SRIC Block 24

External Latent-Reasoner Causal-Semantic Admission - Protocol v0.3.0

Komorebi P.

September 3, 2026

Contents

1 Executive status **3**

2 Research strategy **3**

3 Part I - Brainstorm and design decisions **3**
3.1 Why the next gap is empirical . . . 3
3.2 Candidate external models considered . . . 3
3.2.1 Multimode GPT-2 Coconut on PrOntoQA . . . 3
3.2.2 ProsQA multimode Coconut . . . 4
3.2.3 SLPO Coconut . . . 4
3.2.4 Hyper 3B . . . 4
3.3 Major methodological choices . . . 4
3.3.1 1. Adapter fidelity before model ambition . . . 4
3.3.2 2. Teacher-forced answer loss as the primary effect measure . . . 4
3.3.3 3. Content specificity is not procedure specificity . . . 4
3.3.4 4. Explicit CoT is post-seal only . . . 5
3.3.5 5. Covariance calibration is disjoint . . . 5
3.3.6 6. Semantic final traces are not computed before sealing . . . 5
3.3.7 7. Fail closed . . . 5
3.3.8 8. Smallest valid quotient . . . 5
3.4 Alternatives rejected for the first run . . . 5
3.5 Remaining judgment calls . . . 5

4 Part II - Canonical protocol **6**
4.1 External Latent-Reasoner Causal-Semantic Admission . . . 6
4.1.1 1. Purpose . . . 6
4.1.2 2. Primary scientific question . . . 6
4.1.3 3. Source lock . . . 6
4.1.4 4. Adapter contract . . . 6
4.1.5 5. Population and grouping . . . 6
4.1.6 6. Causal interventions . . . 7
4.1.7 7. Causal status ordering . . . 7
4.1.8 8. Mode controls . . . 7
4.1.9 9. Semantic construction interface . . . 8
4.1.10 10. Candidate grammar . . . 8
4.1.11 11. Candidate gates . . . 8
4.1.12 12. Semantic chronology . . . 8
4.1.13 13. Post-seal proof-instance witness . . . 9
4.1.14 14. Causal subspace patching . . . 9
4.1.15 15. Statistical reporting . . . 9
4.1.16 16. Result interpretation . . . 9
4.1.17 17. Engineered positive control . . . 9

4.1.18 18. Threats to validity . . . 9
4.1.19 19. Hardware plan . . . 10
4.1.20 20. Escalation path . . . 10
4.1.21 21. Stopping rules . . . 10
4.1.22 22. Current execution status . . . 10
4.1.23 23. Decision after the first real receipt . . . 10

5 Engineered positive-control result **11**

6 Current runtime boundary **11**

7 Part III - Source interface and chronology **11**
7.1 Trace identity . . . 11
7.2 Supplied-trace replay . . . 12
7.3 Source parity . . . 12
7.4 Fixed surfaces . . . 12
7.5 Modes . . . 12
7.6 Fingerprint . . . 12
7.7 Failure . . . 12
7.8 Source repository . . . 13
7.9 Primary checkpoint . . . 13
7.10 Base model . . . 13
7.11 Runtime . . . 13
7.12 Secondary planned checkpoint . . . 13

8 Part IV - Real execution status and runbook **14**
8.1 Canonical status . . . 14
8.2 What was executed . . . 14
8.3 What must happen on the user's machine . . . 14
8.4 Failure policy . . . 14
8.5 Before setup . . . 14
8.6 After setup . . . 14
8.7 Smoke run . . . 15
8.8 Primary run . . . 15
8.9 After result . . . 15
8.10 Objective . . . 15
8.11 Recommended first command . . . 15
8.12 Advancement rule . . . 15
8.13 Scientific rule . . . 15
8.14 Returned files to preserve . . . 15
8.15 Current package evidence . . . 16

9 Part V - Structure-sensitive follow-up **17**
9.1 Purpose . . . 17
9.2 Stronger claim . . . 17
9.3 Why the contrast matters . . . 17
9.4 Additional adapter work . . . 17
9.5 Frozen source references . . . 17

10 Part VI - Claim boundary and decision policy **17**
10.1 Current release status . . . 17
10.2 Supported claims . . . 17
10.3 Unsupported claims . . . 18
10.4 Interpretation boundary . . . 18
10.5 Revision boundary . . . 18

1 Executive status

Canonical status: `INFRASTRUCTURE_VALIDATED_EXTERNAL_EXECUTION_PENDING`

Block 24 is the first SRIC package whose intended evidence source is an independently authored and independently trained latent-reasoning checkpoint. It translates the synthetic Neuralese program of Blocks 16-23 into a source-locked external-model admission protocol.

