Projector-Resolved Large Algebraic Signal Model

Exact Multi-Atlas Coordinates, Bayesian Operator Programs, Residual-Driven Refinement, Memory, and Active Planning

A signal-native world model in which spatial, Fourier, wavelet, polyphase, and categorical coordinates coexist inside one exact reconstruction system, while learned operator programs remain typed, probabilistic, and auditable.

Komorebi P.

Abstract

This paper presents the Projector-Resolved Large Algebraic Signal Model (PR-LASM), a finite, executable world-model architecture built from exact projector decompositions, coordinate transport, polyphase refinement, operator programs, Bayesian evidence, residual-driven adaptation, memory packets, and active planning. Each observation is retained in a complete multi-atlas archive consisting of spatial, unitary Fourier, orthonormal Haar, two-dimensional polyphase, and categorical color coordinates. These atlases are not concatenated informally: their weighted direct sum is proved to be a finite tight frame with an explicit canonical synthesis map. The model then forms deterministic interpretable features and nested macro, meso, and micro states for statistical prediction without confusing those compressed states with the exact archive.

The learned dynamics are a posterior over a typed finite grammar containing linear, quadratic, action-bilinear, delayed, power-memory, exponential-memory, phase-conditioned, gated, and hybrid programs. A normal-inverse-gamma regression is fitted inside every program, and predictive evidence is combined with explicit complexity and mechanism penalties. Exact deterministic replay transitions are stored in a state-action graph; the planner combines known graph outcomes, replay probes, posterior progress, win and loss probabilities, novelty, information value, role statistics, and loop penalties.

The release contains exact reconstruction tests, adaptive mixed-depth packet-tree synthesis, controlled program-identification benchmarks, deterministic interactive planning tests, a static audit of twenty-five supplied local environments, a reproducible Python package, and offline run scripts. All thirty-six automated tests pass. On seven controlled algebraic systems, the intended operator family and intended atlas are identified in every case. The median state-prediction RMSE relative to a single linear portfolio baseline is 0.2393. Four deterministic interactive systems are solved. No official benchmark score or claim of general intelligence is made.

Status. The exact finite coordinate and reconstruction theorems, program definitions, Bayesian formulas, graph algorithms, and supplied tests are complete under the hypotheses written here. The controlled benchmarks verify the implementation of the declared mechanisms. The twenty-five bundled environment trees are statically validated, but they were not executed with the official local runtime in the present build environment.

Contents

1 The final construction problem **4**

1.1 What “large” means . . . 4

1.2 The strongest defensible claim . . . 5

2 The PR-LASM object **5**

3 Exact multi-atlas coordinates **6**

3.1 A redundant tight-frame atlas . . . 7

4 Concrete coordinate atlases **8**

4.1 Spatial coordinates . . . 8

4.2 Fourier coordinates . . . 8

4.3 Haar-wavelet coordinates . . . 9

4.4 Two-dimensional polyphase coordinates . . . 10

4.5 Categorical color packets . . . 11

4.6 Reconstruction certificates . . . 12

5 From exact coordinates to nested statistical states **13**

6 Adaptive exact packet refinement **14**

7 The typed operator-program grammar **16**

7.1 Linear and action-bilinear programs 16

7.2 Quadratic packet interactions 17

7.3 Delay and memory programs 17

7.4 Phase-conditioned programs 18

7.5 Gated and hybrid programs 18

8 Bayesian operator selection **18**

8.1 Posterior predictive mixture 19

9 Exact transition memory **19**

10 Active planning **20**

11 Residual-driven model evolution **21**

12 Algorithms **22**

12.1 Exact multi-atlas packetization 22

12.2 Program fitting 22

12.3 Active agent loop 23

13 Computational organization **23**

14 Controlled program-identification benchmark **24**

14.1 Linear ablation 26

15 Deterministic interactive benchmarks **27**

16 The supplied twenty-five-environment suite **28**

17 Software architecture **29**

18 Verification **30**

19 Claim ledger **31**

20 What has been completed **32**

21 Remaining research directions **33**

21.1 Program grammar growth . . . 33
21.2 Learned invertible atlases . . . 33
21.3 Continuous and stochastic worlds . . . 34
21.4 Scaling and hardware . . . 34
21.5 Empirical evaluation . . . 34

22 Conclusion **34**

A Exact transform formulas **35**
A.1 Haar analysis and synthesis . . . 35
A.2 Polyphase analysis and synthesis . . . 35
A.3 Fourier selector projectors . . . 35

B Bayesian regression formulas **35**

C Reproducibility commands **36**
C.1 Linux or macOS shell . . . 36
C.2 PowerShell . . . 36
C.3 Local twenty-five-environment run . . . 36

D Machine-readable result files **36**

E Test map **36**

1 The final construction problem

The technical series developed seeds, projectors, transfer blocks, nonlinear interactions, polyphase refinement, special functions, stochastic and quantum realizations, computational packet algorithms, arithmetic packets, and local-to-global completion. The remaining engineering question is whether those ingredients can be assembled into one coherent model of a changing signal world.

A conventional predictor usually commits early to one representation. A vector model receives coordinates that have already forgotten geometry. A convolutional model privileges translation-local structure. A Fourier model privileges global frequency. A wavelet model privileges scale and location. A graph model privileges relational neighborhoods. None of these choices is wrong, but no single one is uniformly natural.

PR-LASM keeps several exact coordinate atlases simultaneously. It then lets the posterior decide which atlas, resolution, operator family, and memory mechanism best explain the observed transitions. The exact frame remains recoverable regardless of that choice.

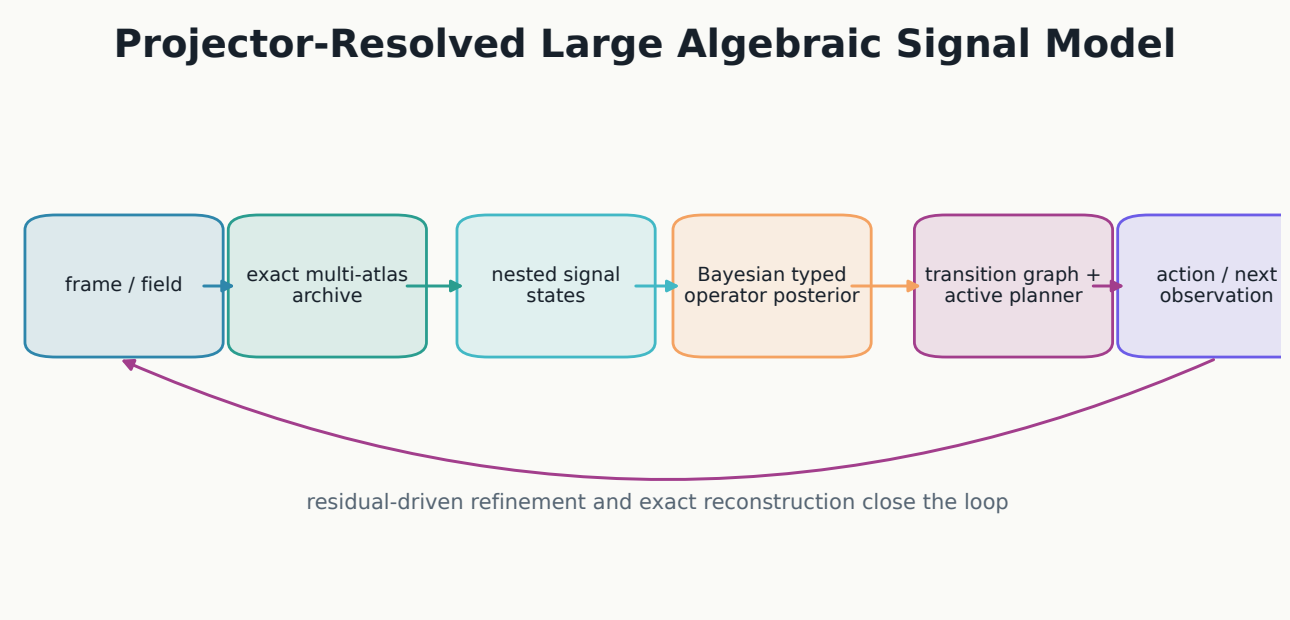


Figure 1: The completed PR-LASM architecture. Exact analysis, statistical inference, graph memory, active planning, and reconstruction form one loop.

1.1 What “large” means

The word *large* does not refer only to parameter count. The model is large because it maintains a structured family of possibilities:

$$\boxed{\text{atlas} \times \text{resolution} \times \text{operator program} \times \text{memory} \times \text{packet refinement} \times \text{action}.}$$

A small number of parameters inside many typed alternatives can express more structural uncertainty than one dense map with the same total coefficient count. The model is algebraic because every alternative has an explicit coordinate map, feature map, operator design, probability, and reconstruction status.

1.2 The strongest defensible claim

The architecture is a complete candidate for a projector-resolved, exact-reconstruction, multi-atlas, fully algebraic large signal model. "Complete" here means that the finite architecture closes its own declared loop: observation, exact analysis, compression, inference, memory, planning, action, residual measurement, refinement, and reconstruction. It does not mean that the operator grammar contains every possible law of nature or that the planner solves every interactive environment.

2 The PR-LASM object

Definition 2.1 (Projector-resolved large signal model)**.** *A finite PR-LASM instance is a tuple*

$$\mathfrak{L} = (X, \mathfrak{U}, \mathfrak{E}, \mathfrak{P}, \mathfrak{Z}, \mathfrak{G}, \mathfrak{B}, \mathfrak{H}, \mathfrak{R}, \mathfrak{C}),$$

where:

1. X *is the observation space;*
2. $\mathfrak{U} = \{U_a\}_{a \in A}$ *is a family of exact coordinate atlases;*
3. $\mathfrak{E} = \{E_{a,\alpha}\}$ *is the family of coordinate selectors;*
4. $\mathfrak{P} = \{P_{a,\alpha}\}$ *is the pulled-back projector family;*
5. $\mathfrak{Z}$ *is a nested family of statistical states;*
6. $\mathfrak{G}$ *is a typed operator-program grammar;*
7. $\mathfrak{B}$ *is the Bayesian posterior over programs and parameters;*
8. $\mathfrak{H}$ *is the explicit memory state;*
9. $\mathfrak{R}$ *is the state-action transition graph;*
10. $\mathfrak{C}$ *is the planning and reconstruction contract.*

The tuple separates exact coordinates from learned states. This is not merely editorial. It prevents a low prediction error in a compressed state from being reported as exact object reconstruction.

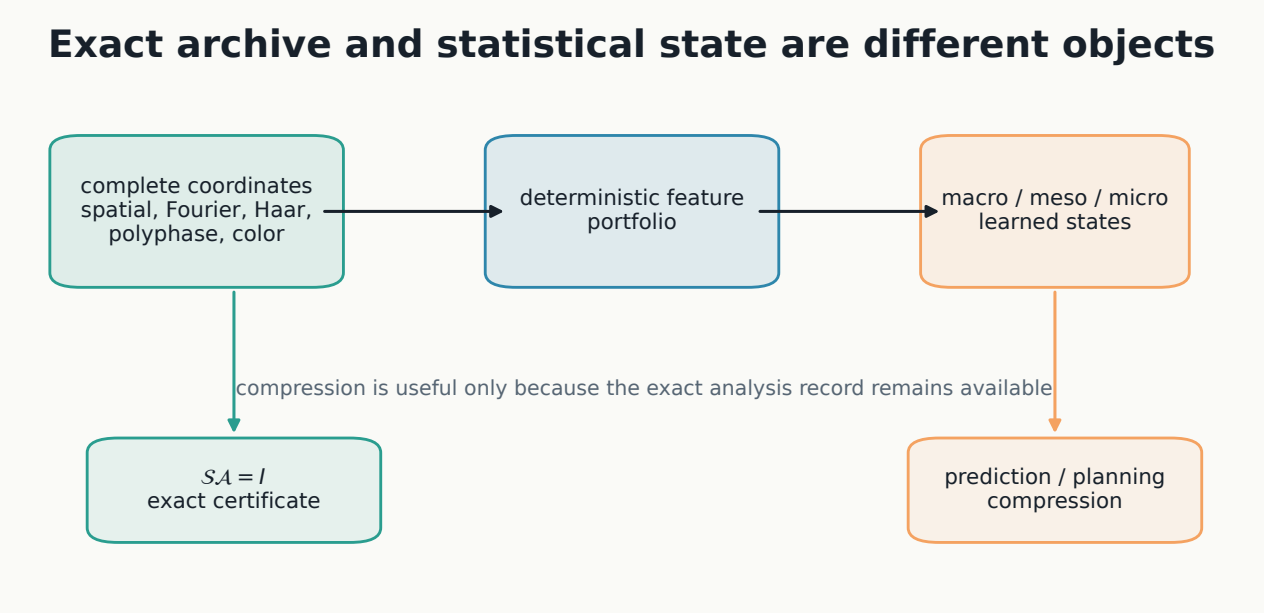


Figure 2: The exact coordinate archive and the statistical state serve different purposes.

