SCALING GRAPH TOPOLOGY AUGMENTATION WITH STRUCTURED VARIATIONS AND SPECTRAL TRUST IN NON-STATIONARY REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Topology Augmentation (GTA) improves prioritized planning under reward change. Yet the current compute model hinder scalability, with critical bottlenecks in CPU residence, exact matrix operator representation and obnoxious eigensolve. Therefore, we introduce GTA^{+}-PS, which stages changed transition rows as rank-r shocks and publishes consistent belief state through coherent commits. Moreover, a trace-derived spectral trust certificate replaces periodic second-largest-eigenvalue-modulus estimation. We derive exact Woodbury updates for resolvent actions, connect the trace to discounted signal variance, and prove convergence under deterministic or variance-scaled stochastic selection. Concurrently, we map all parallelizable operations onto GPU-resident computation model, coupling with an efficient implementation that enables streamlined inference-learning. We evaluate the new design on an extensive suite of varying environments, representation, host model, and learn update. Our augmented methods gain persistence performance over their standard baseline, in both tabular and visual-control setting.

1 INTRODUCTION

Model-based reinforcement learning improves sample efficiency by reusing a learned model for simulated updates. When the factual information changes: reward, policy, or even the model itself, planning becomes a non-stationary resource-allocation problem. A limited backup budget must track both the size of each local error and its reach through the new dynamics.

1.1 PLANNING UNDER DRIFT

Prioritized sweeping focuses computation on states with larger estimate residuals and propagates their influence through model predecessors (Moore & Atkeson, 1993; Peng & Williams, 1993). Dyna places this mechanism inside a broader loop that interleaves acting, learning, and simulated updates (Sutton, 1990). Yet a one-hop residual says little about how far a change can travel through the current transition graph. Graph topology augmentation (Pham & Dam, 2026) addresses this gap by diffusing residual evidence through a policy-induced resolvent and aggregate the bonus into the priority.

And when the environment drifts, the topology must change with the model. New reward changes the right-hand side of planning, whereas state transition update tweaks the operator itself. Treating these events alike either delays useful information or rebuilds expensive topology objects without need.

1.2 SCALING BARRIER

The topology operator is a regularized resolvent of a directed Markov kernel. It aggregates paths of all lengths, much like the *Successor Representation* (Dayan, 1993), and thus supplies global context to a local residual via a right-hand-side multiplication. Direct recomputation costs cubic time for a dense n-state model. Repeated eigensolver calls add another quadratic pass when a spectral statistic controls the strength of topology guidance. Both costs are misaligned with local model changes, which often alter only a small set of transition rows.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

However, exact row updates alone *does not* make incremental maintenance efficacious. The operator, its statistic, and the queue must describe the same model version. New beliefs often comes exogenously and asynchronously, which may trigger skewed operator update. Moreover, finite-precision corrections eventually accumulate to cease a reliable representation, especially for the inverse.

A scalable planner therefore needs an exact update identity, an atomic publication boundary, and a numerical certificate with an admitted representation-capacity policy that decides when to rebuild.

1.3 Contributions

We introduce GTA$^+$-PS, an extension of the GTA initiative, with refinement over topology-maintenance and update-scheduling layer for prioritized model-based planning. The improvement comes with a general enhancement of the incremental update model, which both account for sparse and dense representation program. The logical contributions are threefold.

- We factor a buffered batch of changed transition rows, called a **Belief Batch** as a rank-r shock and derive exact Woodbury updates for resolvent actions, the topology signal, and the maintained trace. The design is rooted in the *Structured Variation* model of (Dam, 2025).
- We replace the SLEM eigensolver of (Pham & Dam, 2026) with a trace-derived statistic, called **Spectral Trust**, that controls deterministic topology trust and an isotropic Gumbel jitter. For episodic models, the modified **Terminal Completion** account for absorbing end state. This extends the canonical scalar reduction to sub-stochastic planning kernels while preserving the resolvent semantic and the host's Bellman backup.
- We define the basic commit unit of the GTA$^+$-PS module, called **Coherent Commits**, that publish operator, trust, and queue state under one contract. The computation model are curated revolving around that contract, with care about representation density, drift accumulation, acceleration work, and concurrent rebase.

We evaluate GTA$^+$-PS at two graded scales against established planning and control baselines.

- Stage I studies adaptation and throughput on sparse tabular environment, consisting up to 4500 numerical states.
- Stage II studies visual control on Pong (Bellemare et al., 2013) and ViZDoom My Way Home (Kempka et al., 2016), using learned abstractions and accelerator-aware packing.

Section 6 illustrates how the models are competing, and how the GTA$^+$-PS mechanisms improve planning quality and throughput under matched budgets.

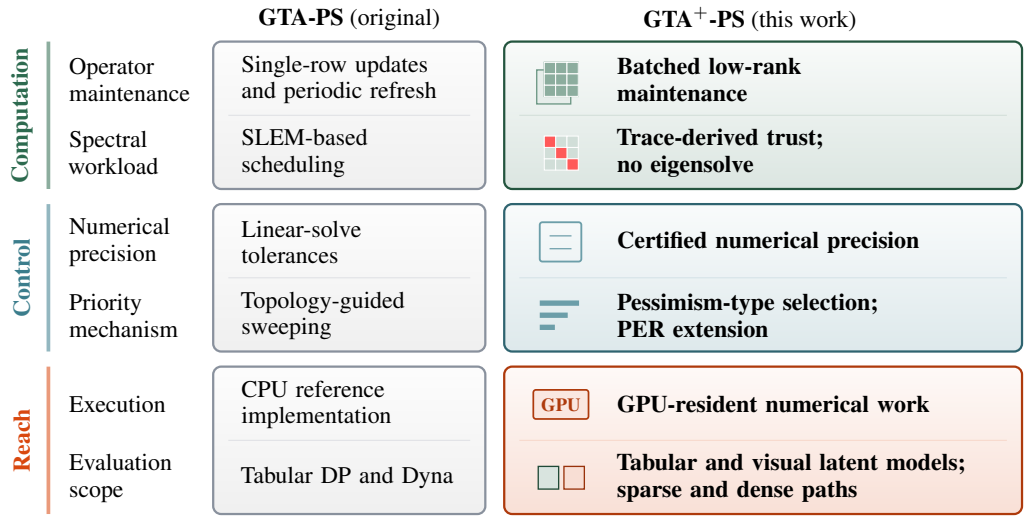

Figure 1: Qualitative comparison with original GTA-PS. The cards summarize computation, control, and evaluation scope.

2 BACKGROUND

Our method, as a successor of GTA-PS, combines two established views of planning. Prioritized Sweeping chooses where to apply a backup, facilitated by spectral analysis on the Successor Representation, which helps surgically reveal how information moves between states. We introduce the background needed to connect those views.

2.1 PRIORITIZE SWEEPING

Let $\mathcal{T}$ denote the host backup map—here, a Bellman operator—and V the current value estimate. **Prioritized Sweeping** (PS) inserts states into a queue using the magnitude of the Bellman residual $|\mathcal{T}V - V|$, then updates high-priority states first (Moore & Atkeson, 1993). The one-step discrepancy poses urgency of value-function mismatch, thus directs fixing effort into more important states. More broadly, update order can materially change the time needed by an MDP solver to reach a given error (Wingate & Seppi, 2005).

GTA-PS (Pham & Dam, 2026) proposed an unified interface for prioritization modules, in both regular Value Iteration as well as Dyna-style planning. Here, the planner is aggregated with a *priority queue*, representing the desired effort strategy (not necessarily the data structure itself). The planner receives inference data, and the priority queue sort these units into a stream-ordering. The in-charge learning method, which is referred as *the Host*, queried the *top* of the queue for an optimization step, whatever DP or TD-learning. The new policy yield a new planner's reception, hence refreshing the queue's inner scoring.