The public checkpoint was **not executed in the build environment**. The release therefore reports no empirical result about Coconut, PrOntoQA, or any other third-party latent reasoner. It establishes the external adapter, source locks, frozen chronology, causal intervention battery, semantic escalation, post-seal witnesses, engineered positive control, and reproducibility machinery required for that run.

A negative real result is fully admissible. The pipeline is designed to classify failure rather than force a positive interpretation.

2 Research strategy

The next major gap is external validity. Another package-local positive control would contribute less evidence than exposing the architecture to an independently authored recurrence and checkpoint. The design therefore prioritizes source fidelity and causal clarity over model size.

The conservative defaults are:

1. GPT-2 multimode Coconut on PrOntoQA is the primary target.
2. PrOntoQA supports a `CONTENT_SPECIFIC` claim, not a different-algorithm claim.
3. Explicit-CoT is revealed only after the semantic quotient is sealed.
4. ProsQA is the first structure-sensitive `PROCEDURE_SPECIFIC` follow-up.
5. Smoke, primary, and full profiles are run in that order without moving scientific thresholds.

3 Part I - Brainstorm and design decisions

3.1 Why the next gap is empirical

Blocks 16-23 progressively established a synthetic theory of latent causal auditing, semantic charts, cross-architecture transport, obstruction-guided repair, unpaired alignment, and endogenous semantic quotient construction. Another synthetic positive control would add less evidence than testing whether the same gates survive contact with a checkpoint designed by an independent group.

The central next question is therefore not whether another controlled fixture can be made to pass. It is whether a source-locked external latent reasoner lands in one of SRIC's predeclared epistemic classes without altering the gates after seeing the result.

3.2 Candidate external models considered

3.2.1 Multimode GPT-2 Coconut on PrOntoQA

Advantages:

- small enough for an 8 GB laptop GPU at batch size one;
- public repository and raw checkpoint;
- latent, direct-answer, and explicit-CoT modes in one trained wrapper;
- recurrent latent vectors are reinjected at identifiable latent-token positions;
- logic answers and proof chains are externally checkable;
- semantic twins and counterfactual query interventions are easy to construct.

Limitation:

- the task family is dominated by fixed-form unary chains. It supports a clean first causal audit but not a strong claim that different latent traces encode different reasoning algorithms.

Decision: use this checkpoint as the primary source-faithful causal admission target, with maximum causal status `CONTENT_SPECIFIC` before semantic escalation.

3.2.2 ProsQA multimode Coconut

Advantages:

- more varied proof structures;
- better setting for structure-near versus structure-far donor contrasts;
- same external research codebase and model family.

Limitations:

- answer parsing and proof-structure extraction require a separate adapter path;
- harder to debug simultaneously with the first source-faithful recurrence adapter.

Decision: reserve as Tier 2. It is the first dataset on which `PROCEDURE_SPECIFIC` should be enabled.

3.2.3 SLPO Coconut

Advantages:

- exposes adaptive stopping information;
- provides latent lengths and gate outputs.

Limitations:

- additional runtime and checkpoint family;
- adaptive stopping introduces another mechanism before basic replay is proven;
- full training experiments exceed the current 8 GB hardware target.

Decision: use after the fixed-length causal audit is stable.

3.2.4 Hyper 3B

Advantages:

- directly recurrent latent reasoning;
- larger and more ambitious external stress.

Limitations:

- at the hardware boundary;
- adapter and memory failures could be confused with theoretical failure;
- current halt behavior is not a strong adaptive-depth positive control.

Decision: later stress test, not the first empirical target.

3.3 Major methodological choices

3.3.1 1. Adapter fidelity before model ambition

A large checkpoint is scientifically useless if the intervention adapter does not reproduce the model's native recurrence. Block 24 therefore gives exact replay and source-code parity priority over scale.

3.3.2 2. Teacher-forced answer loss as the primary effect measure

Free generation is useful descriptively but introduces parsing, decoding, and stochasticity. The main causal effects use teacher-forced protected answer loss while holding the prompt and suffix constant. Free-generation accuracy is a secondary behavioral witness.

3.3.3 3. Content specificity is not procedure specificity

A same-answer donor rules out pure final-answer encoding. It does not automatically prove a different algorithm is represented. On PrOntoQA the result is called content/proof-instance specificity. Procedure specificity is reserved for a task family with varied proof topology and an explicit far-versus-near structural contrast.

3.3.4 4. Explicit CoT is post-seal only

The multimode checkpoint can be given gold proof steps in verbalized mode. Allowing those traces into semantic construction would leak the target ontology. Therefore explicit CoT is revealed only after the quotient seal as a comparison witness.