3 Exact multi-atlas coordinates

Let X be a finite-dimensional Hilbert space. For every atlas $a \in A$, let

$$U_a : X \to Y_a$$

be unitary, and let $\{E_{a,\alpha}\}_{\alpha \in I_a}$ be orthogonal coordinate projectors satisfying

$$E_{a,\alpha}E_{a,\beta} = \delta_{\alpha\beta}E_{a,\alpha}, \qquad \sum_{\alpha \in I_a} E_{a,\alpha} = I_{Y_a}.$$

Define the object-space packets

$$\boxed{P_{a,\alpha} = U_a^* E_{a,\alpha} U_a.}$$

Theorem 3.1 (Per-atlas packet reconstruction)**.** *For every atlas a, the operators $P_{a,\alpha}$ are orthogonal projectors and*

$$\sum_{\alpha \in I_a} P_{a,\alpha} = I_X.$$

Consequently,

$$x = \sum_{\alpha} P_{a,\alpha}x, \qquad T = \sum_{\alpha,\beta} P_{a,\beta}TP_{a,\alpha}.$$

Proof. Conjugation by a unitary operator preserves multiplication, adjoints, and sums. Therefore

$$P_{a,\alpha}P_{a,\beta} = U_a^* E_{a,\alpha}E_{a,\beta}U_a = \delta_{\alpha\beta}P_{a,\alpha},$$

and the complete sum is $U_a^* I_{Y_a} U_a = I_X$. Inserting that identity on one or both sides gives object and operator reconstruction. □

3.1 A redundant tight-frame atlas

Choose weights $w_a > 0$ and define

$$\mathcal{A}x = (w_a U_a x)_{a \in A},$$

with synthesis

$$\mathcal{S}(c_a)_{a \in A} = \frac{1}{c} \sum_{a \in A} w_a U_a^* c_a, \qquad c = \sum_{a \in A} w_a^2.$$

Theorem 3.2 (Exact multi-atlas tight-frame reconstruction)**.** *The analysis and synthesis maps satisfy*

$$\boxed{\mathcal{S}\mathcal{A} = I_X.}$$

Moreover,

$$\|\mathcal{A}x\|^2 = c \|x\|^2 .$$

Proof. By unitarity,

$$\mathcal{S}\mathcal{A}x = \frac{1}{c} \sum_a w_a U_a^* (w_a U_a x) = \frac{1}{c} \sum_a w_a^2 x = x.$$

Similarly,

$$\|\mathcal{A}x\|^2 = \sum_a w_a^2 \|U_a x\|^2 = c \|x\|^2 .$$

□

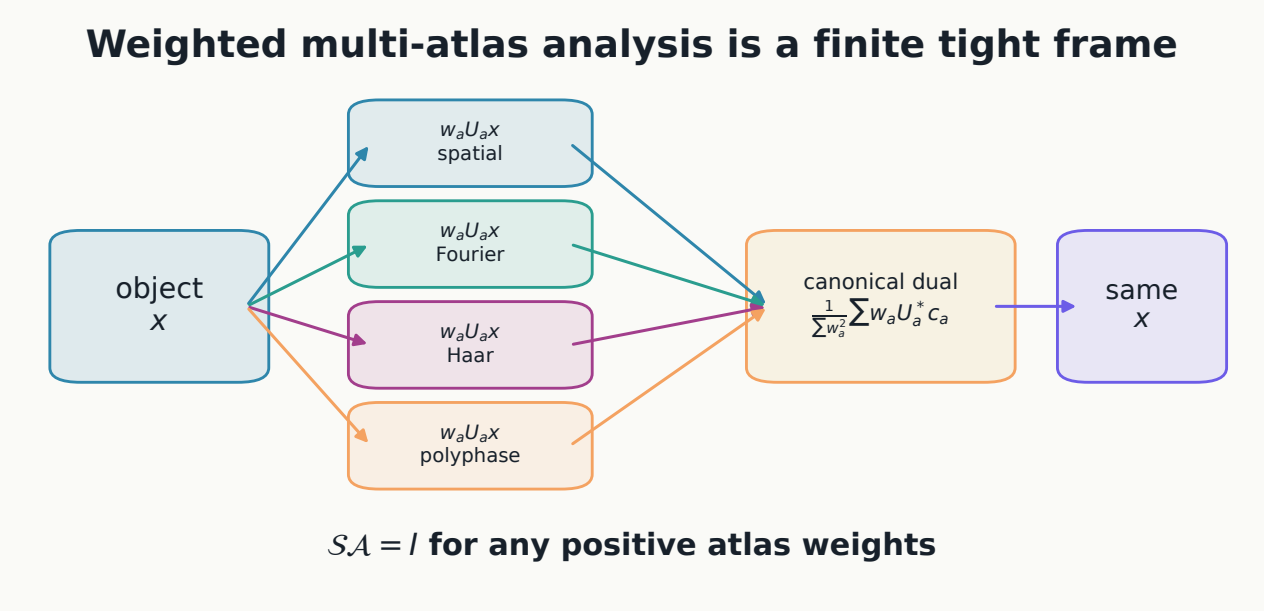


Figure 3: The canonical dual averages exact reconstructions with the correct frame normalization.

The normalized positive packets

$$\widetilde{P}_{a,\alpha} = \frac{w_a^2}{c} P_{a,\alpha}$$

satisfy

$$\sum_{a,\alpha} \widetilde{P}_{a,\alpha} = I_X.$$

They need not be mutually annihilating across different atlases. They are frame packets, not one enlarged orthogonal projector family. This distinction prevents the simultaneous use of several atlases from being mistaken for a direct-sum decomposition inside the original space.

4 Concrete coordinate atlases

4.1 Spatial coordinates

The identity atlas stores the observation itself. It is the natural chart for local geometry, connected components, boundaries, coordinates, and direct pixel or field values.

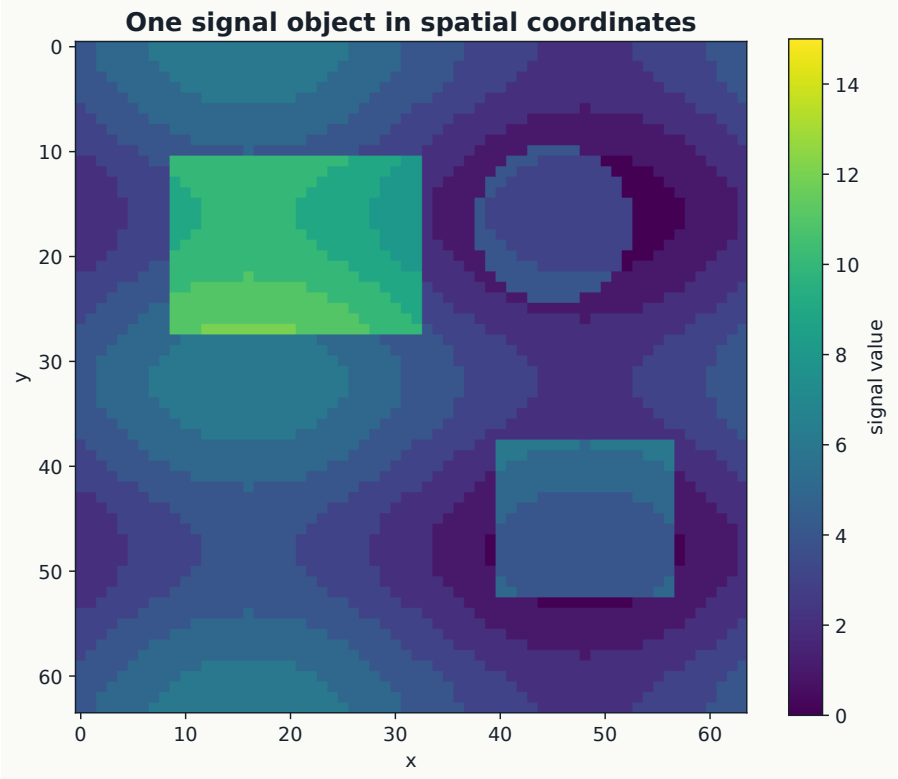


Figure 4: A sample finite signal in its spatial chart.

4.2 Fourier coordinates

For an $H \times W$ array, the normalized two-dimensional transform is

$$(Fx)_{k,\ell} = \frac{1}{\sqrt{HW}} \sum_{m=0}^{H-1} \sum_{n=0}^{W-1} x_{m,n} e^{-2\pi i(km/H + \ell n/W)}.$$

The inverse is F^*. Disjoint frequency masks E_B produce projectors

$$P_B = F^* E_B F.$$

Fourier space therefore appears inside the same packet language as any other unitary atlas.

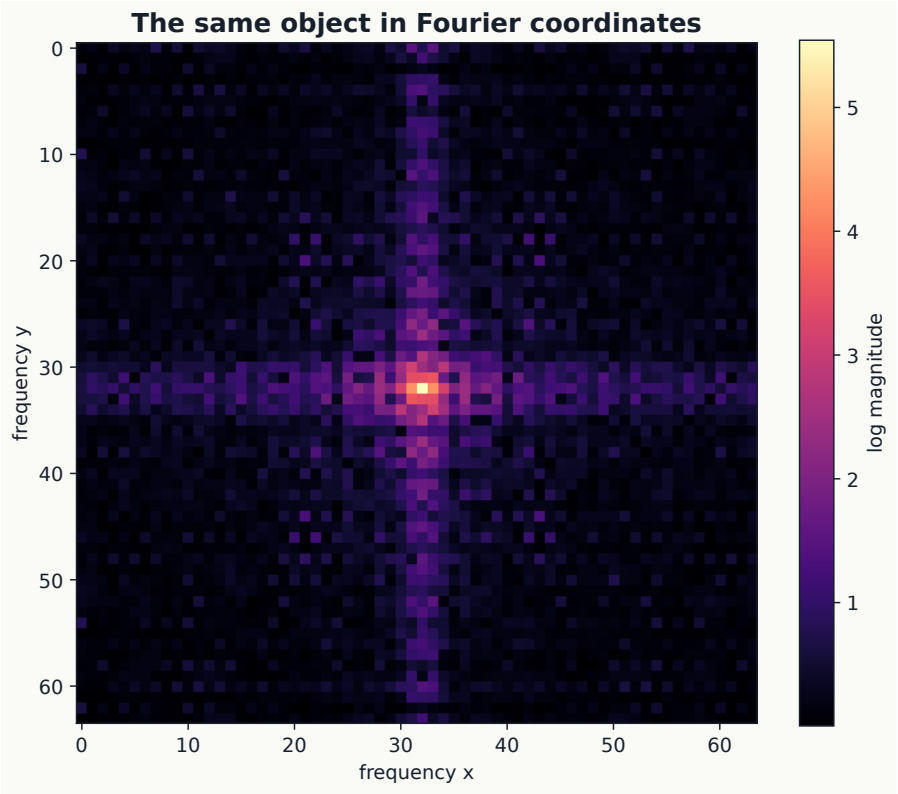


Figure 5: Global oscillatory structure becomes visible in Fourier coordinates.

4.3 Haar-wavelet coordinates

The separable Haar transform repeatedly applies

$$(a, b) \longmapsto \left(\frac{a+b}{\sqrt{2}}, \frac{a-b}{\sqrt{2}} \right)$$

along columns and rows. The inverse reverses those operations. Scale-location selectors in the packed transform pull back to exact wavelet projectors.

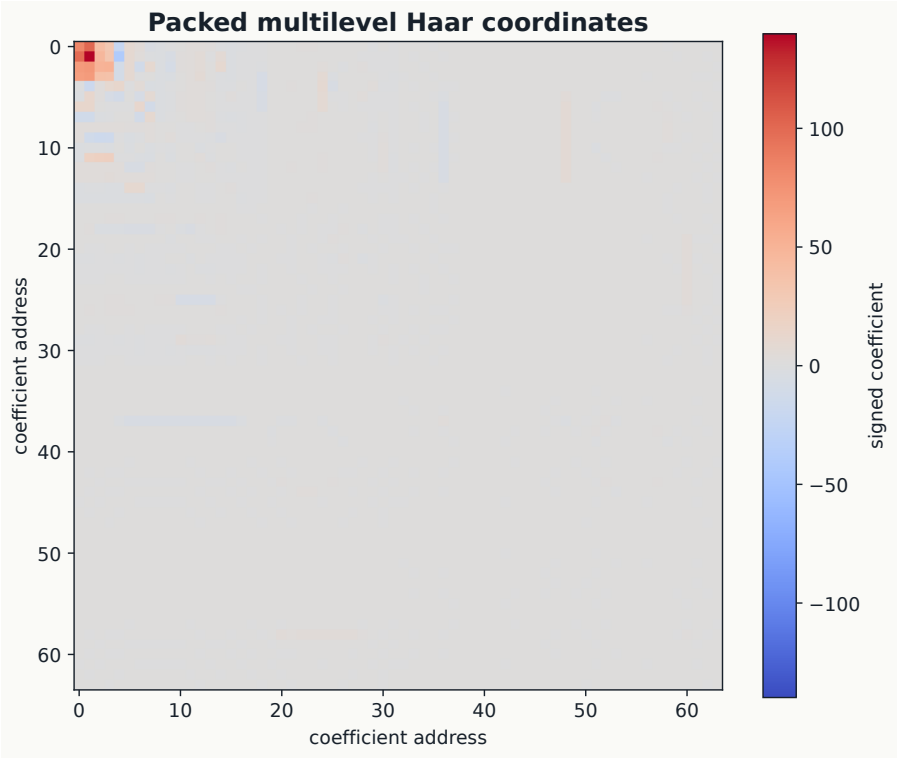


Figure 6: The same object in a multilevel wavelet chart.