In short, the priority playground is separated, and be orthogonal, of any other agent's facility. The module keeps some representation of the MDP's topology, receives model/reward deltas, keeping the score value consistent.

2.2 DIFFUSION GEOMETRY OF MDP

A Markov Decision Process can be seen as a state-wise directed graph, with its Markov-chain kernel being row-stochastic, given the policy is fixed. Repeated application and resolvent filters on such kernel simulate signal propagation in the topology (Sandryhaila & Moura, 2013; Shuman et al., 2013). For an exogenous signal ξ, the resolvent expansion $\sum_{h\geq 0}(1+\eta)^{-(h+1)}P^h\xi$ assigns weight to every reachable walk. Consequently, its value is large where present error can persist or spread. The resulting magnitude complements the local geometry.

The **Laplacian** of that kernel, which envelope the semantic of signal diffusion, is more conveniently employed in analysis. Spectral surgery of the Laplaciannturns local transition structure into global geometric features, revealing connectivity, bottlenecks, and diffusion patterns (Chung, 2005). These features provide coordinates adapted to the state space. In reinforcement learning, eigenvectors of suitable symmetric Laplacians define proto-value functions, supporting compact representations, value-function approximation, and policy learning (Mahadevan & Maggioni, 2007).

2.3 RELATED WORK

The **Successor Representations** encode discounted state occupancy under the policy-induced kernel (Dayan, 1993). Spectral and geometric analysis has been studied for such operator to extract useful features (Mahadevan & Maggioni, 2007; Machado et al., 2017).

Nonstationary MDP analyses measure reward and transition variation under exogenous environment drifts (Cheung et al., 2020), and recent models exploit low-rank transition structure (Cheng et al., 2023; Dam, 2025). The rank assumption often concerns each observed update — only a small number of rows need change between commits even if the full kernel has high rank.

Finally, inverse-maintenance identities date to rank-one and block corrections (Sherman & Morrison, 1950; Woodbury, 1950). The spectral variance identity is inspired from (Sobel, 1982). We adopt them to a row-stochastic directed kernel, couple the correction to queue publication, and maintain the trace needed for trust. These additions turn an algebraic update into a successive planning protocol.

3 Problem Formulation

We seek a priority rule that reacts to local model changes without totally rebuilding global graph operator. The rule must also remain well-defined when planning, model publication, and physical execution advance at different rates. We now fix these objects and their clocks.

3.1 Dynamic planning model

Consider a finite discounted model at Operator Version v, $\mathcal{M}_v = (\mathcal{S}, \mathcal{A}, p_v, r_v, \gamma)$, where $n = |\mathcal{S}|$, $m = |\mathcal{A}|$ and $\gamma \in (0, 1)$. A smoothed topology policy π_v^{GTA} induces the directed row-stochastic kernel

$$P_v = \mathcal{P}(\pi_v^{\mathrm{GTA}}, p_v) = \left[\sum_{a \in \mathcal{A}} \pi_v^{\mathrm{GTA}}(a|s) p_v(s'|s, a)\right]_{s,s'} \qquad P_v \mathbf{1} = \mathbf{1}. \tag{1}$$

Here P_v is a derived mathematical view, rather than an independently maintained state object. The policy owner normalizes π_v^{GTA}, and the model owner normalizes p_v. The numerical backend realizes the outgoing and incoming systems induced by their joint state.

The smoothing is a finite-temperature Boltzmann map of the host action scores, $\pi_v^{\mathrm{GTA}}(a \mid s) \propto \exp(A_v(s, a)/\lambda_\pi)$, with policy temperature λ_π. This convention, though artificial, prevents a partially updated policy from defining the active graph.

For $\eta > 0$, define the outgoing resolvent

$$G_{\eta,v}^{\mathrm{out}} = G_{\eta,v} = \big((1 + \eta)I - P_v\big)^{-1}. \tag{2}$$

It exists because $\rho(P_v) = 1 < 1 + \eta$. The incoming resolvent is defined in the same way as

$$G_{\eta,v}^{\mathrm{in}} = \big(\mathrm{diag}(P_v^\top \mathbf{1}) - P_v^\top + \eta I\big)^{-1}. \tag{3}$$

Its defining matrix is strictly row-diagonally dominant with margin η, so it is also nonsingular. We write $G_v = G_{\eta,v}^{\mathrm{out}}$ in the outgoing update and retain the superscript for the incoming resolvent.

The host supplies the entrywise absolute residual $\xi_v = |\mathcal{T}_v V_v - V_v|$. The topology signal is

$$\tau_v = (1 - \alpha) G_{\eta,v}^{\mathrm{out}} \xi_v + \alpha G_{\eta,v}^{\mathrm{in}} \xi_v, \qquad 0 \le \alpha \le 1. \tag{4}$$

Thus ξ_v measures local Bellman error, while τ_v combines forward propagation and backward attribution through the active model. Successor diffusion is the special case when $\alpha = 0$.

The inherited GTA-PS rule combines these signals into the joint priority

$$p_v^{\mathrm{GTA}}(s) = \xi_v(s) + \beta_v \tau_v(s), \qquad \beta_v \ge 0. \tag{5}$$

Here β_v controls the contribution of topology relative to local Bellman error. The revised scheduler retains this deterministic score and specifies its adaptive weight and queue perturbation below.

3.2 Structured variation

Suppose one belief batch changes the rows indexed by $R_v = \{i_1, \ldots, i_r\}$. Let $U_v = [e_{i_1}, \ldots, e_{i_r}] \in \mathbb{R}^{n \times r}$ and collect the row differences in $V_v^\top \in \mathbb{R}^{r \times n}$. Then

$$P_{v+1} = P_v + U_v V_v^\top, \qquad \mathrm{rank}(U_v V_v^\top) \le r, \qquad V_v^\top \mathbf{1} = 0. \tag{6}$$

The last equality preserves row stochasticity. This factorization is exact for any batch of changed rows; it generally does not assume that the full transition kernel is low rank. The rationale of this model is discussed in (Dam, 2025). We target the regime $r \ll n$, where an incremental commit is more proper than a fresh factorization.

4 GTA$^+$-PS

GTA$^+$-PS turns each belief batch into one versioned priority state. It first updates the resolvent representation, then derives trust and priorities from that candidate, and finally publishes the state atomically. The host continues to execute every Bellman backup.

4.1 RANK-r MAINTENANCE

Write $G_v = G_{\eta,v}$ for brevity and define

$$Z_v = G_v U_v, \qquad C_v = (I_r - V_v^\top Z_v)^{-1}. \tag{7}$$

If the $r \times r$ core is nonsingular, the Woodbury identity gives

$$G_{v+1} = G_v + Z_v C_v V_v^\top G_v. \tag{8}$$

The update acts on blocks without requiring an explicit policy-induced kernel. After numerical maintenance and any scheduled certificate, the diffusion joint applies the current right-hand side $H_{v+1} = H_v + \Delta H_v$:

$$Y_{v+1} = G_{v+1} H_{v+1}. \tag{9}$$

An RHS-only refresh reuses the maintained G_v and leaves Operator Version v and the certificate schedule unchanged.