3.3.5 5. Covariance calibration is disjoint

Covariance-matched noise must not estimate its nuisance distribution from the same rows used to measure the final effect. Twenty percent of proof groups are deterministically reserved for covariance calibration.

3.3.6 6. Semantic final traces are not computed before sealing

The previous prototype split the records correctly but captured every semantic orbit before choosing the quotient. The v0.3 implementation now captures discovery, then calibration, seals the selected quotient, and only then captures final semantic twins and counterfactuals.

3.3.7 7. Fail closed

Replay failure terminates the causal claim. Missing source objects terminate setup. Dirty or wrong upstream commits terminate source verification. No semantic candidate produces `ONTOLOGY_UNDERIDENTIFIED`, not a threshold adjustment.

3.3.8 8. Smallest valid quotient

Semantic construction is validity first, then minimum rank. Full hidden state is a descriptive upper control and is ineligible for selection.

3.4 Alternatives rejected for the first run

- Training a new latent model: confounds model construction with external validation.
- Beginning with a 3B checkpoint: raises adapter and memory risk without improving causal clarity.
- Using explicit CoT during quotient discovery: leaks proof semantics.
- Calling any same-answer transplant procedure-specific: overstates what fixed-chain tasks identify.
- Estimating covariance from the final audit: weakens the intervention's independence.
- Selecting only tasks where latent recurrence helps: creates post-selection bias. The full causal population is reported first.
- Lowering gates after failure: invalidates the sealed experiment.

3.5 Remaining judgment calls

The user was asked whether to prioritize a small clean causal proof or the strongest latent model. No answer was available before implementation, so the conservative default was chosen: small and source-faithful first.

The user was also asked whether explicit CoT should help construct semantics or remain post-seal. The conservative default is post-seal only.

The next strategic choice after the PrOntoQA primary result is whether to proceed immediately to ProsQA or first expand the PrOntoQA final population. The recommended default is to inspect the primary receipt, then use ProsQA only if replay and at least useful ordered recurrence are supported.

4 Part II - Canonical protocol

4.1 External Latent-Reasoner Causal-Semantic Admission

4.1.1 1. Purpose

Block 24 is the first SRIC protocol designed to admit or reject a third-party latent-reasoning checkpoint under a frozen causal and semantic standard. The primary target is the multimode GPT-2 Coconut model trained on PrOntoQA. The protocol treats a negative result as complete evidence when the failure is reproducible and correctly localized.

4.1.2 2. Primary scientific question

Does the recurrent latent trace of the external checkpoint carry content-specific, ordered, structured information that causally contributes to its answer and supports a post-seal low-rank semantic quotient?

The protocol does not initially ask whether every latent step has a unique English translation. It asks whether the latent computation has operationally identifiable causal content.

4.1.3 3. Source lock

The external model is admitted only when all of the following match the bundled lock:

- repository: `connordilgren/are-lrms-easily-interpretable`;
- commit: `32f413d8d55239d9bc54bb6b6ec37b0630891ed4`;
- multimode wrapper Git blob: `d0b353cd78f98ab7aae0d20ce4e8859d9fd9958e`;
- Coconut recurrence Git blob: `646acb08f149afec075eefb212c8aa1f3ac8ceda`;
- requirements Git blob: `4a63ea4b92f0dd565f5d632241e36299695237e4`;
- PrOntoQA test-data Git blob: `7222832226883a545328f2398dc4976b1bb85c7d`;
- checkpoint repository revision: `4ec759a9f9b8a8e7e1a8d1e3e2d43eef969ac83f`;
- checkpoint filename: `checkpoint_37`;
- GPT-2 base revision: `607a30d783dfa663caf39e06633721c8d4cfcd7e`.

The downloaded checkpoint is hashed in full. The complete local base-model payload is hashed as a path-size-file-digest tree with download metadata excluded.

4.1.4 4. Adapter contract

The adapter must expose the exact vectors reinjected at latent-token positions. It must support:

- ordinary latent capture;
- exact supplied-trace replay;
- direct-answer mode;
- post-seal verbalized mode;
- teacher-forced protected answer scoring.

The trace engine is compared against a separately written reference implementation of the reviewed source recurrence. Exact replay is then tested on every audit row. A replay error above the frozen tolerance returns `ADAPTER_INVALID` and terminates all later interpretation.

4.1.5 5. Population and grouping

Tasks are selected deterministically from the pinned public data. Splits are made by `group_id`, not by transformed text row, so semantic twins and related records cannot cross evidence partitions.