4.4 Two-dimensional polyphase coordinates

For modulus N, define

$$x_{r,s}[q,t] = x[Nq + r, Nt + s], \qquad 0 \le r, s < N.$$

The complete channel family is exactly inverted by interleaving. In object space, the phase projector is

$$(P_{N,r,s}x)[m,n] = \mathbf{1}_{m\equiv r \ (N)}\mathbf{1}_{n\equiv s \ (N)}x[m,n].$$

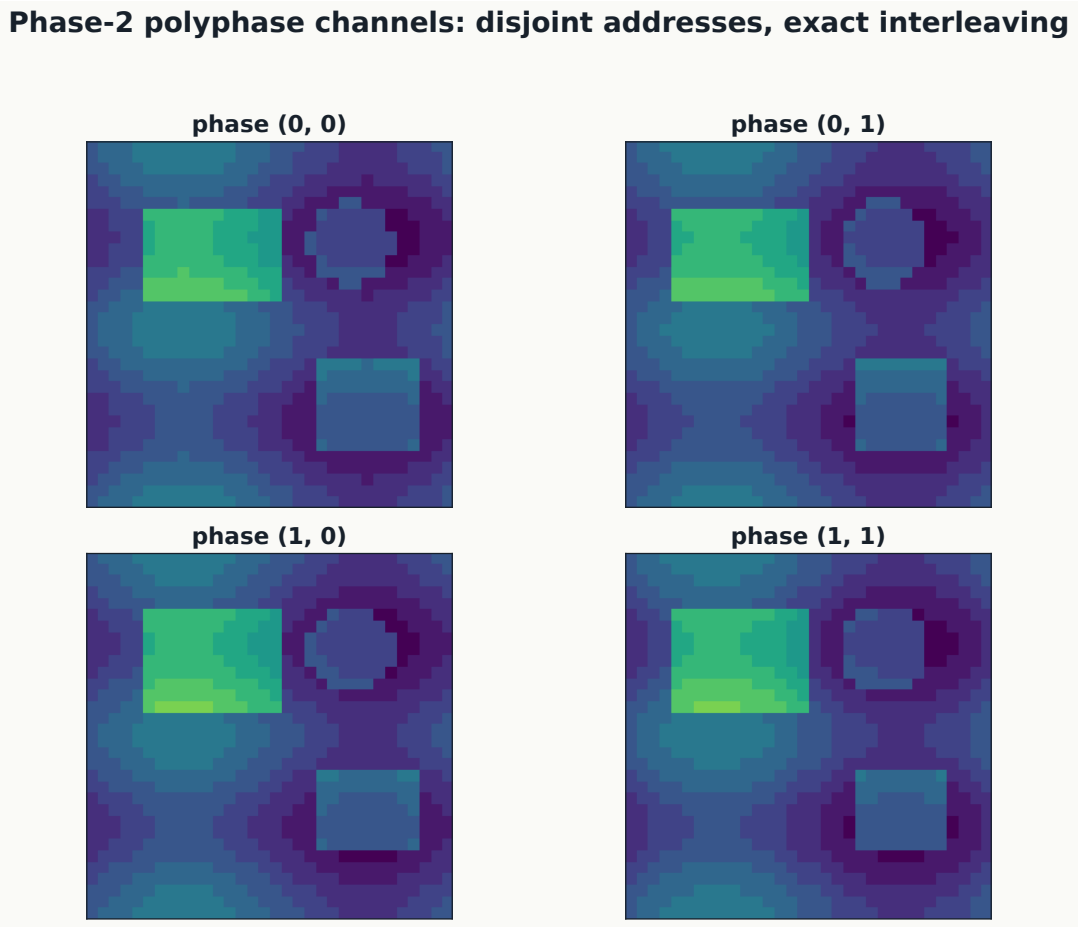


Figure 7: Polyphase coordinates expose address classes without dropping samples.

4.5 Categorical color packets

A finite-valued frame can be encoded by the one-hot coordinates

$$C_j[m, n] = \mathbf{1}_{x[m,n]=j}.$$

The original frame is recovered by the unique active color at each address. This reconstruction is categorical rather than linear over the original scalar encoding, so it is kept as a separately typed exact atlas.

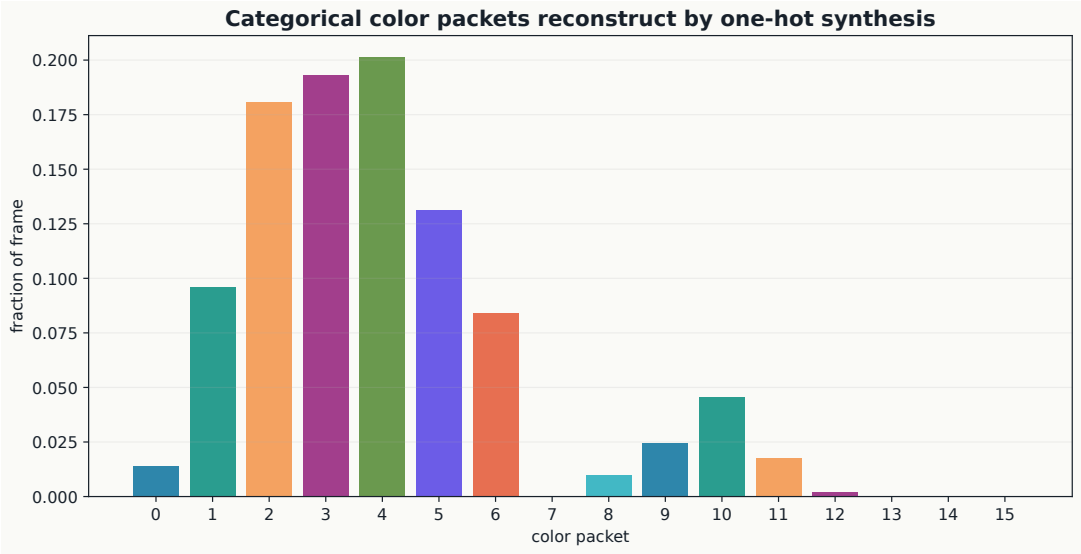


Figure 8: Color packets preserve categorical mass and exact label reconstruction.

4.6 Reconstruction certificates

Finite certificate 4.1 (Exact atlas residuals). Across the tested frames, the maximum observed residuals were

atlas	maximum absolute residual
spatial	0
Fourier	3.55×10^{-15}
Haar	2.49×10^{-14}
polyphase-2	0
polyphase-4	0
categorical color	0
weighted multi-atlas frame	5.33×10^{-15}

The nonzero values are double-precision roundoff.

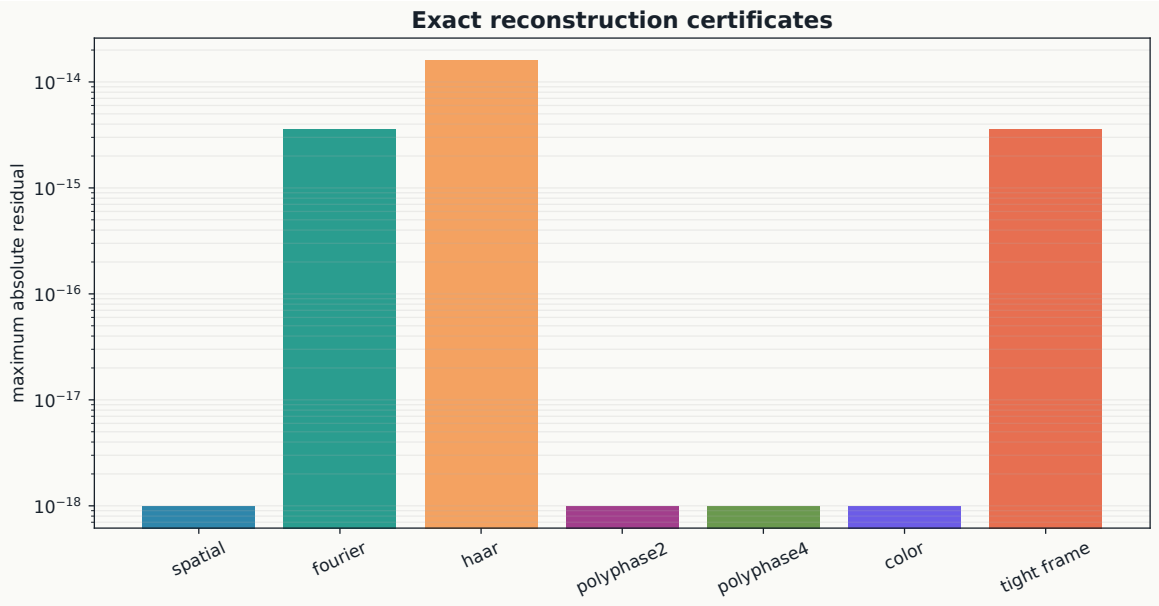


Figure 9: The exact coordinate archive is accompanied by explicit numerical residuals.

5 From exact coordinates to nested statistical states

The exact archive is too large to place directly inside every candidate regression. PR-LASM therefore computes interpretable summaries:

- pooled spatial values at several block sizes;
- color masses and first spatial moments;
- radial and angular Fourier energies plus low-frequency complex coefficients;
- wavelet subband energies and coarse signed coefficients;
- phase-channel means, deviations, and occupancy values;
- connected-component, boundary, and change descriptors inherited from the interactive packetizer.

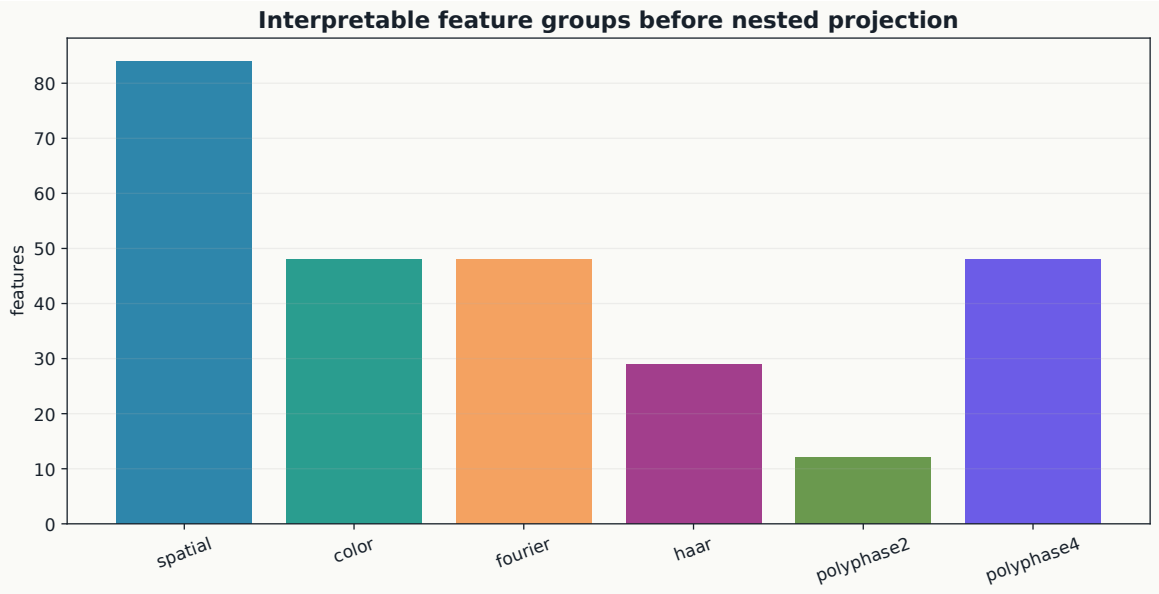


Figure 10: The feature groups remain named before they are projected.

Let $r(x) \in \mathbb{R}^d$ be the concatenated feature vector. A deterministic seeded QR factorization produces an orthonormal row system Q. The nested states are

$$z^{\text{macro}} = Q_{1:k_1} r(x), \qquad z^{\text{meso}} = Q_{1:k_2} r(x), \qquad z^{\text{micro}} = Q_{1:k_3} r(x),$$

with

$$k_1 < k_2 < k_3 < d.$$

The supplied dimensions are 48, 96, and 192.

Boundary 5.1 (Projection status). The nested states are deterministic compressions. They are not claimed to reconstruct the original frame. Exact reconstruction belongs to the coordinate archive, not to the statistical projections.

6 Adaptive exact packet refinement

A packet description should become finer only where the added structure improves the declared objective. The release uses a complete Haar packet tree on a finite vector.