4.2 PRIORITY RULE

Trace scale. Define the maintained resolvent trace as $F_v = \mathrm{tr}(G_v)$. With $W_v = G_v^\top V_v$, the same low-rank blocks and cyclic invariance yield

$$F_{v+1} = F_v + \mathrm{tr}(C_v W_v^\top Z_v). \tag{10}$$

We remove the stationary eigenmode and normalize the remaining mass through

$$\Psi_{\eta,v} = \frac{(1+\eta)F_v - n - 1/\eta}{n-1}, \qquad \mathcal{U}_{\eta,v} = \sqrt{\frac{[1 + 2\Psi_{\eta,v}]_+}{1 - (1+\eta)^{-2}}}. \tag{11}$$

Section 5 connects $\mathcal{U}_{\eta,v}$ to discounted signal variance. Here it serves as a dimensionless trust measure. Slow mixing or persistent directed modes increase $\mathcal{U}_{\eta,v}$ and reduce reliance on topology diffusion.

Episodic planning kernels can be substochastic because a terminal outcome has no ordinary successor. Appendix C.3 defines a trust-only Terminal Completion that restores the row-stochastic trace reduction while leaving the host's terminal Bellman update unchanged.

Priority score. Using the residual and directional mixture from Equation (4), replace the inherited SLEM penalty by the trace-derived scale ν_v:

$$\nu_v = \sqrt{\lambda_\omega}\,\mathcal{U}_{\eta,v}, \qquad \beta_v = \beta_1[1 + \beta_2 - \nu_v]_+, \qquad d_v(s) = \xi_v(s) + \beta_v \tau_v(s). \tag{12}$$

Our anti-starvation mechanism adds i.i.d. jitter of equal variance across eligible states, giving an isotropic covariance perturbation. Spectral Trust sets its common amplitude, coupling exploration to topology-induced variability. We implement this construction with the Gumbel-max trick (Kool et al., 2019), drawing one canonical perturbation per candidate in the frozen set $\mathcal{C}_v$:

$$T_v = \sqrt{n-1}\,\mathcal{U}_{\eta,v}, \qquad g_v(s) \overset{\text{iid}}{\sim} \mathrm{Gumbel}(0,1). \tag{13}$$

Since $\tau_v(s) \geq 0$, the concrete GTA$^+$-PS priority is

$$\begin{aligned} p_v(s) &= d_v(s) + T_v g_v(s) \\ &= \xi_v(s) + [\beta_1(1+\beta_2)\tau_v(s) - \beta_1 \tau_v(s)\nu_v]_+ + T_v g_v(s). \end{aligned} \tag{14}$$

The penalty $\beta_1 \tau_v(s)\nu_v$ reduces each state's nominal topology contribution, while T_v sets the breadth of stochastic queue selection.

For $T_v > 0$, the Gumbel-max identity reveals that this perturb-and-select rule induces canonical softmax sampling (Appendix C.5):

$$\Pr\left(\arg\max_{s \in \mathcal{C}_v} p_v(s) = i \mid d_v, T_v, \mathcal{C}_v\right) = \frac{\exp(d_v(i)/T_v)}{\sum_{j \in \mathcal{C}_v} \exp(d_v(j)/T_v)}. \tag{15}$$

Taking the top-k entries of the same perturbed field samples without replacement. Thus Spectral Trust controls both the topology penalty and the breadth of exploration. Theorem 1 establishes almost-sure anti-starvation under its bounded normalized-score and recurring-eligibility conditions. Zero temperature recovers deterministic selection.

Lower Confidence Bound semantic. Under the assumptions of Proposition 2, Spectral Trust measures the normalized standard-deviation width of discounted topology propagation. Equation (14) application of the topology bonus can be seen as an LCB-style form: nominal diffusion credit minus a width-scaled penalty, clipped at zero. Adding this conservative bonus to the local residual yields a pessimistic arm-selection score, while Gumbel perturbations retain exploration.

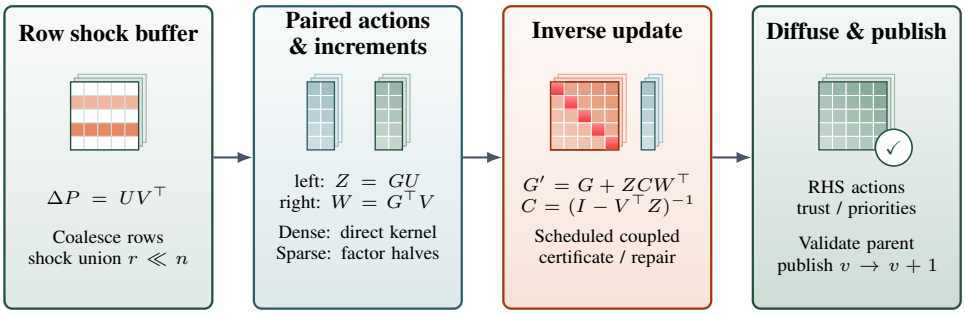

Figure 2: Effective Coherent Commit model. Coalesced rows produce paired actions and updated inverses. Scheduled coupled checks precede diffusion and atomic publication.

4.3 Operation Models

Both execution paths realize and incrementally update the outgoing and incoming systems. The dense path also stores their inverse resolvents as full matrices. The sparse path represents inverse actions through sparse factors and bounded low-rank corrections. We use it for thin tabular transitions, while condensed neural representations use the dense path. Appendix E.7 gives more grained operation-level assurances and structural comparison.

4.4 Precision Certificate

One Inverse Certificate checks the completed outgoing and incoming resolvents against both primitive-induced and resident systems. Its unscaled maximum row-sum residual uses the threshold $\epsilon_{\text{inv}} = 5 \times 10^{-4}$ at scheduled Operator Versions. An exceeded threshold requests internal Numerical Rebase; rank growth governs physical corrections. Appendix D.2 specifies the bounds and cadence.

5 Analysis

We establish three properties of the design. First, every accepted commit is **algebraically exact**. Next, the spectral trace controls the variance scale used by the successive topology. Finally, positive repeated selection recovers standard asynchronous value convergence after drift stops.

5.1 Exact Operator Updates

Proposition 1 (Commit exactness)**.** *Suppose $G_v = ((1+\eta)I - P_v)^{-1}$ and $I_r - V_v^\top G_v U_v$ is non-singular. Then Equations (8)–(10) equal the resolvent, its action on any right-hand side, and its trace for $P_{v+1} = P_v + U_v V_v^\top$.*

Proof. Set $A_{\eta,v} = (1+\eta)I - P_v$, so $A_{\eta,v+1} = A_{\eta,v} - U_v V_v^\top$. Applying the block inverse identity to $A_{\eta,v+1}^{-1}$ proves Equation (8). Right multiplication gives Equation (9), and cyclic invariance of the trace gives Equation (10). Repeating the argument inductively proves exactness across commits. □

Thus the Woodbury core generalized the rank-1 Sherman-Morrison update of (Pham & Dam, 2026). The nonsingularity check also has a direct meaning. It rejects a correction whose proposed core cannot represent the next inverse from the current factors.

5.2 TRUST CERTIFICATE

For the determinacy of the signal's discounted variance sum, the MDP itself requires stationarity. Under this assumption, the stationary distribution $\mu_v = \mu(p, \pi, v)$ exists as the convergence of the successor occupancy.

Let g be centered signal in $L_2(\mu_v)$. Its discounted sum $Y_g = \sum_{h\geq 0}(1+\eta)^{-h} g(X_h)$ satisfies:

$$\mathrm{Var}(Y_g) = \frac{1}{1-(1+\eta)^{-2}} \left(\|g\|^2_{2,\mu_v} + 2\sum_{h\geq 1}(1+\eta)^{-h}\langle g, P_v^h g\rangle_{\mu_v} \right), \tag{16}$$

which is the covariance expansion for a discounted Markov reward process (Sobel, 1982).