For the primary 64-task profile:

- 20% of groups are reserved for covariance calibration;
- the remainder forms the causal audit population;
- if causal admission succeeds, that population is split into 50% semantic discovery, 25% semantic calibration, and 25% semantic final support.

The exact row counts depend on group sizes and are frozen in the pre-audit seal.

4.1.6 6. Causal interventions

For task `i` with clean trace `H_i` and clean answer loss `L_i`, the protocol computes:

4.1.6.1 Exact replay `r_i = max_abs(logits(clean) - logits(replay(H_i)))`.

4.1.6.2 Latent necessity `n_i = L_i(zero trace) - L_i`.

4.1.6.3 Step order `o_i = L_i(permuted trace) - L_i`.

4.1.6.4 Norm-matched structure `c_i = L_i(norm-matched noise) - L_i`.

4.1.6.5 Covariance-matched structure A nuisance covariance is estimated only on the reserved calibration support. The audit effect is:

`c_cov_i = L_i(covariance-matched noise) - L_i`.

4.1.6.6 Same-answer content transplantation A donor has the same final answer, preferentially the same proof depth, a different orbit, and maximal proof-signature distance within the frozen donor population:

`s_i = L_i(H_donor_same_answer) - L_i`.

This is a content/proof-instance specificity test. On PrOntoQA it is not labeled a different-algorithm procedure test.

4.1.6.7 Different-answer transplantation This is a weaker positive control showing sensitivity to answer-bearing task differences.

4.1.6.8 Semantic-twin transplantation The premise order is deterministically changed without changing the protected answer, proof chain, group, or proof signature:

`t_i = L_i(H_twin) - L_i`.

A strong result requires content sensitivity together with twin stability.

4.1.6.9 Truncation The model is scored with zero through K retained latent steps to estimate when near-clean performance appears.

4.1.7 7. Causal status ordering

The status is assigned lexicographically:

1. replay;
2. necessity;
3. order;
4. structured corruption;
5. semantic-twin stability;
6. declared specificity tier.

The first failed gate determines the terminal causal status. This prevents a strong later statistic from compensating for an invalid adapter or unused latent loop.

4.1.8 8. Mode controls

Direct-answer mode is permitted during causal admission because it uses no gold proof. Verbalized mode appends the protected proof steps and is therefore inaccessible before the semantic quotient seal.

The explicit-CoT result is descriptive. It may help interpret a sealed quotient, but it may not determine its basis, rank, thresholds, or acceptance.

4.1.9 9. Semantic construction interface

Semantic escalation occurs only after `CONTENT_SPECIFIC` or `PROCEDURE_SPECIFIC` causal admission.

The constructor sees:

- current latent state;
- next latent state;
- answer-residual future state;
- anonymous query-polarity intervention response;
- final answer label;
- semantic-twin relation.

It does not see the protected proof-instance witness or explicit-CoT trace.

4.1.10 10. Candidate grammar

The finite grammar contains:

- answer-only rank-one direction;
- PCA directions;
- predictive directions;
- intervention-response directions;
- joint predictive-interventional directions;
- equal-rank random directions;
- full-state descriptive upper control.

Candidate bases are canonicalized for deterministic signs and orthonormality.

4.1.11 11. Candidate gates

A candidate must exceed frozen thresholds for:

- future-state `R^2`;
- answer-residual future-state `R^2`;
- intervention-response `R^2`;
- final-answer accuracy;
- semantic-twin drift;
- maximum rank.

Among valid candidates, the minimum rank is selected. Ties are broken by weakest normalized margin and then a frozen method order. Full state is not eligible.

4.1.12 12. Semantic chronology

The order is enforced in code and tested:

1. declare group-disjoint discovery, calibration, and final IDs;
2. capture discovery semantic orbits;
3. capture calibration semantic orbits;
4. construct candidates;
5. evaluate candidates only on calibration;
6. select quotient;
7. hash basis, center, split IDs, and thresholds into the quotient seal;
8. capture final semantic orbits only after that seal exists;
9. fit heads on discovery plus calibration;
10. evaluate final construction metrics;
11. reveal post-seal witness;
12. run causal subspace patching;
13. reveal explicit-CoT comparison.

4.1.13 13. Post-seal proof-instance witness

The current PrOntoQA witness includes answer polarity, recurrent progress, and an audit-only proof-instance coordinate derived from the protected proof signature. It is deliberately described as proof-instance content, not an algorithmic procedure ontology. A stronger ProsQA witness will use explicit graph-structural features.

4.1.14 14. Causal subspace patching

For a selected basis `B`, the projector is `P = B B^T`. A donor semantic packet is inserted while preserving the orthogonal complement:

`H_patch = H_recipient + (H_donor - H_recipient) P`.