Definition 6.1 (Complete pruning)**.** *A set Λ of packet-tree nodes is a complete pruning when every root-to-leaf path meets Λ exactly once.*

Theorem 6.2 (Exact mixed-depth synthesis)**.** *Every complete pruning reconstructs the root signal exactly.*

Proof. Every selected node either is the root or has a unique sibling selected or reconstructed at the same depth. Applying inverse Haar synthesis to sibling pairs moves the selected representation one level upward. Repeating finitely many times produces the root. The complete-pruning condition prevents both missing branches and overlapping ancestor-descendant selections. □

Let $c(v)$ be any finite additive node cost. Define recursively

$$V(v) = \min \left\{ c(v), \sum_{u \in \text{ch}(v)} V(u) \right\}.$$

Theorem 6.3 (Finite best-basis dynamic program)**.** *The recursion returns a globally minimum-cost complete pruning in time linear in the number of tree nodes.*

Proof. At a leaf, the only complete pruning is the leaf. At an internal node, every complete pruning either keeps the node or uses complete prunings of both children. Induction on depth proves optimality, and each node is evaluated once. □

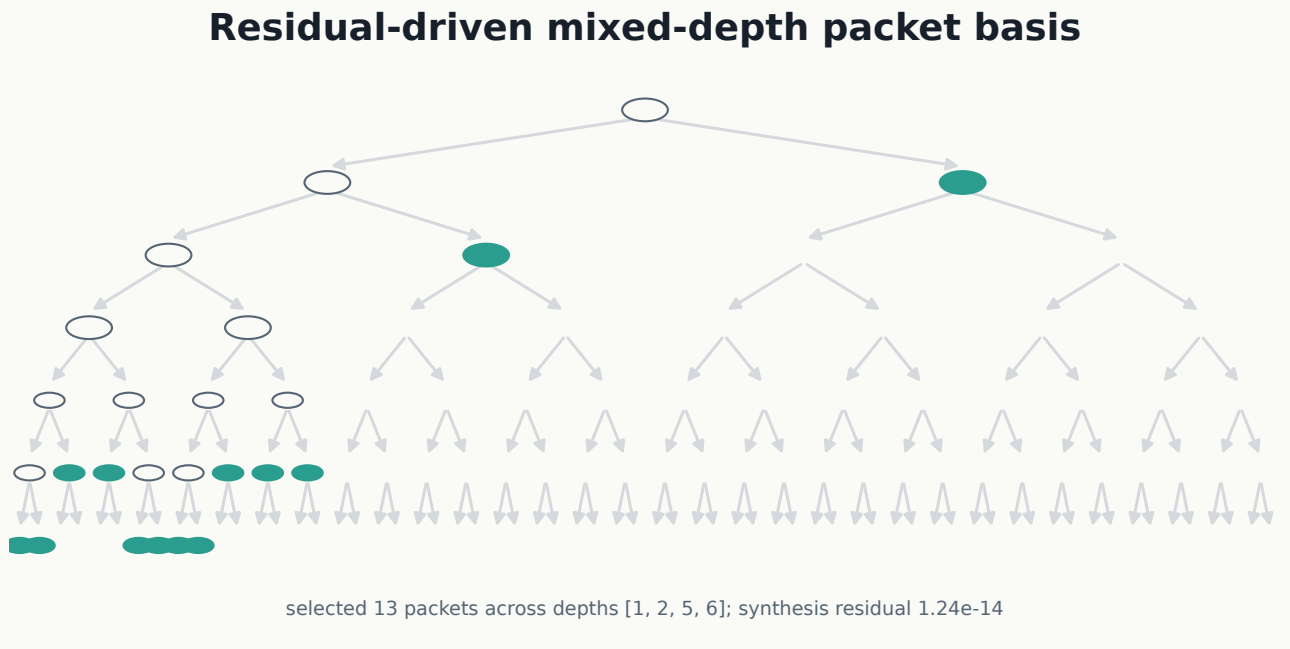


Figure 11: A mixed-depth complete pruning selected by dynamic programming.

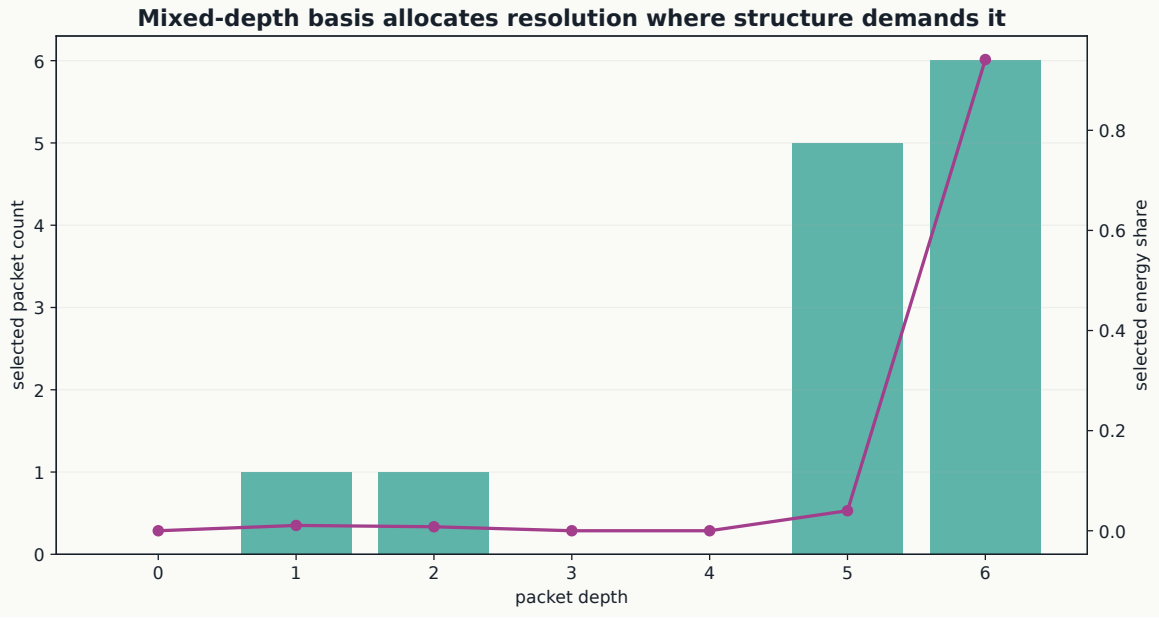


Figure 12: Selected packet counts and energy distribution across refinement depth.

In twenty-four structured frame tests, the selected basis used between 11 and 61 packets, with mean 32.58. The maximum reconstruction residual was 1.95×10^{-14}.

7 The typed operator-program grammar

A large signal model should not hide all mechanism uncertainty inside one dense matrix. PR-LASM uses a typed program

$$g = (a, r, f, \kappa),$$

where a is an atlas, r is a resolution, f is an operator family, and κ contains family-specific ranks or periods.

For a packet state z_t, atlas features q_t, action features u_t, and memory state m_t, every program defines a design map

$$\phi_g(z_t, q_t, u_t, m_t, t) \in \mathbb{R}^{d_g}.$$

The target is

$$y_t^{(g)} = \begin{bmatrix} z_{t+1}^{(g)} - z_t^{(g)} \\ R_t/100 \\ \Delta L_t \\ \mathbf{1}_{\text{win}} \\ \mathbf{1}_{\text{loss}} \\ \mathbf{1}_{\text{changed}} \end{bmatrix}.$$

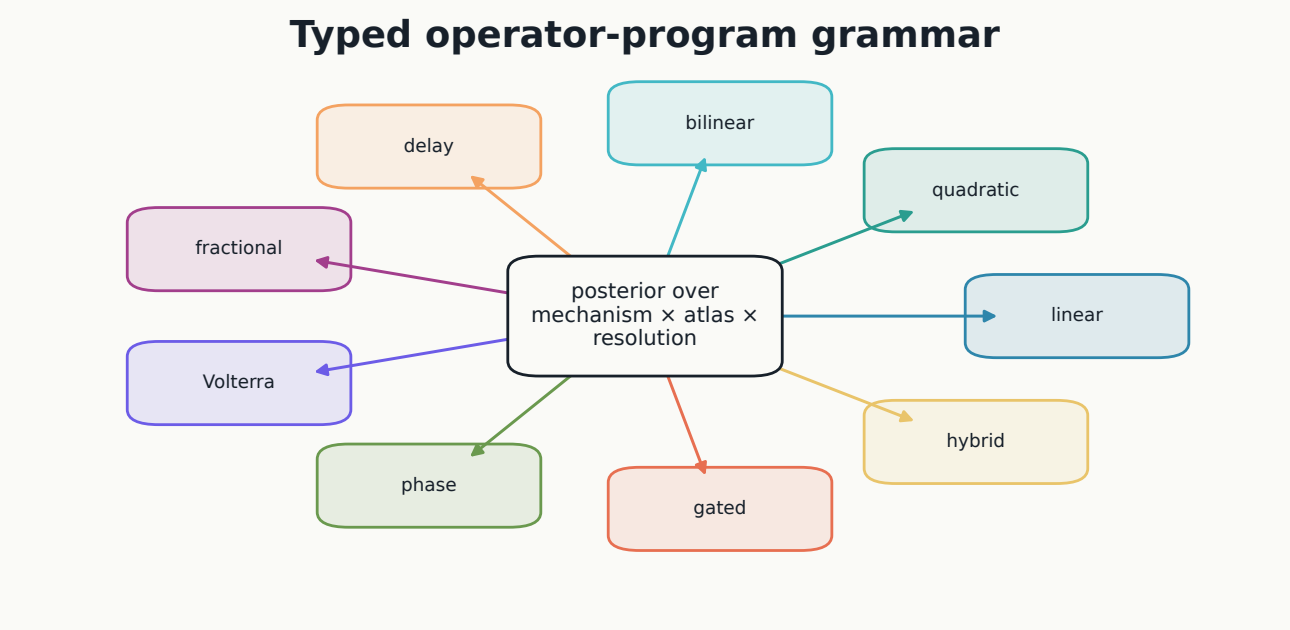


Figure 13: The finite grammar keeps mechanism names and penalties explicit.

7.1 Linear and action-bilinear programs

The basic linear program uses

$$\phi_{\text{lin}} = [1, z_t, q_t, u_t].$$

The action-bilinear extension adds products

$$z_{t,i}u_{t,j}, \qquad q_{t,i}u_{t,j},$$

for a declared low interaction rank.

7.2 Quadratic packet interactions

The quadratic program adds

$$z_{t,i}z_{t,j}, \qquad q_{t,i}q_{t,j}, \qquad i \le j,$$

within a controlled rank. This is a finite packet jet, not an unrestricted polynomial expansion.

7.3 Delay and memory programs

The delayed program appends z_{t-1}. The power-memory packet is

$$m_t^{\mathrm{pow}} = \frac{\sum_{\ell=1}^{H} \ell^{-\alpha} z_{t-\ell}}{\sum_{\ell=1}^{H} \ell^{-\alpha}},$$

while the exponential packet is

$$m_t^{\mathrm{exp}} = \frac{\sum_{\ell=1}^{H} \rho^{\ell-1} z_{t-\ell}}{\sum_{\ell=1}^{H} \rho^{\ell-1}}.$$

The supplied defaults are $\alpha = 0.68$ and $\rho = 0.82$.

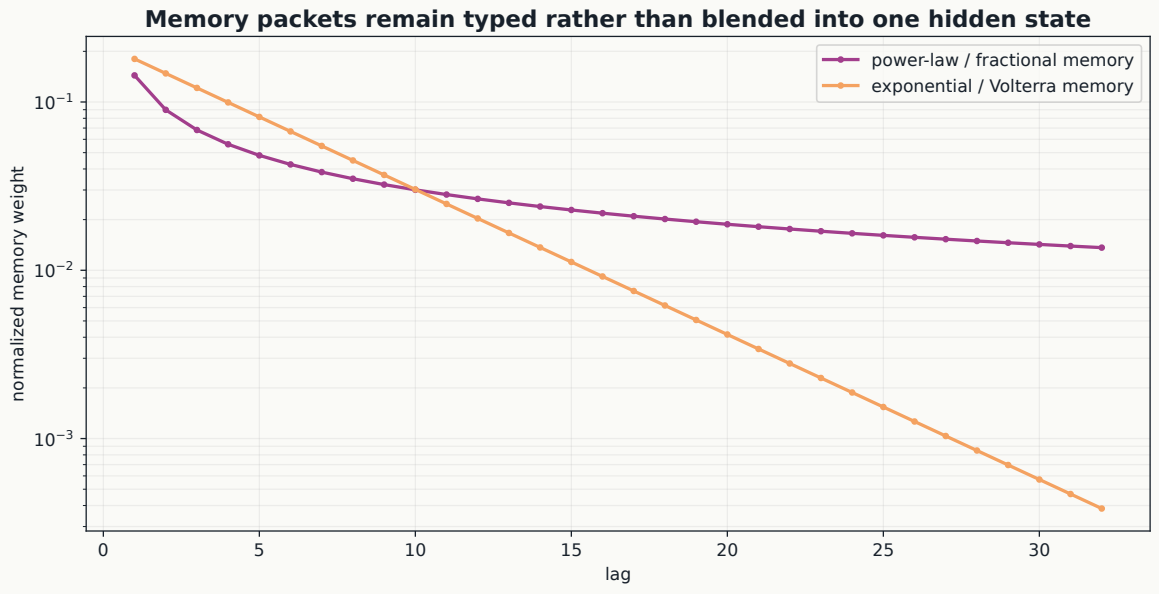


Figure 14: Power-law and exponential histories are retained as distinct memory mechanisms.

7.4 Phase-conditioned programs

For modulus N, the phase program appends the one-hot phase

$$e_{t \bmod N}$$

and interactions $z_{t,i} e_{t \bmod N}$. Alternating, cyclic, or clocked dynamics therefore become linear in an expanded polyphase state.

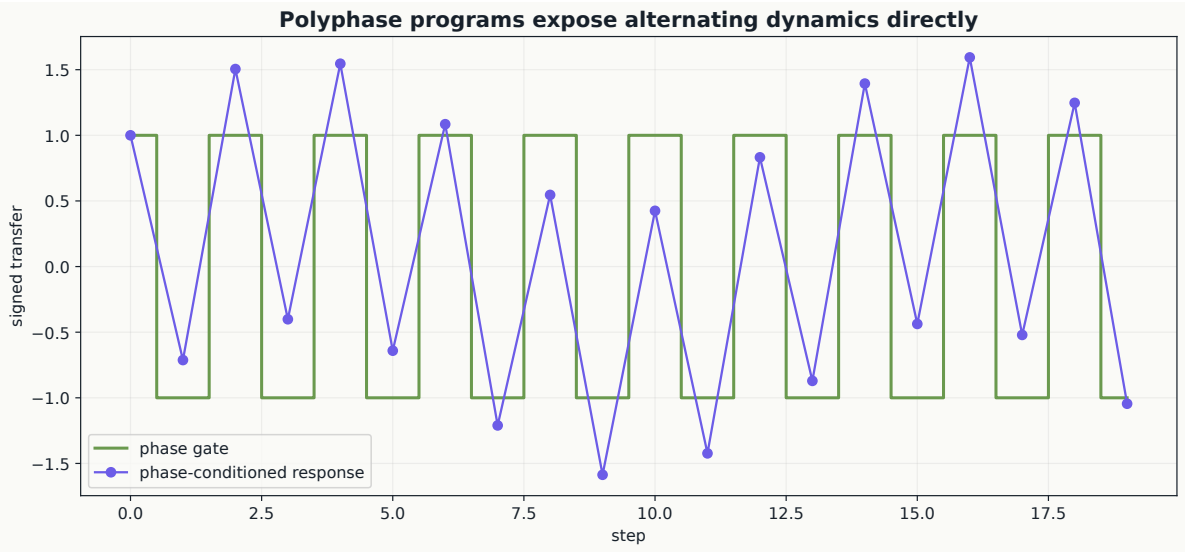


Figure 15: A periodic mechanism is simpler in its natural phase chart.