Under the full-support assumption below, let $D_{\mu_v} = \mathrm{diag}(\mu_v)$, let Q_v contain a μ_v-orthonormal basis of the centered subspace, and write $\kappa = (1+\eta)^{-1}$. Isotropy and the Frobenius identity $\langle A, B\rangle_F = \mathrm{tr}(A^\top B)$ reduce each lag and the full covariance series to

$$\begin{aligned} \mathbb{E}_g\langle g, P_v^h g\rangle_{\mu_v} &= \sigma^2_{\xi,v}\left\langle D_{\mu_v}^{1/2}Q_v, D_{\mu_v}^{1/2}P_v^h Q_v\right\rangle_F = \sigma^2_{\xi,v}\big(\mathrm{tr}(P_v^h) - 1\big), \\ \sum_{h\geq 1}\frac{1}{(1+\eta)^h}\mathbb{E}_g\langle g, P_v^h g\rangle_{\mu_v} &= \sigma^2_{\xi,v}\big((1+\eta)F_v - n - \eta^{-1}\big) = (n-1)\sigma^2_{\xi,v}\Psi_{\eta,v}. \end{aligned} \tag{17}$$

The first line removes the stationary direction; the second resums all lags from the maintained resolvent trace. Appendix C.2 gives the complete derivation.

Our notation bashes of $\mathcal{U}_{\eta,v}$ normalized the one-freedom-degree unit variance width. This dimensionless statistic is assigned as Spectral Trust, and our derivation is synthesized

Proposition 2 (Spectral trust)**.** *Suppose μ_v has full support. For an isotropic centered signal ensemble with coefficient variance $\sigma^2_{\xi,v}$,*

$$\mathbb{E}_g[\mathrm{Var}_{\mu_v}(Y_g)] = (n-1)\sigma^2_{\xi,v}\mathcal{U}^2_{\eta,v}. \tag{18}$$

Its root-mean-square scale is therefore $\sigma_{\xi,v}\sqrt{n-1}\,\mathcal{U}_{\eta,v}$.

Appendix C derives this identity by averaging the covariance expansion over centered directions. Accordingly, it determines both the topology penalty and the queue temperature.

5.3 CONVERGENCE CONDITIONS

After the final model change, assume a finite discounted MDP with an exact model and exact host backups. Each backup reads the current coherent value snapshot.

Theorem 1 (Fair selection and DP convergence)**.** *At each nonempty queue refresh k, draw a fresh independent Gumbel field and select at least one state. Suppose $T_k > 0$ and, for a fixed finite M,*

$$\frac{\max_{s\in\mathcal{C}_k} d_k(s) - \min_{s\in\mathcal{C}_k} d_k(s)}{T_k} \leq M.$$

Then every infinitely often eligible state is selected infinitely often almost surely, and

$$V_k \longrightarrow V^* \qquad \textit{almost surely}.$$

for the Value Iteration (VI) task using DP algorithm, matching the GTA-PS main result.

Proof. The softmax law assigns each eligible state a first-entry probability of at least e^{-M}/n. Its inclusion probability in the batch is at least as large. Consequently,

$$\Pr(\text{state } s \text{ is missed at } m \text{ eligible refreshes}) \leq (1 - e^{-M}/n)^m \longrightarrow 0.$$

Each infinitely often eligible state is therefore selected infinitely often almost surely.

States with eventually zero residual admit virtual Bellman updates that leave their values unchanged. The augmented schedule updates every state infinitely often. Let E_j be the maximum value error after the jth complete state-update cycle. Bellman contraction gives $E_{j+1} \leq \gamma E_j$. Each cycle is finite almost surely, so $E_j \to 0$ and $V_k \to V^*$. □

5.4 Computational complexity

The matrix jobs relies on the data structure used to hold its entry, in which sparse CSR array can hold sparse graph more efficient than of dense tensors. The regime separates representation cost and subsequent maintenance strategies, which envelopes into two different backend formulation.

Let c be the number of right-hand sides, ϕ_f the solve cost for the sparse factor anchor, and ℓ the retained physical correction width.

Backend	Resolvent action	Fresh commit
Dense	$O(n^2c)$	$O(n^2r + nr^2 + r^3)$
Sparse	$O((\phi_f + n\ell_v + \ell_v^2)c)$	$O(\phi_f r + n\ell_v r + \ell_v^2 r + nr^2 + r^3)$

Table 1: Operator-layer cost for a rank-r topology commit. Terms assume $r \ll n$ and exclude host backups.

The dense representation uses $O(n^2)$ memory, with incremental update reduces a small-row commit from a fresh $O(n^3)$ inversion to $O(n^2r)$. The sparse path store matrices by CSR form, aiding factor solves and low-rank actions, adding at most $O(nr^2)$ work. Both are realized as two compatible backend branches, treating the kernel density region.

6 Evaluation

We asks whether topology guidance remains useful after its operator, queue, and execution model are scaled beyond the setting in which GTA-PS was introduced. For this objective, we organize the evidence around planning quality, physical cost, and transfer to visual control, with one measurement boundary for each claim.

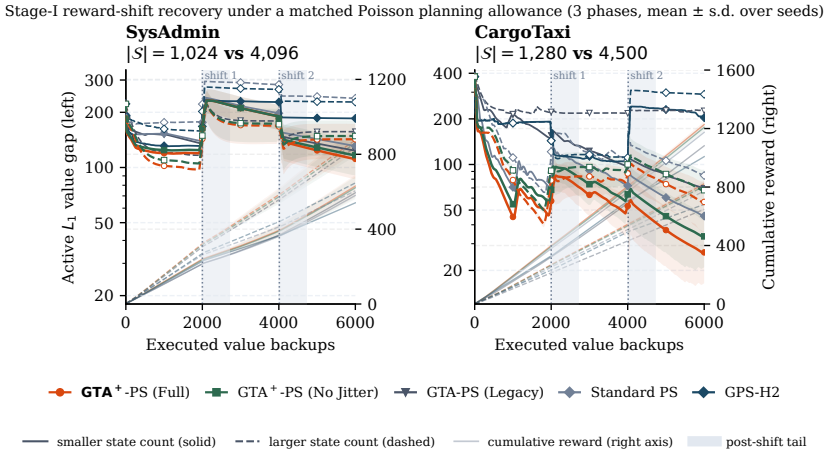

Figure 3: Stage-I task completion plots. Our two augmented methods raise distinctive planning recovery and reward gain relative to backup axis, claiming sample efficiency.

6.1 Evaluation ground

We run every comparison in Google Colab environment, each container is equipped with 12 logical CPUs and an NVIDIA A100 GPU. Except for the vanilla Standard-PS and GPS-H2 solvers - which host Numpy within CPU residence, all other methods are implemented with JAX for prioritization workloads and PyTorch for visual components. Acceleration substrate is tied to CUDA, 12th generation. In our intended resource bound, each method profile is assigned to no more than 4 concurrent threads and 10GB of peak GPU residence. Hyperparameter selection is made from a completely distant task, notably the FourRooms suite for the tabular domain, and the CartPole task with a sole visual grasp for the encoder domain. Further inquiries is resolved in F.6.

6.2 Tabular planning

Stage I tests planning under controlled reward drift in tabular settings. Five methods–Standard PS (Sutton, 1990), GPS-H2 (Wingate & Seppi, 2005), GTA-PS, deterministic GTA$^+$-PS, and full GTA$^+$-PS, are chosen for our measurement. The environment matrix consists of SysAdmin with 1,024 and 4,096 states and CargoTaxi with 1,280 and 4,500 realized states.

	SysAdmin		CargoTaxi	
Method	1,024	4,096	1,280	4,500
Standard PS	32.6	211.9	9.2	29.7
GPS-H2	43.1	243.4	12.3	50.3
GTA-PS†	129.8	1203.7	32.4	145.2
GTA$^+$-PS (det.)†	38.1	239.1	6.0	19.3
GTA$^+$-PS (full)†	32.2	202.3	5.1	16.3

Table 2: Physical planning seconds to 80% reduction of the final-shift L_1 gap, mean over seeds; dashes mark runs that exhaust the planning censor without crossing the threshold. †Projected recovery pending the transfer campaign.