The selected packet is compared with:

- full-trace transplant;
- answer-only packet;
- random equal-rank packet;
- PCA packet as a descriptive control.

The causal gate requires a minimum recovery ratio, positive-task fraction, and superiority over answer-only and random shortcuts.

4.1.15 15. Statistical reporting

Effects are reported for the complete causal audit population. Bootstrap intervals resample task rows. For final publication, proof-template or group-level resampling should be preferred whenever public metadata support it.

Correct and incorrect model predictions must be reported separately in the real-result summary. A necessity-positive stratum may be reported only as a secondary predeclared analysis; it may not replace the full population.

4.1.16 16. Result interpretation

- `LATENT_UNUSED`: no evidence that the checkpoint relies materially on its latent loop in this population.
- `LATENT_GENERIC`: evidence for useful ordered latent computation, but not enough same-answer content specificity.
- `CONTENT_SPECIFIC`: the latent trace is tied to proof-instance content beyond final-answer identity.
- `ONTOLOGY_UNDERIDENTIFIED`: no small quotient satisfied the observable causal-predictive contract.
- `SEMANTIC_CANDIDATE`: a quotient passed construction but not the post-seal proof-instance witness.
- `SEMANTICALLY_IDENTIFIED`: the witness passed, but subspace editing did not.
- `CAUSALLY_VALIDATED`: the sealed quotient passed construction, post-seal witness, and causal patching.

No status means that the trace is a complete or unique transcript of hidden thought.

4.1.17 17. Engineered positive control

The package includes a synthetic adapter whose trace contains a low-rank causal content packet and high-variance nuisance dimensions. The fixture validates the transaction and analysis mechanics. It reaches `CAUSALLY_VALIDATED` by design and is always marked `engineered_positive_control: true`.

4.1.18 18. Threats to validity

4.1.18.1 Adapter mismatch Mitigation: source-lock verification, recurrence parity test, exact replay, fixed tokenization and suffix.

4.1.18.2 Final-support leakage Mitigation: final semantic orbits are not captured until after quotient sealing; explicit CoT is post-seal only.

4.1.18.3 Answer shortcut Mitigation: same-answer donors and answer-residual future prediction.

4.1.18.4 Surface shortcut Mitigation: premise-order twins and later distractor/renaming orbit extensions.

4.1.18.5 Procedure overclaim Mitigation: PrOntoQA status is content-specific; procedure-specific status is reserved for varied structural tasks.

4.1.18.6 Noise miscalibration Mitigation: covariance is estimated on group-disjoint calibration rows.

4.1.18.7 Unsupported semantic edit Mitigation: causal patching is compared with equal-rank shortcut controls and can fail independently of probe scores.

4.1.18.8 Post-result threshold changes Mitigation: config, model, source, dataset, donors, splits, and output contract are sealed before execution.

4.1.19 19. Hardware plan

The primary target is intentionally small. The recommended configuration for an 8 GB RTX 3070 Ti is batch size one, deterministic inference, short prompts, capture caching, and moving saved traces to CPU. The package does not retrain the external model.

4.1.20 20. Escalation path

1. Run 12-task smoke solely to validate source parity and replay.
2. Run 64-task primary without changing gates.
3. Inspect the canonical status and failure coordinate.
4. Run the 256-task profile only if the adapter remains valid.
5. Add ProsQA only after PrOntoQA produces a stable causal result.
6. Add CODI or another latent method for cross-model transport only after one checkpoint has a valid semantic object.
7. Add SLPO for adaptive-depth analysis.
8. Add Hyper as a scale stress.

4.1.21 21. Stopping rules

- Replay failure: stop and repair the adapter; do not interpret latent interventions.
- Out-of-memory during smoke: reduce runtime memory, not scientific thresholds.
- `LATENT_UNUSED`: preserve as the primary result; do not filter to successful tasks as the headline.
- `LATENT_GENERIC`: do not run semantic escalation.
- No valid quotient: report `ONTOLOGY_UNDERIDENTIFIED`.
- Causal patch failure: do not call the quotient causally validated.

4.1.22 22. Current execution status

The source-locked external checkpoint was not executed in the environment used to build this package. That environment had CPU-only PyTorch, no Transformers installation, no external checkout, no checkpoint, no public dataset, and no CUDA device. The package's validated result is the engineered fixture only.

4.1.23 23. Decision after the first real receipt

The first real receipt should determine the next move:

- adapter invalid -> repair only adapter fidelity;
- latent unused -> test a second published checkpoint or task, without rewriting the result;
- latent generic -> compare latent and direct modes and stop semantic claims;
- content specific -> execute sealed quotient construction;
- semantic candidate -> improve independent witness design in a new protocol version;
- causally validated -> proceed to ProsQA structure sensitivity and cross-model transport.