7.5 Gated and hybrid programs

The gated program uses a smooth gate

$$\gamma_t = \frac{1}{1 + e^{-4\bar{q}_t}}$$

and appends $\gamma_t z_t$ and $(1 - \gamma_t) z_t$. The hybrid program contains a controlled combination of quadratic, delayed, power-memory, exponential-memory, and interaction terms. Its higher prior penalty prevents it from winning merely because it has more columns.

8 Bayesian operator selection

For one program g, write

$$Y = X_g B_g^\top + \varepsilon,$$

with independent output variances and a normal-inverse-gamma prior. If the prior precision is $\Lambda_0 = \lambda_0 I$, then

$$\Lambda_n = \Lambda_0 + X_g^\top X_g, \qquad M_n = Y^\top X_g \Lambda_n^{-1}.$$

The posterior coefficient covariance is Λ_n^{-1}, and each output variance has an inverse-gamma posterior.

The implementation computes held-out Student predictive likelihood when enough rows are available. The program log weight is

$$\boxed{W_g = \ell_{\text{pred}}(g) - \lambda\ \text{Complexity}(g) - \eta\ \text{Mechanisms}(g) - \rho\ \text{ResolutionPenalty}(g).}$$

The posterior is

$$p(g \mid D) = \frac{e^{W_g}}{\sum_h e^{W_h}}.$$

8.1 Posterior predictive mixture

If program g predicts scalar vector μ_g with variance V_g, then

$$\bar{\mu} = \sum_g p_g \mu_g,$$

and the total predictive variance is

$$\boxed{V_{\text{total}} = \underbrace{\sum_g p_g V_g}_{\text{within-program}} + \underbrace{\sum_g p_g (\mu_g - \bar{\mu})^2}_{\text{between-program}}.}$$

The second term is model disagreement and drives information-seeking behavior.

9 Exact transition memory

Each resolved frame receives a deterministic hash of its complete normalized array. An observed public action defines a graph edge

$$(s, a) \longmapsto s'.$$

Repeated observations accumulate reward, progress, win, loss, and visit counts.

Definition 9.1 (Deterministic replay edge)**.** *An edge is deterministic when repeated replay of the same action history and current action returns the same state signature. If two targets are observed, the edge is marked nondeterministic and is excluded from exact shortest-path reasoning.*

The graph supports:

- known winning and progressing actions;
- untried actions at the current state;
- frontier states with incomplete action coverage;
- shortest known paths to a frontier;
- loop and failure counts.

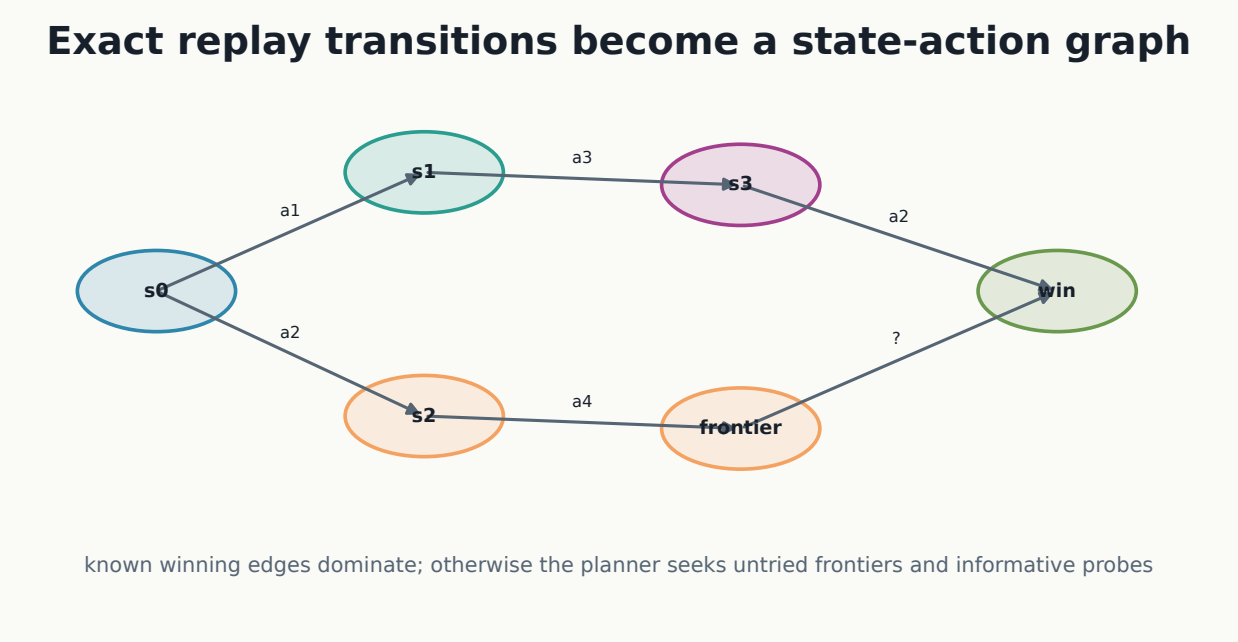


Figure 16: Exact graph memory complements rather than replaces the learned world model.

Proposition 9.2 (Replay safety rule)**.** *If replay of the committed history produces a signature different from the expected current signature, all later replay probes are disabled for that run.*

This rule sacrifices probe efficiency rather than treating a nondeterministic or hidden-state environment as deterministic.

10 Active planning

For candidate action a, the planner forms a utility

$$\begin{aligned} U(a) = {} & \widehat{R}(a) + \lambda_L \widehat{\Delta L}(a) + \lambda_W \widehat{p}_{\text{win}}(a) - \lambda_D \widehat{p}_{\text{loss}}(a) \\ & + \lambda_N \,\text{Novelty}(a) + \lambda_I \,\text{Information}(a) - \lambda_V \,\text{Variance}(a) \\ & + \lambda_H \,\text{RoleHistory}(a) - \lambda_Q \,\text{RepeatCount}(a)^2 + \lambda_F \mathbf{1}_{\text{untried}}. \end{aligned}$$

Exact probe outcomes replace the corresponding predicted terms. Known graph wins and progress receive dominant bonuses. This ordering ensures that an already observed winning action is not ignored because a statistical model has not yet fit it.

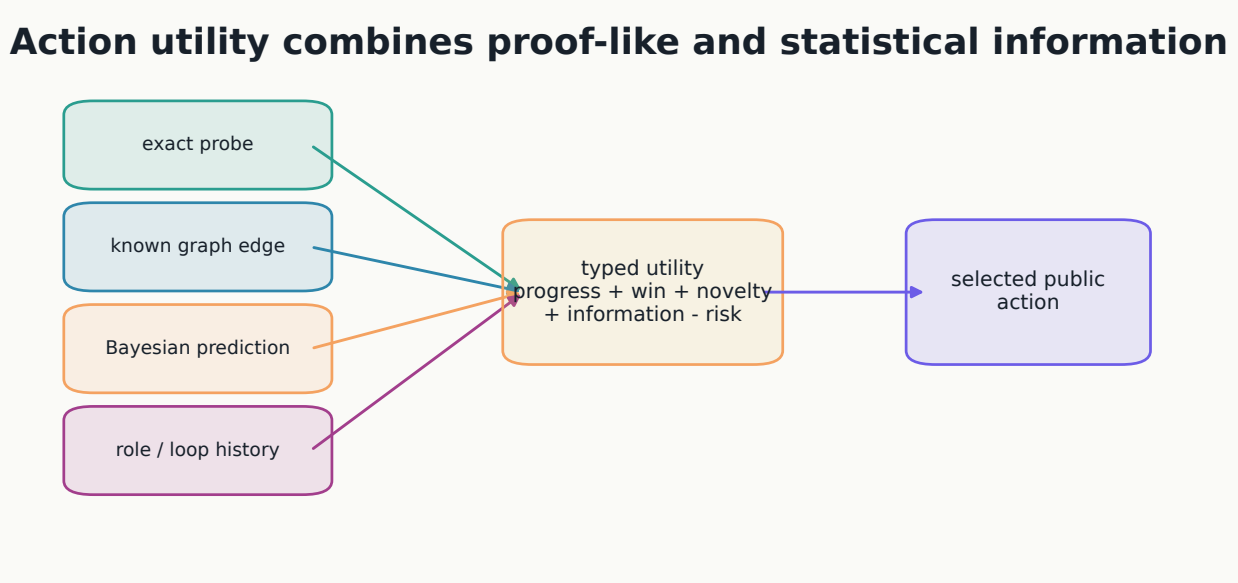


Figure 17: The planner combines exact and probabilistic evidence without merging their logical types.

11 Residual-driven model evolution

The model loop is

analyze exactly → fit typed posterior → predict and act → measure residual → refine selectively.

A residual can trigger:

- a finer statistical projection;
- a different atlas expert;
- a packet-tree split;
- a delay or memory mechanism;
- an action interaction;
- an explicit graph edge rather than a learned approximation.

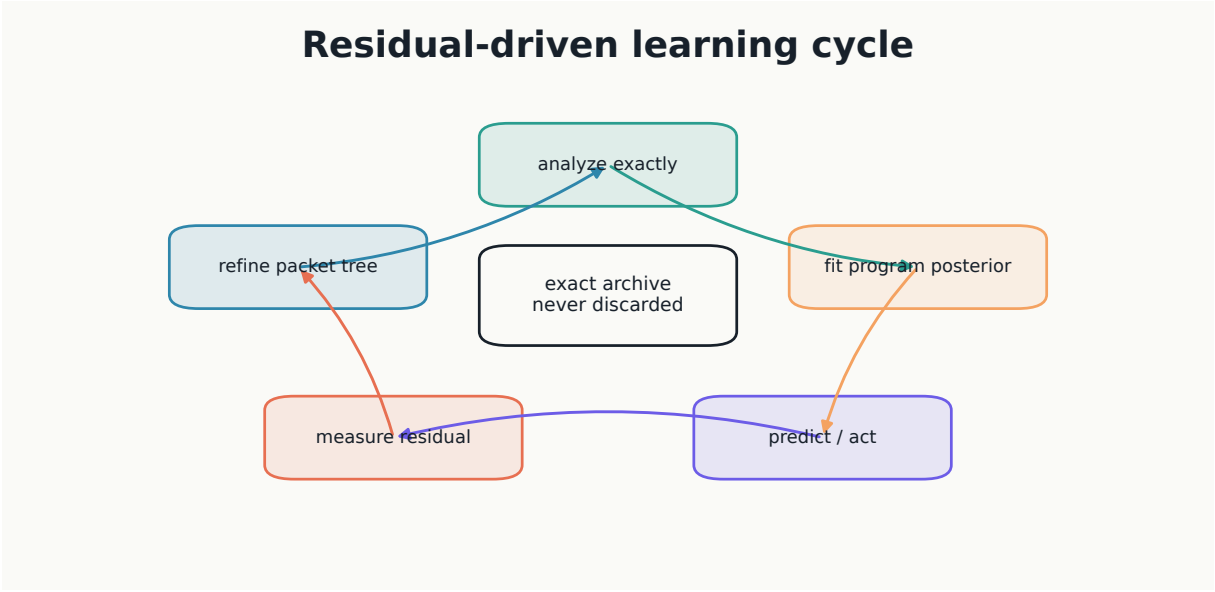


Figure 18: The exact archive remains fixed while the predictive view changes.

12 Algorithms

12.1 Exact multi-atlas packetization

Listing 1: Core exact analysis contract.

```
analysis = analyzer.analyze(frame)
assert max(analysis.reconstruction_residuals().values()) < tolerance
exact = analysis.tight_frame_synthesis(weights)
assert max_abs(exact - frame) < tolerance
features = analyzer.feature_groups(analysis)
z_macro = Q[:48] @ concatenate(features)
z_meso = Q[:96] @ concatenate(features)
z_micro = Q[:192] @ concatenate(features)
```

12.2 Program fitting

Listing 2: Typed Bayesian program comparison.

```
for program in grammar:
    X = stack(program.design_transition(row) for row in records)
    Y = stack(program.targets(row) for row in records)
    posterior = normal_inverse_gamma_fit(X_train, Y_train)
    score = predictive_log_likelihood(posterior, X_valid, Y_valid)
    weight[program] = score - complexity_penalty(program)
probability = softmax(weight)
```

12.3 Active agent loop

Listing 3: PR-LASM planning loop.

```
while budget_remains and not terminal:
    packets = exact_and_statistical_analysis(observation)
    candidates = public_action_candidates(observation, packets)
    probes = deterministic_replay_probes(history, candidates)
    if enough_records:
        world_model.fit(memory)
    scores = utility(candidates, probes, graph, posterior, memory)
    action = argmax(scores)
    next_observation = environment.step(action)
    record_transition(observation, action, next_observation)
    update_graph_and_residuals()
```

13 Computational organization

Structured transforms are applied before dense learning. The exact Fourier transform uses FFT structure, the Haar transform uses local butterfly operations, and polyphase extraction is a permutation of addresses. Dense linear algebra is confined to projected states and program-specific design matrices.