6.3 Visual control

Stage II reports task outcomes at the 1M phase boundary and at 2M, 3M, 4M, and 5M cumulative decisions. The Pong task uses complete policy matches, whereas VizDoom reports Goal Success together with Navigation Performance over its fixed episode cohort.

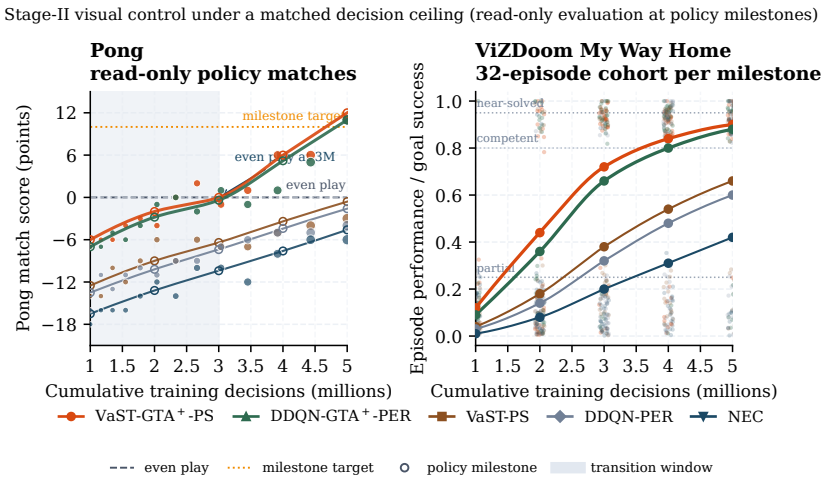

Figure 4: Stage-II task completion plots. For Pong, our methods kickoff with superior learning momentum and functioning persistence, gaining positive milestone score of 12 in double less decisions, comparing to the unaugmented control methods.

7 Discussion

7.1 Conclusion

We introduced GTA$^+$-PS, a scalable framework for graph-topology-augmented planning under non-stationary dynamics. By casting transition updates as low-rank shocks GTA$^+$-PS maintains policy resolvents through exact Woodbury updates and replaces costly eigensolves with a trace-derived spectral trust certificate. Combined with coherent versioned commits and an accelerator-resident execution model, GTA$^+$-PS guarantees fair state selection while delivering consistent recovery and throughput gains across both tabular MDPs and visual control domains.

LLM USAGE

We used generative AI tools to aid and polish the writing and to draft portions of the manuscript, as declared in our OpenReview submission form. The tools proposed prose, organization, and LaTeX expressions for author-specified arguments and results. The authors selected, checked, and revised every assisted passage, and thus take full responsibility for the final text, mathematical statements, citations, experimental descriptions, and claims.

REFERENCES

Walter E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9(1):17–29, 1951. doi: 10.1090/qam/42792. URL https://doi.org/10.1090/qam/42792.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013. doi: 10.1613/jair.3912. URL https://doi.org/10.1613/jair.3912.

Yuan Cheng, Jing Yang, and Yingbin Liang. Provably efficient algorithm for nonstationary low-rank MDPs. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0277. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/145c28cd4b1df9b426990fd68045f4f7-Abstract-Conference.html.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Reinforcement learning for non-stationary Markov decision processes: The blessing of (more) optimism. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1843–1854. PMLR, 2020. URL https://proceedings.mlr.press/v119/cheung20a.html.

Fan Chung. Laplacians and the Cheeger inequality for directed graphs. *Annals of Combinatorics*, 9(1):1–19, 2005. doi: 10.1007/s00026-005-0237-z. URL https://doi.org/10.1007/s00026-005-0237-z.

Dane Corneil, Wulfram Gerstner, and Johanni Brea. Efficient model-based deep reinforcement learning with variational state tabulation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1049–1058. PMLR, 2018. URL https://proceedings.mlr.press/v80/corneil18a.html.

Tuan Dam. Optimal regret bounds via low-rank structured variation in non-stationary reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3319. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/8f130483b866619fbad1c7fa31475960-Abstract-Conference.html.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993. doi: 10.1162/neco.1993.5.4.613. URL https://doi.org/10.1162/neco.1993.5.4.613.

James W. Demmel, Stanley C. Eisenstat, John R. Gilbert, Xiaoye S. Li, and Joseph W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM Journal on Matrix Analysis and Applications*, 20(3):720–755, 1999. doi: 10.1137/S0895479895291765. URL https://doi.org/10.1137/S0895479895291765.

Thomas G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000. doi: 10.1613/JAIR.639. URL https://doi.org/10.1613/jair.639.

Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, 19:399–468, 2003. doi: 10.1613/JAIR.1000. URL https://doi.org/10.1613/jair.1000.

Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. ViZDoom: A Doom-based AI research platform for visual reinforcement learning. In *2016 IEEE Conference on Computational Intelligence and Games*, pp. 1–8, 2016. doi: 10.1109/CIG.2016.7860433. URL https://doi.org/10.1109/CIG.2016.7860433.

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The Gumbel-top-k trick for sampling sequences without replacement. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3499–3508. PMLR, 2019. URL https://proceedings.mlr.press/v97/kool19a.html.

Xiaoye S. Li, James W. Demmel, John R. Gilbert, Laura Grigori, Piyush Sao, Meiyue Shao, and Ichitaro Yamazaki. SuperLU users' guide. Technical Report LBNL-44289, Lawrence Berkeley National Laboratory, September 1999. URL https://portal.nersc.gov/project/sparse/superlu/ug.pdf. Last updated June 2018.

Lin Lin, Chao Yang, Juan C. Meza, Jianfeng Lu, Lexing Ying, and Weinan E. SelInv: An algorithm for selected inversion of a sparse symmetric matrix. *ACM Transactions on Mathematical Software*, 37 (4):40:1–40:19, 2011. doi: 10.1145/1916461.1916464. URL https://doi.org/10.1145/1916461.1916464.

Hao Liu and Pieter Abbeel. APS: Active pretraining with successor features. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6736–6747. PMLR, 2021. URL https://proceedings.mlr.press/v139/liu21b.html.

Marlos C. Machado, Marc G. Bellemare, and Michael Bowling. A Laplacian framework for option discovery in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2295–2304. PMLR, 2017. URL https://proceedings.mlr.press/v70/machado17a.html.

Sridhar Mahadevan and Mauro Maggioni. Proto-Value Functions: A Laplacian Framework for Learning Representation and Control in Markov Decision Processes. *Journal of Machine Learning Research*, 8:2169–2231, 2007.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236. URL https://doi.org/10.1038/nature14236.

Andrew W. Moore and Christopher G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13(1):103–130, 1993. doi: 10.1007/BF00993104. URL https://doi.org/10.1007/BF00993104.

Jing Peng and Ronald J. Williams. Efficient learning and planning within the Dyna framework. *Adaptive Behavior*, 1(4):437–454, 1993. doi: 10.1177/105971239300100403. URL https://doi.org/10.1177/105971239300100403.

Hung Pham and Tuan Dam. Spectral prioritized sweeping in nonstationary reinforcement learning. 2026. URL https://arxiv.org/abs/2609.06186.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adrià Puigdomènech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2827–2836. PMLR, 2017. URL https://proceedings.mlr.press/v70/pritzel17a.html.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994. ISBN 978-0-471-61977-2. doi: 10.1002/9780470316887. URL https://onlinelibrary.wiley.com/doi/book/10.1002/9780470316887.

Aliaksei Sandryhaila and José M. F. Moura. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656, 2013. doi: 10.1109/TSP.2013.2238935. URL https://doi.org/10.1109/TSP.2013.2238935.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *4th International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.05952.