5 Engineered positive-control result

The included fixture is deterministic and deliberately contains a low-rank causal semantic packet. It validates the mechanics only; it is not a trained language model.

Quantity	Frozen result
Fixture tasks	48
Covariance calibration tasks	8
Causal audit tasks	40
Causal status	`CONTENT_SPECIFIC`
Semantic status	`CAUSALLY_VALIDATED`
Selected quotient	`joint`, rank 3
Replay maximum	0
Necessity	5.814125 [5.655045, 5.964764]
Order	0.829070 [0.784764, 0.874355]
Norm-matched structure	7.342361 [7.027469, 7.659845]
Covariance-matched structure	12.649418 [12.020959, 13.362811]
Same-answer content transplant	4.408597 [3.409079, 5.523186]
Different-answer transplant	14.829357 [13.318282, 16.259007]
Semantic-twin transplant	0.000000 [0.000000, 0.000000]
Final future R2	0.751820
Final answer-residual future R2	0.636851
Final intervention R2	0.984125
Post-seal proof-witness R2	0.954829
Causal patch passed	`true`

`fixture pre-audit seal: 68b04ea973f5030a7085d2b1891e87bedbf058db007d004fefee047beaee70b9`
`fixture result receipt: 924157159d14b8d3715f2f886c2ad294b55f5cac15a3b63f1552df1b3731e620`
`independent verification: e4e0c5806022faa79e41a363a2100695a8b9d0f33a85c931014f8c732b84c9ca`

This result validates the decision logic, evidence chronology, and intervention plumbing. It establishes nothing about the public checkpoint.

6 Current runtime boundary


```
{
  "checkpoint_exists": false,
  "cuda_available": false,
  "data_exists": false,
  "external_repo_exists": false,
  "ready": false,
  "torch": "2.10.0+cpu",
  "transformers_error": "No module named 'transformers'"
}
```


The third-party checkpoint result remains unknown.

7 Part III - Source interface and chronology

The adapter is valid only when it satisfies all of the following.

7.1 Trace identity

`capture(task)` returns the vectors actually reinjected at the model's latent-token positions, in recurrent order, with shape `[latent_steps, hidden_size]`.

7.2 Supplied-trace replay

`score_with_trace(task, trace)` runs the same teacher-forced prompt and answer suffix while replacing only the reinjected latent vectors. Exact own-trace replay must reproduce answer logits within the frozen tolerance.

7.3 Source parity

The package includes a separate reference recurrence derived from the pinned upstream source. The adapter engine must match it exactly on a controlled PyTorch fixture.

7.4 Fixed surfaces

Across clean and intervention arms, the following remain fixed:

- checkpoint and base model;
- tokenizer and special tokens;
- prompt text for the recipient task;
- token positions;
- attention mask;
- answer suffix;
- teacher-forced label mask;
- decoding settings.

7.5 Modes

- `latent`: admissible for causal and semantic construction.
- `direct`: admissible as a pre-seal mode control.
- `verbalized`: contains gold proof steps and is post-seal only.

7.6 Fingerprint

The adapter fingerprint commits:

- upstream repository and commit;
- source-verification receipt;
- checkpoint repository, revision, filename, and full SHA-256;
- local base-model path, revision, and tree SHA-256;
- latent-step count;
- device type.

7.7 Failure

Any replay, source-lock, cache representation, latent-position, or trace-shape mismatch returns an adapter failure. It must not be treated as evidence about the model's semantics.

```
T0  package code, thresholds, claim tier, source lock frozen
T1  external repository checked out at exact commit
T2  required Git blobs verified and dirty tree rejected
T3  pinned checkpoint and GPT-2 revision downloaded
T4  full checkpoint and local GPT-2 payload hashed
T5  deterministic task population and group IDs frozen
T6  covariance-calibration and causal-audit groups frozen
T7  donor plans and semantic split IDs frozen
T8  pre-audit seal written
T9  calibration traces captured and covariance committed
T10 causal audit executed on disjoint audit rows
T11 causal status assigned
T12 if admitted, semantic discovery traces captured
T13 semantic calibration traces captured
T14 candidate quotient chosen
```

```
T15 quotient basis, center, thresholds, and splits sealed
T16 semantic final traces captured for the first time
T17 post-seal proof-instance witness revealed
T18 causal subspace patching executed
T19 explicit-CoT comparison revealed
T20 canonical result receipt written
```

No T16-T19 object may influence T14. The implementation records and tests the semantic chronology.

The external assets are intentionally not redistributed.