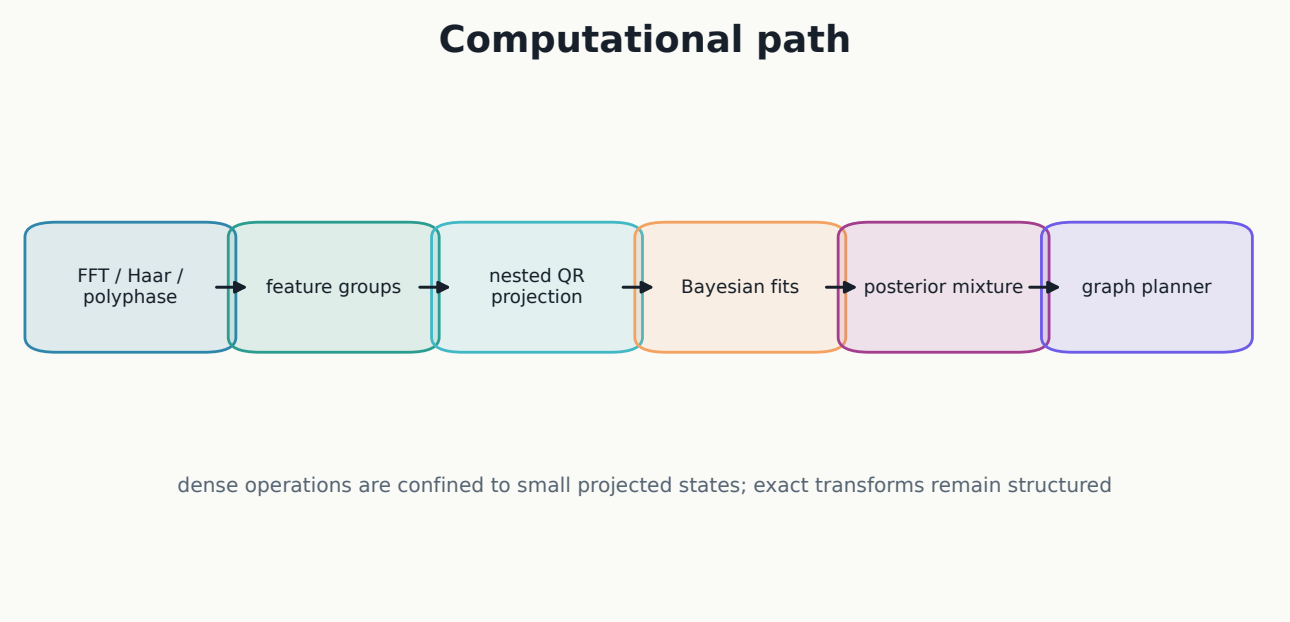


Figure 19: The implementation keeps full-resolution transforms structured and small-state inference dense.

For n signal coordinates, A exact unitary atlases, projected state dimension k, G candidate programs,

and m observed transitions, the dominant modeled costs are:

FFT atlas	$O(n \log n)$,
Haar atlas	$O(n)$,
polyphase atlas	$O(n)$,
feature projection	$O(dk)$,
one regression fit	$O(md_g^2 + d_g^3)$,
all program fits	$\sum_{g=1}^{G} O(md_g^2 + d_g^3)$.

The architecture is designed so that d_g is controlled by the selected rank and resolution rather than by the full signal dimension.

14 Controlled program-identification benchmark

The mechanism benchmark is deliberately algebraic. It does not claim to reproduce the uncertainty of arbitrary physical or interactive data. Its purpose is narrower: verify that when data are generated by one typed mechanism, the implementation can distinguish that mechanism from simpler alternatives on held-out records.

Seven systems were generated:

1. Fourier-atlas linear transfer;
2. spatial quadratic interaction;
3. delayed portfolio dynamics;
4. Fourier power-memory dynamics;
5. portfolio exponential-memory dynamics;
6. phase-two polyphase dynamics;
7. color-gated dynamics.

For each system, 84 records were used for fitting and 36 independent records for testing. The target-driving variables absent from simpler candidate programs were independently randomized. This prevents the benchmark from being solved by time-index memorization.

Finite certificate 14.1 (Mechanism and atlas identification). The highest posterior family matched the declared generating family in all seven systems. The highest posterior atlas matched the declared atlas in all seven systems.

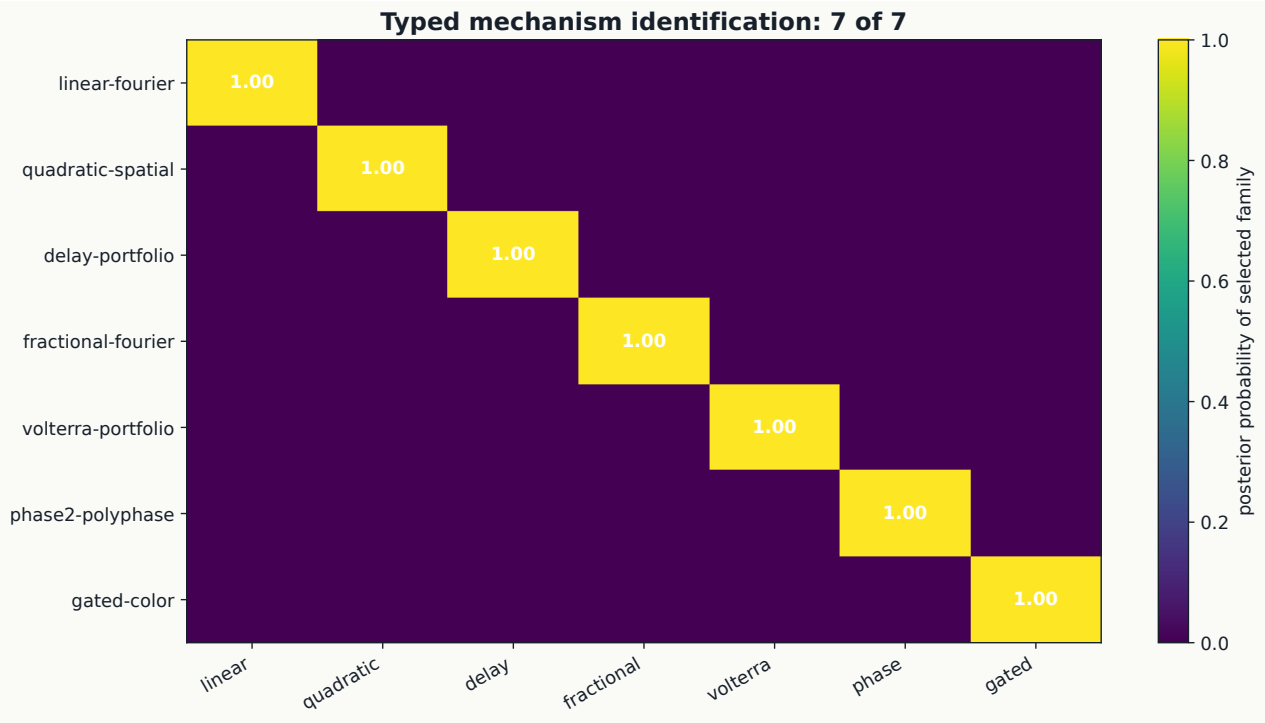


Figure 20: Every controlled mechanism was identified by its typed posterior.

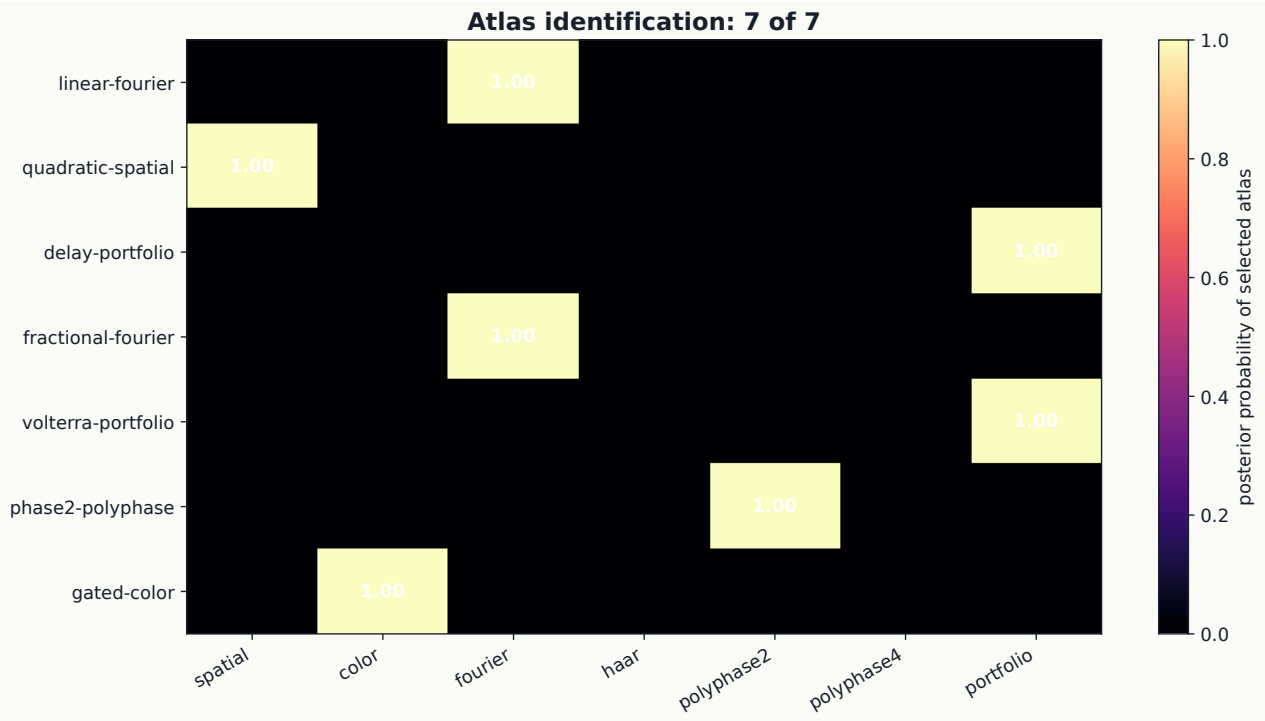


Figure 21: Every controlled coordinate atlas was identified.

14.1 Linear ablation

The baseline is one linear macro-resolution program using the portfolio atlas. PR-LASM uses the full typed posterior. The test-state RMSE values are:

system	linear baseline	PR-LASM	ratio
linear Fourier	0.2449	0.0221	0.0903
quadratic spatial	0.2708	0.1240	0.4580
delay portfolio	0.3112	0.0629	0.2020
fractional Fourier	0.2735	0.0654	0.2393
Volterra portfolio	0.2793	0.0642	0.2300
phase-two polyphase	0.1465	0.0436	0.2978
gated color	0.0502	0.0385	0.7674

Table 1: Held-out state-prediction RMSE. The benchmark is controlled and mechanism-specific.

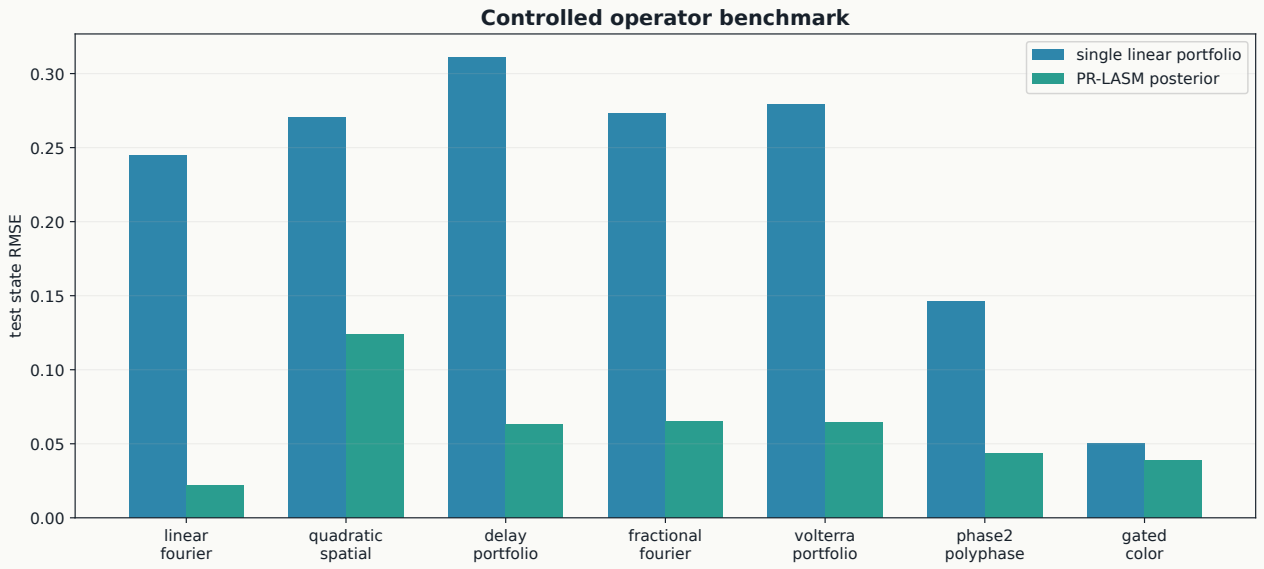


Figure 22: The full typed model versus the single linear portfolio baseline.