Jack Sherman and Winifred J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, 1950. doi: 10.1214/aoms/1177729893. URL https://doi.org/10.1214/aoms/1177729893.

David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013. doi: 10.1109/MSP.2012.2235192. URL https://doi.org/10.1109/MSP.2012.2235192.

David Silver, Richard S. Sutton, and Martin Müller. Sample-based learning and search with permanent and transient memories. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 968–975. ACM, 2008. doi: 10.1145/1390156.1390278. URL https://doi.org/10.1145/1390156.1390278.

Matthew J. Sobel. The variance of discounted Markov decision processes. *Journal of Applied Probability*, 19(4):794–802, 1982. doi: 10.2307/3213832. URL https://doi.org/10.2307/3213832.

Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine Learning Proceedings 1990*, pp. 216–224. Morgan Kaufmann, 1990. doi: 10.1016/B978-1-55860-141-3.50030-4. URL https://doi.org/10.1016/B978-1-55860-141-3.50030-4.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6306–6315, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2094–2100, 2016. doi: 10.1609/aaai.v30i1.10295. URL https://doi.org/10.1609/aaai.v30i1.10295.

Harm Van Seijen and Richard S. Sutton. Planning by prioritized sweeping with small backups. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 361–369. PMLR, 2013. URL https://proceedings.mlr.press/v28/vanseijen13.html.

David Wingate and Kevin D. Seppi. Prioritization methods for accelerating MDP solvers. *Journal of Machine Learning Research*, 6(29):851–881, 2005. URL https://www.jmlr.org/papers/v6/wingate05a.html.

Max A. Woodbury. Inverting modified matrices. Technical Report 42, Statistical Research Group, Princeton University, 1950.

APPENDIX CONTENTS

A Notation . . . 13
B Operator derivations . . . 16
C Spectral trust . . . 19
D Numerical stability . . . 21
E Algorithms . . . 24
F Experimental protocol . . . 29
G Additional results . . . 32
H Reproducibility . . . 33

A NOTATION

This section fixes the objects that the proofs and execution protocol share. The distinctions among mathematical state, logical progress, and physical execution are part of the method: collapsing them would make the update rule or its measurements ambiguous.

A.1 SYMBOLS AND INDICES

Throughout our paper, f denotes a *Topology Version*, v an *Agent Version*, k a Planning Backup index, d an environment decision index, and θ elapsed wall time.

Table 3: Canonical notation used throughout the paper.

Symbol	Meaning
P_v^a, P_v	Admitted action-conditioned model and the derived topology-policy view $P_v = \mathcal{P}(\pi_v, p_v)$.
$A_{\eta,v}$, $G_{\eta,v}$	$A_{\eta,v} = (1+\eta)I - P_v$ and $G_{\eta,v} = A_{\eta,v}^{-1}$, with $\eta > 0$.
$A_{\eta,v}^{\text{in/out}}$, $G_{\eta,v}^{\text{in/out}}$	Incoming/outcoming system $\text{diag}(P_v^\top \mathbf{1}) - P_v^\top + \eta I$ and its inverse.
ξ_v, τ_v	Entrywise residual magnitude and its in-out mixture from Equation (4).
U_v, V_v, Z_v, K_v	Version-local factors $\Delta P_v = U_v V_v^\top$, basis actions $Z_v = G_{\eta,v} U_v$, and core $K_v = I - V_v^\top Z_v$.
Z_v, W_v, C_v	$Z_v = G_{\eta,v} U_v$, $W_v = G_{\eta,v}^\top V_v$, and $C_v = (I - V_v^\top Z_v)^{-1}$.
F_v	Maintained resolvent trace, $F_v = \text{tr}(G_{\eta,v})$.
$\Psi_{\eta,v}, \mathcal{U}_{\eta,v}$	Integrated non-Perron covariance and its normalized standard-deviation multiplier.
c_{op}, c_{rhs}, c_Q	Operator staging width, inverse-action width, and planning-batch width.
f, ℓ	Sparse factor-anchor version (Topology Version) and physical correction width.
ρ_v, ϵ_{inv}	Coupled inverse residual and its single recovery threshold.
L_f, R_f	Sparse triangular factors of the realized system at anchor version f.
λ_π, T_v	Topology-policy softmax temperature and Common queue temperature.
$w_v(c)$, $h_v(c)$, $\chi_v(c)$	Positive PER base priority, GTA replay score, and their joint logit for candidate c.
α_{PER}, β_{PER}	PER priority exponent and importance-weight exponent

A.2 ASSUMPTIONS

The algebraic results use a finite state space and a nonnegative row-stochastic P_v. The kernel may be directed, non-normal, non-reversible, or periodic. Since $\rho(P_v) = 1 < 1 + \eta$, the resolvent exists and

$$G_{\eta,v} = \frac{1}{1+\eta} \sum_{h \geq 0} (1+\eta)^{-h} P_v^h, \qquad \|G_{\eta,v}\|_\infty = \eta^{-1}. \tag{19}$$

The resolvent and trace identities require only a finite row-stochastic kernel. The trajectory-variance result additionally assumes $X_0 \sim \mu_v$ for a stationary distribution μ_v. Its full-state isotropic interpretation requires $\mu_v(s) > 0$ for every state. Within these, strong connectivity plus aperiodicity assumption are sufficient. (Pham & Dam, 2026) provides entry-wise perturbation that guarantee both assumption be satisfied.

Even if absorbing states exist, which render strong connectivity fails, appendix C.3 develops a reconciled trace statistic to mitigate non-communication. Overall, the main convergence theorem separately assumes eventual stationarity, oracle value backups, and persistent positive ground.

For computational efficiency in sparse tabular models, the transition graph possesses bounded local degree with at most $b \ll n$ successors per action-conditioned row and effective update rank $r_v \ll n$, enabling near-linear operator action and commit scaling.

A.3 Agent Semantics

An *Agent Version* is the canonical time-series index v of one agent's learned state, incremented once per inference-learning cadence. It is frozen starting the inference, with the learning opens a small window for the update. Figure 5 sketches this cadence.

For inference, the frozen version is cloned into *Actors*, which concurrently samples its policy to generate data within separated sub-seed. Then, the data is collected in the planning module. The Planner commits its learning work, and publishes operator, trust, priority, policy, and queue state. Only after prioritization is affirmed, the learning proceeds with its data ordering. The agent sequentially consume the data stream, optimizes itself, and finally incrementing to the next agent version.

The *Operator Version* is the planner's operator-side index and advances only when publication changes the induced topology operator. While the Operator Version and Agent Version are separated (operator maintenance versus phase publication), both operate within the shared life interval v. In contrast, the *Topology Version* f defines the coarser reference anchor interval for the sparse Laplacian and resolvent factorization ($A_{\eta,f} = L_f R_f$), which is maintained across multiple operator and agent versions and updated only upon scheduled rebase or physical compaction.

