block 15 Compiler-Language Growth from Open-Set Research Conflicts Reference Implementation*, package-closed executable companion with conflict-grounded probe selection, anonymous diagnostic refinement, ordered pair-method synthesis, conservative inherited fallback, matched foreign audits, controls, tests, and frozen manifests, 2026.

[17] Cymela, *Hyper v1 — 3B, Neuralese Thinking*, model documentation and local inference runtime, 2026, https://cymela.com/models/hyper-3b-latent; model repository https://huggingface.co/Cymela/hyper-3b-latent.

[18] Komorebi P., *SRIC v1.7 Block 16 Direct Neuralese Causal-Semantics Reference Implementation*, package-closed direct Hyper adapter contract, synthetic parent–child causal fixture, trace interventions, semantic transport, receipts, tests, figures, and manifest, 2026.

[19] Komorebi P., *SRIC v1.8 Block 17 Executable Neuralese Binding-Repair Compiler Reference Implementation*, package-closed exact affine-program environment, trained recurrent parent, matched repair candidates, answer-confound donor controls, external trace probe, preserved failed pilot, sealed fresh audits, independent NumPy

and gauge verification, receipts, tests, figures, and manifest, 2026.

[20] Komorebi P., *SRIC v1.9 Block 18 Cross-Implementation Neuralese Repair Transport Reference Implementation*, package-closed executable companion with exact program induction, three recurrent implementations, source-compiled FiLM repair transport, sealed foreign audits, failed family receipt, distributional semantic-chart obstruction, post-hoc localization, tests, and frozen manifests, 2026.

[21] Komorebi P., *SRIC v2.0 Block 19 Distribution-Indexed Neuralese Semantic Atlas Growth Reference Implementation*, package-closed executable companion with exact contextual program induction, semantic-twin affine chart compilation, conservative atlas extension, set-structured recurrent Neuralese, selective admission, fresh supported and unsupported audits, independent NumPy replay, tests, figures, and manifest, 2026.

[22] Komorebi P., *SRIC v2.1 Block 20 Atlas-Indexed Cross-Architecture Neuralese Transport Reference Implementation*, package-closed executable companion with a frozen six-chart source atlas, trained GRU and gated-residual targets, controlled architecture–chart gauges, pooled and chart-indexed full-trajectory transport, commutation and curvature audits, paired fresh evaluations, controls, tests, independent NumPy replay, figures, and manifest, 2026.

[23] Komorebi P., *SRIC v2.2 Block 21 Obstruction-Guided Neuralese Transport-Grammar Growth Reference Implementation*, package-closed executable companion with a frozen recurrent packet, hidden affine, nonlinear, temporal, chart-alias, benign-compression, and destructive-collision families, a minimum-description typed compiler, fresh foreign audits, no-go certificates, controls, tests, figures, and independent replay, 2026.

[24] Komorebi P., *SRIC v2.3 Block 22 Correspondence-Free Neuralese Transport from Intervention Orbits Reference Implementation*, package-closed executable companion with independent source and target construction populations, finite response-algebra intertwiners, anonymous intervention matching, declared-gauge certificates, selective partial transport, semantic underidentification, collision-grounded no-go, fresh audits, tests, figures, and deterministic replay, 2026.

[25] Connor Dilgren and Sarah Wiegreffe, *Are Latent Reasoning Models Easily Interpretable?*, arXiv:2604.04902, 2026.

[26] Komorebi P., *SRIC v2.4 Block 23 Endogenous Semantic-Witness Growth and External Latent-Reasoner Pilot Reference Implementation*, package-closed finite causal-predictive quotient compiler, witness-separated hidden audit, semantic-twin, answer-shortcut, gauge, partiality, underidentification, collision no-go, counterfactual packet controls, source-locked Coconut adapter, tests, figures, and frozen receipts, 2026.