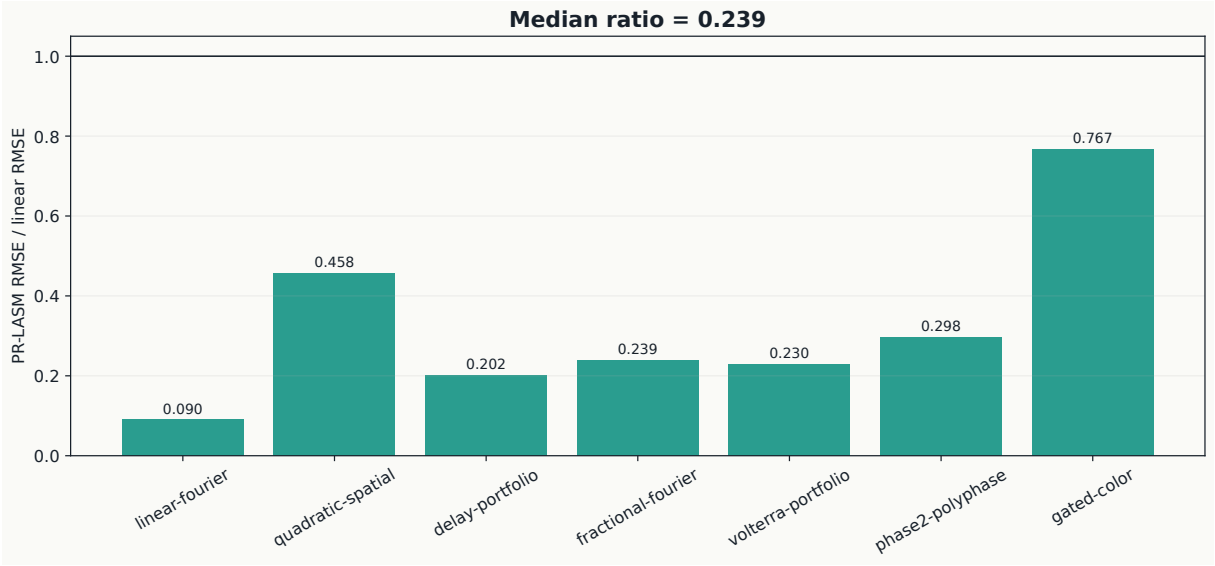


Figure 23: All seven state RMSE ratios are below one; the median is 0.2393.

Boundary 14.2 (Interpretation of the controlled benchmark). The result verifies program construction, posterior comparison, and held-out prediction when the generating family belongs to the supplied grammar. It does not prove that the grammar contains the mechanism of an arbitrary environment.

15 Deterministic interactive benchmarks

Four source-blind environments test the integrated planning loop:

- a five-step action sequence;
- a spatial click target;
- a grid-navigation task with walls;
- a six-step alternating phase switch.

All expose only observations and public actions. Deterministic replay is available, so the graph and probe system can be tested exactly.

Finite certificate 15.1 (Interactive systems). PR-LASM solved all four systems. The recorded committed-action counts were 38, 1, 29, and 14. The corresponding graph sizes ranged from one to twenty states and from twelve to thirty-two transition edges.

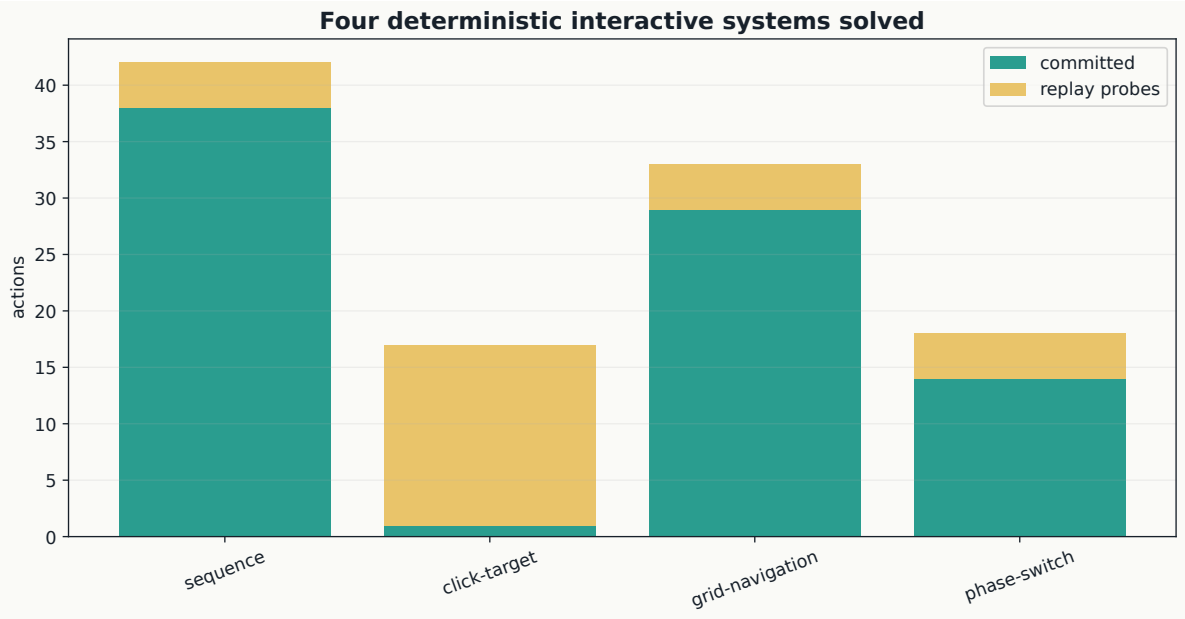


Figure 24: Committed actions and replay probes in the four interactive systems.

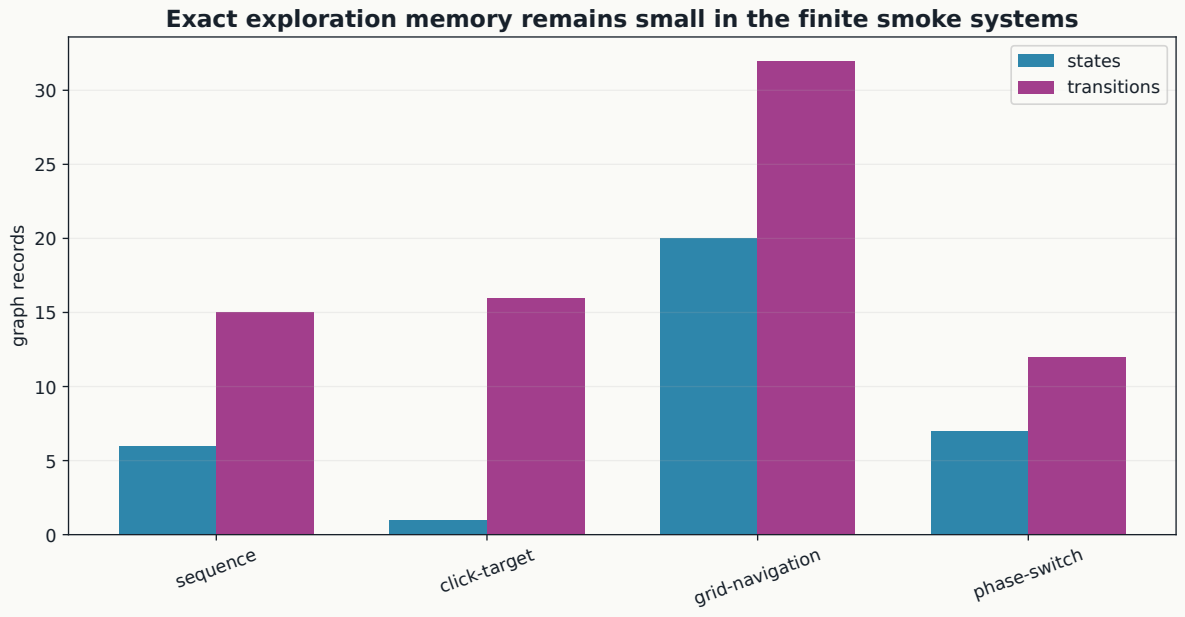


Figure 25: Exact graph memory remained compact in the finite smoke tasks.

16 The supplied twenty-five-environment suite

The release bundles twenty-five local environment trees. Static validation checks metadata, source presence, source syntax, the expected game-class structure, and prohibited network or process-control imports. All twenty-five trees pass.

The supplied baseline-action totals vary substantially, so the runner retains per-environment budgets rather than using one small global cap.

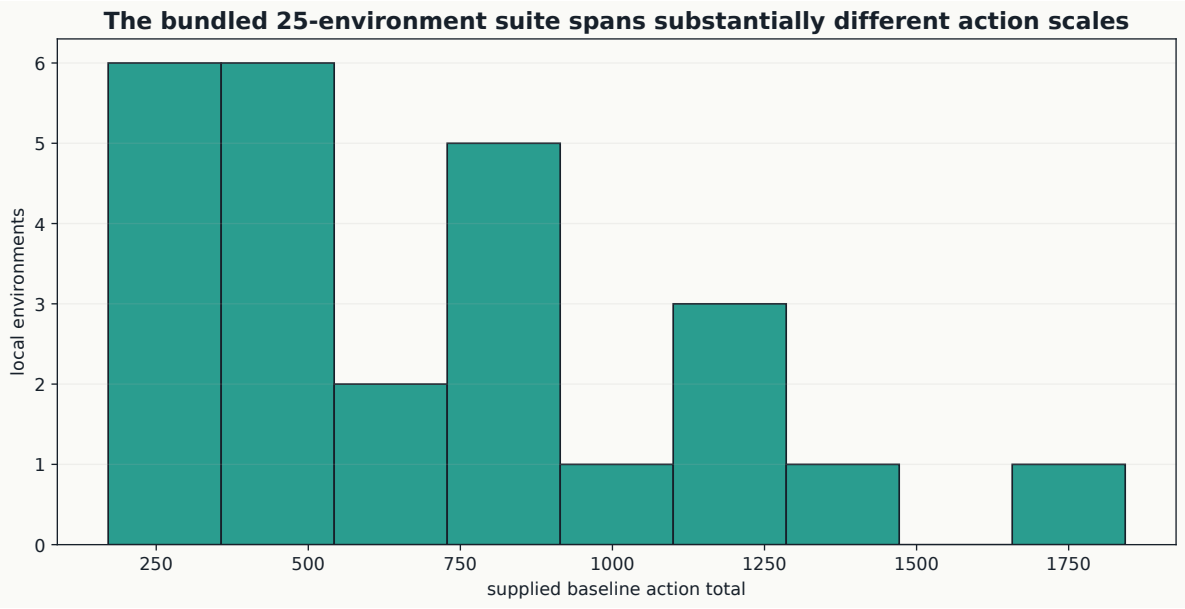


Figure 26: Distribution of supplied baseline-action totals across the local suite.

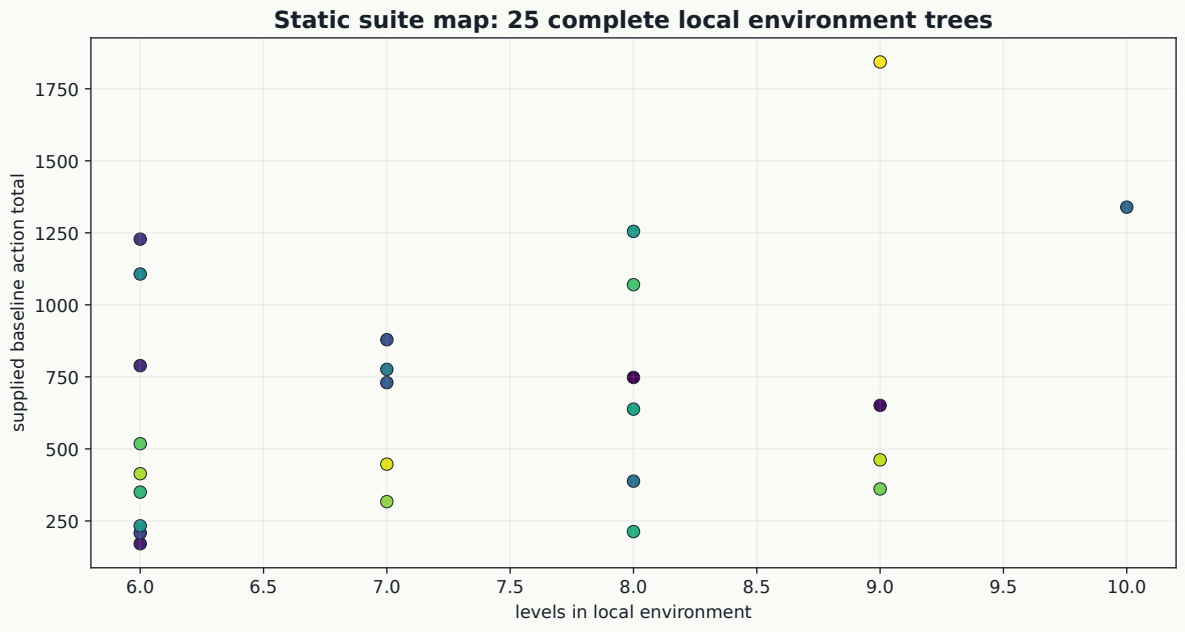


Figure 27: Level counts and baseline-action totals for the twenty-five local environment trees.

Boundary 16.1 (No official execution claim). The official local runtime was not available in the present build container. Therefore this paper reports no solved count, score, or timing for the twenty-five environments. The release includes offline run scripts for the target machine and preserves the original strict network guard.