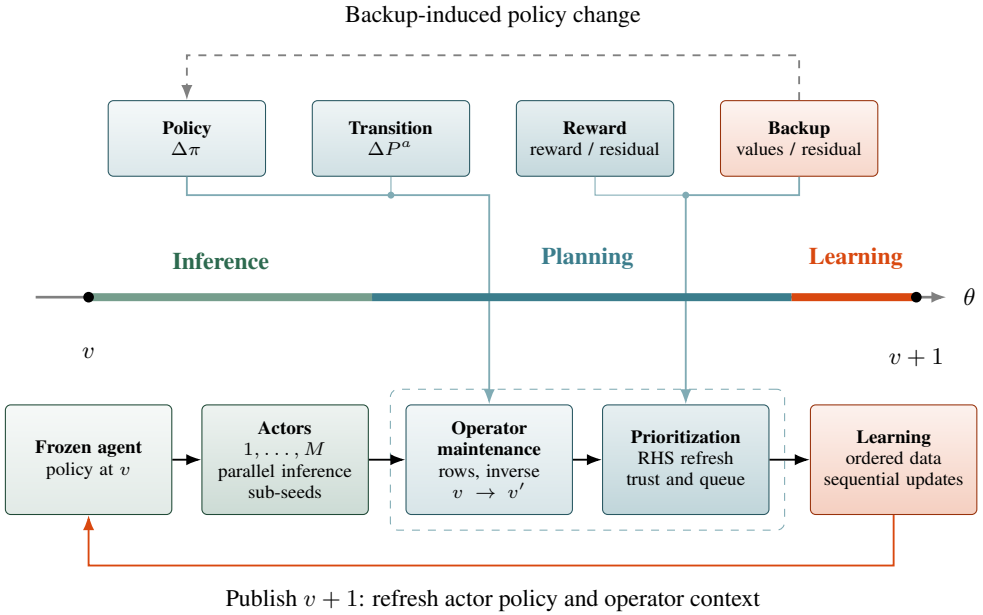

Figure 5: The Inference-Learning phase timeline. The prioritization joint is right before data consumption

A.4 Event Semantics

During inference, four event classes enter the planner model, and only two can change the operator.

A *Belief Batch* is the net set of staged policy and model row changes since the last publication. A *Coherent Commit* makes the new policy, realized systems, resolvent representation, trace, trust statistic, topology field, priority snapshot, and *Operator Version* visible together. Numerical Rebase

repairs the systems and inverse representations against the same admitted primitives after a certificate failure, preserving v without requiring another numerical index.

Across the timeline in Figure 5, events reside in sequential lanes: transition model and reward updates stream concurrently during inference; the resolvent and spectral trust update during planning upon Coherent Commit; and backup and neural updates execute during learning. Finally, policy updates land after learning, initializing the next cycle with the new Operator Version.

Table 4: Event ownership and Operator Version semantics.

Event	Changed object	Publication rule
Policy	π_v and the induced system shock	Stage rows; advance v once at Coherent Commit.
Transition model	Rows of p_v and the induced system shock	Stage rows; advance planning representation at Coherent Commit.
Reward	Host reward field and derived residual	Refresh right-hand sides; leave v unchanged.
Backup	Host value state, policy and the residual	the host report consequences, advancing v

Two numerical observations serve separate owners:

- **Primitive mass diagnostic:** Softmax and the transition-model owner establish stochasticity. Derived row-mass error, including the former 10^{-8} observation, is diagnostic and never requests numerical recovery.
- **Inverse certificate precision (5×10^{-4}):** The scheduled coupled residual checks the complete moving systems and resolvents against their defining primitives and resident realizations.

A failed Inverse Certificate requests internal numerical repair. Rank controls physical staging and compaction within a version transition. Appendix D details the certificate and sparse representation.

A.5 Device Semantics

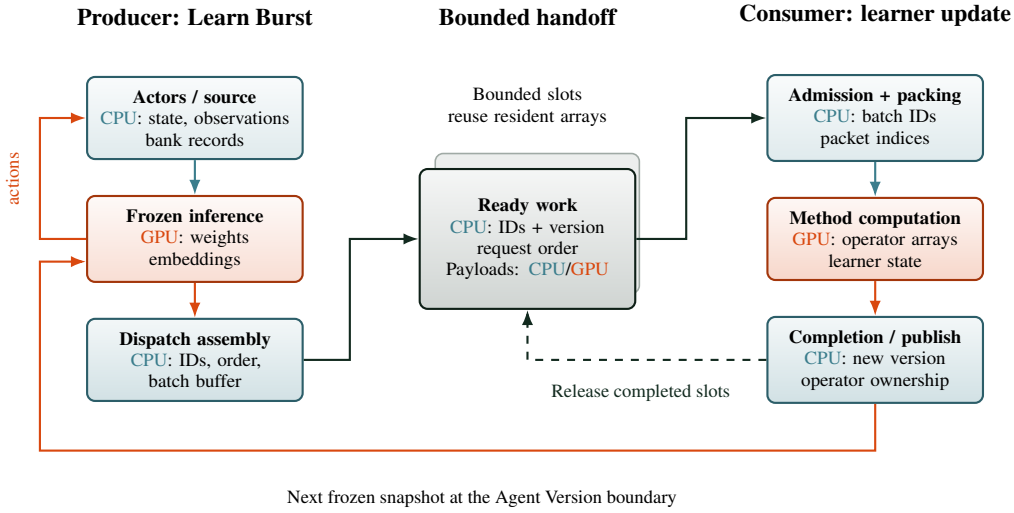

Figure 6: Producer and consumer each span CPU and GPU. Bounded handoff preserves work identity; publication refreshes the frozen snapshot.

Following the acceleration model, our mapping assigns orchestration and environment stepping to the CPU, while batched inference, learning arithmetic, and supported agent operations execute on the GPU. CPU–GPU transmission required inputs and results following the producer-consumer pattern.

In the inference phase, the *Actors* produce *Generation Dispatch*, consisting of learning units of backups or trajectories, in defined number of waves, called *Learn Burst*. In the learning phase, the GPU Packing *consumer*, which follows bin-packing style, pack learning units onto the desired device, ingested by the method's algorithm.

Our methods are evaluated within concurrent or sequential turns, inside a *Runtime Container*. The evaluation runtime is enveloped in the *Execution Profile*, which assigns method, task, seeds, and resources. As described in 6.1, each profile declares four logical CPU slots plus its host and device memory allocation within the provider envelope. Up to three profiles can be hosted concurrently, which may share admitted resources and transmission line. This, though might cause minor clock drift, still ensure integrity and accuracy of the result.

B Operator Derivations

The structured-variation model guide us as a blueprint for crafting the planner's operator identity. Programmatically, we want to sparingly realize intermediate identities and reduce active-entry storage. This section is to dictate the algebraic operator derivation, plus its mapping onto program-side residence. Once those factors are fixed, the resolvent, its block actions, and its trace follow from simple update routine.

B.1 Shock factorization

Let $\mathcal{R}_v = \{i_1, \dots, i_r\}$ contain the distinct rows changed by a Belief Batch. Define

$$U_v = [e_{i_1} \quad \dots \quad e_{i_r}], \qquad V_v^\top = \begin{bmatrix} (\Delta P_v)_{i_1,:} \\ \vdots \\ (\Delta P_v)_{i_r,:} \end{bmatrix}. \tag{20}$$

Then

$$\Delta P_v = U_v V_v^\top, \qquad \operatorname{rank}(\Delta P_v) \le \ell, \qquad V_v^\top \mathbf{1} = 0. \tag{21}$$

The last identity is the row-stochasticity constraint. Repeated writes to one state contribute one net row, and exact rank compression may remove dependent directions.

For completeness, a joint policy and transition event decomposes as

$$\begin{aligned} \Delta P_v = &\sum_a \operatorname{diag}(\Delta\pi_v(\cdot, a)) P_v^a + \sum_a \operatorname{diag}(\pi_v(\cdot, a)) \Delta P_v^a \\ &+ \sum_a \operatorname{diag}(\Delta\pi_v(\cdot, a)) \Delta P_v^a. \end{aligned} \tag{22}$$

The third term is second order in the primitive changes. The committed row difference retains it exactly.