7.8 Source repository

- `connordilgren/are-lrms-easily-interpretable`
- pinned commit: `32f413d8d55239d9bc54bb6b6ec37b0630891ed4`

The package verifies the wrapper, latent runtime, requirements, and selected dataset Git blobs listed in `external_source_lock.json`.

7.9 Primary checkpoint

- repository: `connordilgren/gpt2-prontoqa-multimode-coconut`
- revision: `4ec759a9f9b8a8e7e1a8d1e3e2d43eef969ac83f`
- file: `checkpoint_37`

7.10 Base model

- repository: `openai-community/gpt2`
- revision: `607a30d783dfa663caf39e06633721c8d4cfcd7e`

7.11 Runtime

The isolated real environment pins PyTorch 2.5.1 and Transformers 4.46.2, matching the reviewed external requirements where exact pins were available.

7.12 Secondary planned checkpoint

ProsQA source and checkpoint revisions are recorded in `external_source_lock.json`. They are not executed by the primary v0.3.0 run path.

8 Part IV - Real execution status and runbook

8.1 Canonical status

`EXTERNAL_EXECUTION_NOT_RUN`

The build environment used for v0.3.0 reported:

- CPU-only PyTorch available;
- CUDA unavailable;
- Transformers unavailable;
- pinned upstream checkout absent;
- checkpoint absent;
- public dataset absent.

No attempt was made to substitute a different model or fabricate external metrics. The external checkpoint remains an open empirical test.

8.2 What was executed

- engineered fixture generation;
- engineered pre-audit sealing;
- causal and semantic pipeline on the fixture;
- automated test suite;
- source-lock unit tests;
- recurrence parity fixture;
- chronology and split-isolation tests;
- independent recomputation of fixture means and semantic-basis properties;
- package closure and manifest verification.

8.3 What must happen on the user's machine

1. Run `Setup_External_PrOntoQA_Coconut.ps1`.
2. Run the smoke profile.
3. Inspect replay and source verification.
4. Run the primary profile without modifying the frozen config.
5. Preserve the complete `frozen`, `results`, `capture_cache`, and environment records.

8.4 Failure policy

The first real status is the result. An adapter failure permits an adapter-only revision because no semantic claim is yet admissible. Any scientific threshold revision requires a new protocol and new final support.

8.5 Before setup

☐ At least 25 GB free disk space.
☐ NVIDIA driver operational.
☐ Python 3.10, 3.11, or 3.12 available through `py`.
☐ Git available in PowerShell.
☐ Network access to GitHub, PyPI, PyTorch wheels, and Hugging Face.

8.6 After setup

☐ `runtime_paths.json` exists.
☐ upstream commit equals the source lock.
☐ source blob verification passes.
☐ checkpoint revision and full SHA-256 are recorded.
☐ GPT-2 revision and local tree SHA-256 are recorded.
☐ `pip check` passes.
☐ CUDA is visible to the isolated environment.

8.7 Smoke run

☐ Seal is written before model audit.
☐ Exact replay passes all 12 rows.
☐ Prompt, suffix, latent positions, and answer mask are stable.
☐ No out-of-memory or cache-format error occurs.

8.8 Primary run

☐ Configuration file unchanged from the sealed copy.
☐ Covariance calibration rows are group-disjoint from audit rows.
☐ Donor plans cover every causal audit row once.
☐ Explicit CoT remains unavailable before semantic seal.
☐ Final semantic orbits are captured only after quotient seal.
☐ Complete result and environment records are retained even on failure.

8.9 After result

☐ Run `verify-result`.
☐ Generate summary.
☐ Copy the complete result directory before any exploratory analysis.
☐ Do not relabel content specificity as algorithmic procedure specificity.

8.10 Objective

Execute a source-locked causal-semantic audit of the public multimode GPT-2 Coconut PrOntoQA checkpoint without changing the v0.3.0 gates after result access.

8.11 Recommended first command

```
.\Setup_External_PrOntoQA_Coconut.ps1
```

Then:

```
.\Run_Real_LRM_Pilot.ps1 -Profile smoke -Device cuda
```

The smoke run is an adapter and replay qualification. It has semantic escalation disabled.

8.12 Advancement rule

Proceed to `primary` only when:

- source verification passes;
- the checkpoint and base-model hashes are recorded;
- exact replay is below tolerance;
- no cache or token-position discrepancy is observed;
- GPU memory is stable.

8.13 Scientific rule

Do not alter `configs/prontoqa_multimode_coconut_gpt2.json` after staging. If the primary result is negative, preserve it. A new hypothesis requires a new version and seal.