17 Software architecture

module	role
`exact_atlas.py`	unitary Fourier, orthonormal Haar, polyphase, categorical color, projector queries, tight-frame synthesis
`lasm_packetizer.py`	exact archive, named feature groups, deterministic nested projections
`lasm_refinement.py`	complete packet trees, mixed-depth synthesis, best-basis dynamic programming
`lasm_programs.py`	typed design maps for linear, nonlinear, phase, delay, and memory mechanisms
`lasm_model.py`	Bayesian fitting, posterior weights, uncertainty decomposition, action prediction
`lasm_memory.py`	committed states, lag packets, power memory, exponential memory, role statistics
`lasm_graph.py`	exact transition edges, nondeterminism marking, frontier and path queries
`lasm_agent.py`	replay probes, utility scoring, graph updates, source-blind interactive loop
`lasm_benchmark.py`	exactness, controlled operator, and interactive benchmarks
`lasm_cli.py`	benchmark, smoke, single-environment, and suite commands

The command

```
python -m komorebi_arc3_offline.lasm_cli benchmark \
  --output results/pr_lasm_benchmark
```

regenerates the benchmark ledgers. The PowerShell wrappers provide the same operation on Windows.

18 Verification

Finite certificate 18.1 (Automated suite). The complete source release passes

$$\boxed{36/36 \text{ automated tests.}}$$

The tests include all seventeen inherited offline-runner tests and nineteen PR-LASM tests covering exact atlas reconstruction, Fourier packet completeness, polyphase synthesis, wavelet subband synthesis, tight-frame reconstruction, deterministic packetization, mixed-depth best-basis reconstruction, graph frontiers, seven typed program-identification cases, and four interactive systems.

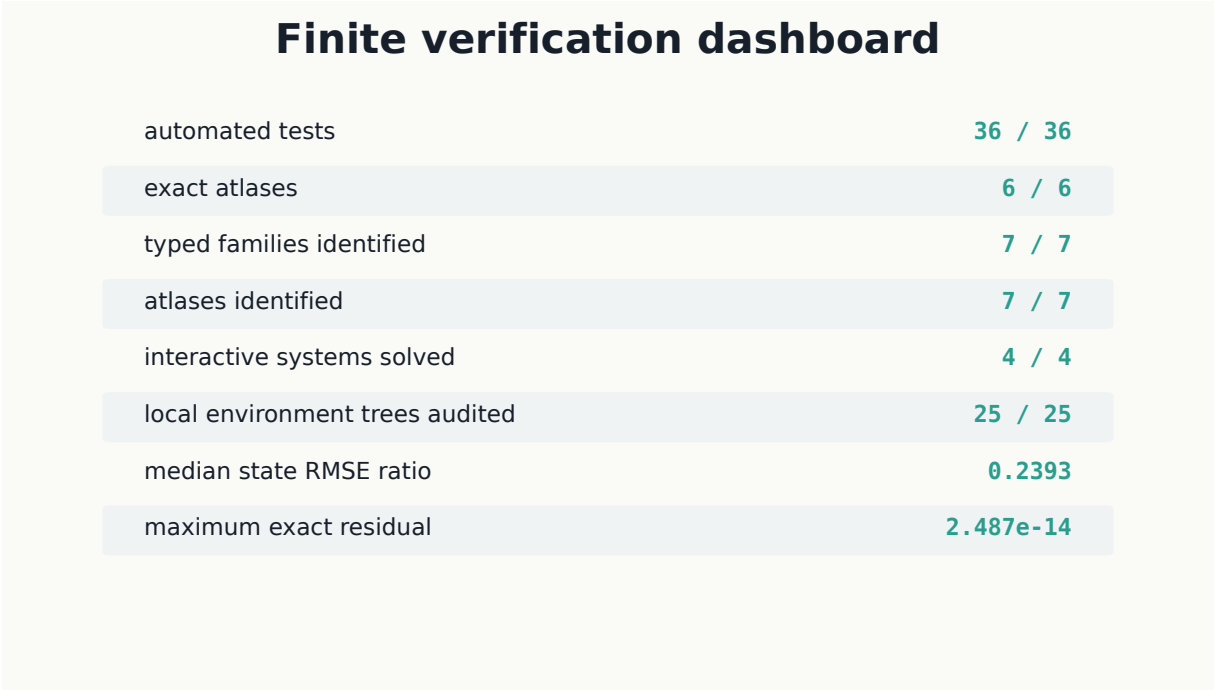


Figure 28: Summary of the finite verification state.

19 Claim ledger

statement	status	boundary
Per-atlas packet reconstruction	theorem	finite unitary atlas and complete coordinate selectors
Weighted multi-atlas synthesis	theorem	finite family of unitary atlases with positive weights
Categorical color reconstruction	exact construction	finite one-hot palette
Mixed-depth packet pruning	theorem	finite complete binary packet tree
Best-basis dynamic program	theorem	finite additive node cost
Bayesian regression formulas	exact finite formulas	declared normal-inverse-gamma model
Typed program selection	finite certificate	seven controlled systems generated from the supplied grammar
RMSE improvement	finite certificate	controlled held-out benchmark only
State-action graph	exact finite data structure	deterministic edges only
Interactive solutions	finite certificate	four bundled deterministic systems
Twenty-five environment integrity	static certificate	source trees, not executed games
Official environment score	not claimed	official local runtime absent from build container
Universal operator discovery	not claimed	finite declared grammar
AGI	not claimed	outside the mathematical and empirical scope

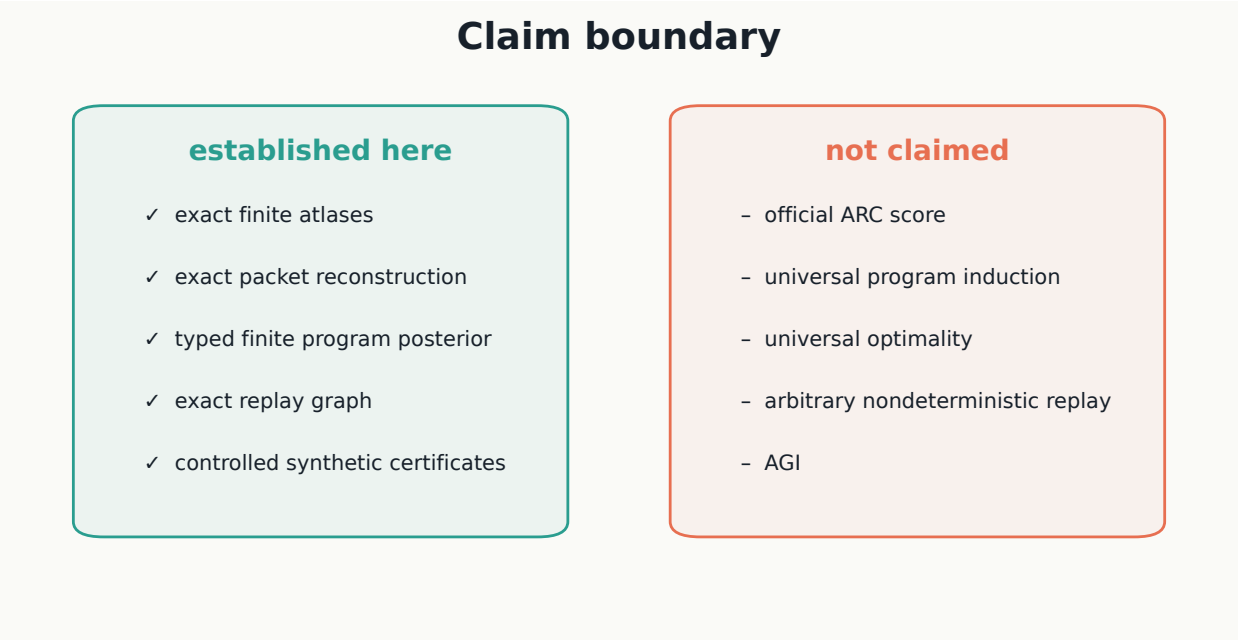


Figure 29: The proof and benchmark boundary is part of the model specification.

20 What has been completed

The construction now contains every layer required for one coherent large signal model:

1. exact multi-atlas observation coordinates;
2. Fourier, Haar-wavelet, polyphase, spatial, and categorical realizations;
3. a normalized redundant reconstruction theorem;
4. named feature groups and nested statistical states;
5. exact adaptive packet-tree refinement;
6. a typed nonlinear and memory-bearing operator grammar;
7. Bayesian evidence and posterior uncertainty;
8. exact transition-graph memory;
9. replay-safe active planning;
10. synthetic mechanism-identification and interactive certificates;
11. a source-blind offline adapter and twenty-five local environment trees;
12. a reproducible package, test suite, figures, tables, and run scripts.



Figure 30: The completed model closes exact analysis, learning, exploration, planning, and reconstruction.

21 Remaining research directions

The model is finished as a finite architecture, but several extensions are mathematically substantive.

21.1 Program grammar growth

A future compiler should be able to propose new typed operator programs while preserving an explicit description length and proof-status boundary. Arbitrary feature generation would destroy the meaning of the posterior over mechanisms.

21.2 Learned invertible atlases

The present exact atlases are fixed. One may add learned transforms only when their inverse, conditioning, and packet selectors remain explicit. A learned bijection without stable synthesis is not an exact atlas in the present sense.

21.3 Continuous and stochastic worlds

The finite graph can be replaced by kernels or stochastic transition operators. Exact replay edges then become probability distributions, and frontier planning requires uncertainty over both the world and the graph itself.

21.4 Scaling and hardware

Packet-sparse expert fitting, batched small regressions, structured transforms, and graph-local updates are natural GPU targets. The present CPU certificates do not establish an optimal hardware design.

21.5 Empirical evaluation

The next empirical step is straightforward: install the official local runtime on the target machine, execute the bundled suite, preserve complete traces, and compare PR-LASM against the earlier Bayesian operator runner and graph-only ablations under identical budgets. That is an experiment, not another missing architectural component.

22 Conclusion

PR-LASM treats a signal world as more than a stream of vectors. An observation has several exact coordinate descriptions; an operator has packet transfer blocks; nonlinear behavior has interaction features; memory has typed kernels; uncertainty lives over mechanisms and representations; exploration creates an exact graph where possible; and every compression remains attached to an exact reconstruction archive.

The result is not a language model disguised as a signal model, and it is not a dense predictor decorated with wavelets after the fact. It is an algebraic production system:

signal object → exact atlas portfolio → packet features and nested states → typed operator posterior

→ memory and transition graph → active action → residual → selective refinement → exact reconstruction

Within its finite declared scope, that loop is complete.

A Exact transform formulas

A.1 Haar analysis and synthesis

For one pair,

$$\ell = \frac{a+b}{\sqrt{2}}, \qquad h = \frac{a-b}{\sqrt{2}},$$

and

$$a = \frac{\ell+h}{\sqrt{2}}, \qquad b = \frac{\ell-h}{\sqrt{2}}.$$

Applying these identities separably and recursively proves the finite two-dimensional transform used in the code.

A.2 Polyphase analysis and synthesis

For modulus N,

$$(\mathcal{A}_N x)_{r,s}[q,t] = x[Nq+r, Nt+s].$$

The synthesis operator places each channel value back at its unique original address. Therefore

$$\mathcal{S}_N \mathcal{A}_N = I.$$

A.3 Fourier selector projectors

For any diagonal zero-one mask E_B,

$$P_B = F^* E_B F,$$

so

$$P_B^2 = P_B, \qquad P_B^* = P_B.$$

Disjoint masks annihilate, and a complete mask partition sums to the identity.

B Bayesian regression formulas

For one output y and design matrix X, take

$$\beta \mid \sigma^2 \sim N(0, \sigma^2 \Lambda_0^{-1}), \qquad \sigma^2 \sim \mathrm{InvGamma}(a_0, b_0).$$

Then

$$\Lambda_n = \Lambda_0 + X^\top X, \qquad \mu_n = \Lambda_n^{-1} X^\top y,$$

$$a_n = a_0 + \frac{n}{2}, \qquad b_n = b_0 + \frac{1}{2}(y^\top y - \mu_n^\top \Lambda_n \mu_n).$$

For a feature row x_*, the predictive mean is $x_*^\top \mu_n$ and the Student predictive scale contains

$$1 + x_*^\top \Lambda_n^{-1} x_*.$$

The implementation applies this independently to every output coordinate while sharing the design covariance.

C Reproducibility commands

C.1 Linux or macOS shell

```
cd Komorebi_PR_LASM_Ultimate
export PYTHONPATH="$PWD/src"
pytest -q
python -m komorebi_arc3_offline.lasm_cli benchmark \
  --output results/pr_lasm_benchmark
```

C.2 PowerShell

```
$Work = "$env:USERPROFILE\Downloads\Komorebi_PR_LASM_Ultimate"
Set-Location $Work
$env:PYTHONPATH = "$Work\src"
python -m pytest -q
& "$Work\Run_PR_LASM_Benchmark.ps1"
& "$Work\Run_PR_LASM_Smoke.ps1"
```

C.3 Local twenty-five-environment run

```
& "$Work\Run_PR_LASM_5_Easiest.ps1" -EnvName "arc3-offline"
& "$Work\Run_PR_LASM_All_25.ps1" -EnvName "arc3-offline"
```

D Machine-readable result files

The benchmark directory contains:

- `exactness_benchmark.json`;
- `operator_benchmark.json` and `operator_benchmark.csv`;
- `agent_benchmark.json` and `agent_benchmark.csv`;
- `benchmark_summary.json`;
- per-environment traces, posterior traces, graph summaries, and result files.

E Test map

test group	content
Inherited offline runner	actions, packetizer, Bayesian model, replay, offline guard, official-adapter contract, CLI, suite discovery, suite bundle
Exact atlases	spatial, Fourier, Haar, polyphase, color, weighted tight-frame synthesis
Packet projectors	Fourier band completeness, polyphase completeness, wavelet subband completeness

test group	content
Nested packetizer	determinism, dimensions, reconstruction ledger
Adaptive basis	mixed-depth selection and exact synthesis
Transition graph	frontier paths and winning edges
Program posterior	seven parametrized mechanism and atlas identifications
Interactive agent	sequence, click, navigation, and phase-switch environments