8.14 Returned files to preserve

- `external_runtime/environment.json`
- `external_runtime/runtime_paths.json`
- `frozen/real_primary_preaudit_seal.json`
- donor plans and experiment plan beside the seal
- `results/real_primary_block24.json`
- `results/real_primary_summary.md`

- `results/capture_cache/` if present

8.15 Current package evidence

The engineered positive control passed. The external model has not been executed. No real-model conclusion should be inferred from the fixture.

9 Part V - Structure-sensitive follow-up

9.1 Purpose

PrOntoQA is the source-faithful causal admission tier. ProsQA is the intended structure-sensitive tier. It should be activated only after the PrOntoQA adapter passes replay.

9.2 Stronger claim

A ProsQA result may use `PROCEDURE_SPECIFIC` only when all of the following are implemented before final audit:

1. proof graphs or derivation structures are extracted independently of text surface;
2. donors are matched on final answer and broad difficulty;
3. a structure-near donor and structure-far donor are assigned for every recipient;
4. the primary statistic is the contrast
   `loss(far donor) - loss(near donor)`;
5. the contrast has a positive frozen interval and positive-task fraction;
6. semantic twins preserve proof topology while changing wording, entity names, and premise order;
7. final proof structures are held out by template family.

9.3 Why the contrast matters

A harmful arbitrary donor can reflect task identity, vocabulary, or context formatting. The far-minus-near contrast asks whether disruption grows with formal solution-structure distance after answer and difficulty are controlled.

9.4 Additional adapter work

ProsQA answers are not limited to Boolean labels. A separate answer parser and protected loss contract are required. The current PrOntoQA adapter must not be silently reused as if these formats were identical.

9.5 Frozen source references

The external source lock records the ProsQA test-data Git blob, checkpoint repository, checkpoint revision, and checkpoint filename. Those entries are planning locks, not evidence that the ProsQA checkpoint has been executed.

10 Part VI - Claim boundary and decision policy

10.1 Current release status

`INFRASTRUCTURE_VALIDATED_EXTERNAL_EXECUTION_PENDING`

The package validates a source-faithful intervention architecture, a staged chronology, and an engineered causal-semantic positive control. The real public checkpoint was not run in the build environment.

10.2 Supported claims

The release supports the following statements:

1. The adapter contains a separately tested implementation of the reviewed Coconut recurrence and exposes supplied-trace replay.
2. Exact source commit and required Git blobs are pinned and verifiable.
3. Checkpoint and base-model revisions are pinned by setup; local payload hashes are sealed at staging.
4. Covariance-matched noise uses a group-disjoint calibration support in the Block 24 pipeline.
5. Semantic discovery, calibration, quotient sealing, final capture, proof witness, causal patching, and explicit-CoT comparison occur in the declared order.
6. The engineered fixture reaches `CAUSALLY_VALIDATED` and all included automated tests pass.
7. PrOntoQA's primary maximum specificity claim is content/proof-instance specificity, not proof of multiple internal reasoning algorithms.

10.3 Unsupported claims

The release does not establish that the real Coconut checkpoint:

- relies on latent recurrence;
- carries ordered latent computation;
- represents proof-instance content beyond answer identity;
- is stable under semantic twins;
- exposes a low-rank semantic quotient;
- supports causal semantic editing;
- matches its explicit natural-language proof mode;
- generalizes to ProsQA or any other task;
- is safe, conscious, sentient, or recursively self-improving.

The engineered fixture cannot be used as evidence for any external model.

10.4 Interpretation boundary

Even a real `CAUSALLY_VALIDATED` result would mean only that, within the pinned model, task population, intervention language, and frozen thresholds, a compact latent subspace predicts and causally mediates declared proof-instance consequences. It would not reveal a verbatim chain of thought or prove that the discovered coordinates are the unique correct semantics.

10.5 Revision boundary

Any threshold, donor rule, counterfactual generator, source revision, checkpoint, model mode, or claim tier changed after a real result must produce a new protocol version and a new pre-audit seal. A failed run remains part of the evidence lineage.

The next action is the 12-task source-parity and replay smoke run on the user's RTX 3070 Ti.

After the primary PrOntoQA result:

- `ADAPTER_INVALID`: repair adapter fidelity only.
- `LATENT_UNUSED`: preserve the result and consider a different public latent checkpoint.
- `LATENT_GENERIC`: stop semantic escalation; compare execution modes descriptively.
- `CONTENT_SPECIFIC`: complete the sealed semantic escalation.
- `CAUSALLY_VALIDATED`: proceed to the ProsQA structure-sensitive protocol before any broader claim.

Default recommendation: do not proceed directly to Hyper or training. Resolve the smallest external checkpoint first.