Reward updates comes from the Belief Batch too, though materialize simply by vector increment:

$$r_{v+1} = r_v + \Delta r_v \tag{23}$$

B.2 Woodbury update

Set $A_{\eta,v} = (1+\eta)I - P_v$. A committed shock gives

$$A_{\eta,v+1} = A_{\eta,v} - U_v V_v^\top. \tag{24}$$

Theorem 2 (Exact structured resolvent update)**.** *If P_v and $P_{v+1} = P_v + U_v V_v^\top$ are row-stochastic and $\eta > 0$, then $I_r - V_v^\top G_{\eta,v} U_v$ is nonsingular and*

$$G_{\eta,v+1} = G_{\eta,v} + G_{\eta,v} U_v C_v V_v^\top G_{\eta,v}. \tag{25}$$

Proof. Both $A_{\eta,v}$ and $A_{\eta,v+1}$ are nonsingular by Equation (19). The matrix determinant lemma gives

$$\det(A_{\eta,v+1}) = \det(A_{\eta,v}) \det(I_r - V_v^\top G_{\eta,v} U_v). \tag{26}$$

The core determinant is therefore nonzero. Applying the Woodbury identity to $A_{\eta,v} - U_v V_v^\top$ proves Equation (25). □

Incoming update. For the same shock $\Delta P_v = U_v V_v^\top$, the incoming system changes by

$$\Delta A^{\text{in}}_{\eta,v} = \operatorname{diag}(\delta_v) - V_v U_v^\top, \qquad \delta_v = V_v U_v^\top \mathbf{1}. \tag{27}$$

Let J_v select the nonzero coordinates of δ_v. Set $B_v = [-V_v,\ J_v \operatorname{diag}(J_v^\top \delta_v)]$ and $D_v = [U_v,\ J_v]$. Then $\Delta A^{\text{in}}_{\eta,v} = B_v D_v^\top$, and

$$G^{\text{in}}_{\eta,v+1} = G^{\text{in}}_{\eta,v} - G^{\text{in}}_{\eta,v} B_v (I + D_v^\top G^{\text{in}}_{\eta,v} B_v)^{-1} D_v^\top G^{\text{in}}_{\eta,v}. \tag{28}$$

Strict row diagonal dominance with margin η makes both incoming systems nonsingular, so the Woodbury core is nonsingular. The uncompressed correction width is at most $r + |\operatorname{supp}(\delta_v)|$. Coalescing its affected columns gives an alternative basis with at most n columns. Thus, the update shock will invoke at most $\mathcal{O}(nr)$ entry operation.

B.3 Sparse and Dense Residence

The methods hold explicit (π_v, p_v) pair. The topology factor thus materialize and incrementally update the Laplacians $A^{\text{out}}_{\eta,v}$ and $A^{\text{in}}_{\eta,v}$. The dense backend reside explicit $n \times n$ systems, their resolvents and applies updates through dense BLAS. Sparse regime, otherhand, can not afford such $\mathcal{O}(n^2)$ storage. The backend retains the Laplacians in CSR form and represents inverse actions through triangular factors plus exact core corrections. Its factor anchor f may move during a version transition for accumulated nonstationary compaction, independently of the certificate schedule.

B.4 Exact inverse actions

Downstream computations require operator inverse actions on right-hand vectors and blocks far more frequently than materializing a dense inverse matrix. For any state operand $X \in \mathbb{R}^{n\times c}$ and co-state stimulus $V \in \mathbb{R}^{n\times r}$, the planner resolves two canonical action equations:

$$\text{Forward action equation:} \qquad A_{\eta,v} Y = X \iff Y = G_{\eta,v} X, \tag{29}$$

$$\text{Dual action equation:} \qquad A_{\eta,v}^\top W = V \iff W = G_{\eta,v}^\top V. \tag{30}$$

The representation substrate determines how these action equations are evaluated across Operator Versions v and numerical reconstructions.

Dense path. The dense backend explicitly maintains $A^{\text{out}}_{\eta,v}$ and $A^{\text{in}}_{\eta,v}$ together with their full inverse resolvents. Evaluating Equations (29)–(30) then reduces to direct matrix multiplication. After these operators complete a version update, an RHS change forms $\widetilde{Y}_v = G_{\eta,v}(H_v + \Delta H_v)$.

Proposition 3a (Exact block-action update)**.** *Under a committed rank-r transition shock $\Delta P_v = U_v V_v^\top$, the post-commit action satisfies the exact v-cadence recurrence:*

$$Y_{v+1} = \widetilde{Y}_v + Z_v C_v V_v^\top \widetilde{Y}_v, \tag{31}$$

where $Z_v = G_{\eta,v} U_v$ and $C_v = (I_r - V_v^\top Z_v)^{-1}$.

Proof. Right-multiply Equation (25) by $H_v + \Delta H_v$ and use $Z_v = G_{\eta,v} U_v$. □

In summary, the computation path is realized as:

$$\begin{aligned}
\text{Linear systems:} \quad & \pi_v \longrightarrow P_v \longrightarrow (A^{\text{out}}_{\eta,v}, A^{\text{in}}_{\eta,v}) \longrightarrow (G^{\text{out}}_{\eta,v}, G^{\text{in}}_{\eta,v}),\\
\text{Spectral trust:} \quad & (G^{\text{out}}_{\eta,v}, G^{\text{in}}_{\eta,v}) \longrightarrow F_v \longrightarrow \Psi_{\eta,v} \longrightarrow \mathcal{U}_{\eta,v} \longrightarrow (\beta_v, T_v),\\
\text{Diffusion:} \quad & (\xi_v, G^{\text{out}}_{\eta,v}, G^{\text{in}}_{\eta,v}) \longrightarrow (\tau_v^{\text{out}}, \tau_v^{\text{in}}) \longrightarrow \tau_v,\\
\text{Joint queue branch:} \quad & (\xi_v, \tau_v, \beta_v, T_v) \longrightarrow d_v \longrightarrow p_v,
\end{aligned} \tag{32}$$

Sparse path. In the sparse regime, $G_{\eta,v}$ is never materialized as an explicit matrix. The backend incrementally maintains the Laplacians and holds triangular factors anchored at version f, together with any exact correction required by the active inverse representation. The version-local shock $U_v V_v^\top$ is consumed after A and LR complete their moving update.

Proposition 3b (Sparse-region factor-solve action)**.** *Let the anchor system $A_{\eta,f}$ have sparse factors L_f, R_f and basis actions $Z_v = G_{\eta,f}U_v$. The forward action equation $A_{\eta,v}Y = X$ is solved with algebraic exactness at version v via:*

1. ***Triangular inverse solve:*** *Solve the sparse triangular systems by forward and backward substitution:*
$$L_f B = \Pi_f^{\text{row}} X, \qquad R_f C = B, \qquad Y_0 = \Pi_f^{\text{col}} C = A_{\eta,f}^{-1} X. \tag{33}$$
2. ***Low-rank core solve and correction:*** *Solve the small $\ell \times \ell$ system $K_v Q = V_v^\top Y_0$ with $K_v = I - V_v^\top Z_v$, and compute*
$$Y = Y_0 + Z_v Q = G_{\eta,v} X. \tag{34}$$

The dual action equation $A_{\eta,v}^\top W = V$ is evaluated analogously by transposing the triangular factor solves and the low-rank correction.

Proof. Applying the Woodbury identity to $A_{\eta,v} = A_{\eta,f} - U_v V_v^\top$ gives the stated action. Setting $Y_0 = G_{\eta,f}X$ and $Z_v = G_{\eta,f}U_v$ yields $Y = Y_0 + Z_v K_v^{-1} V_v^\top Y_0$. Transposing the factors yields the dual action. □

The Woodbury layer therefore depends on inverse actions solely, and its algebraic identities are conserved between backend storage. Program mapping, on the other hand, require more intricate storage. The computation path is reconciled as below.

$$\begin{aligned}
\text{Factor anchor systems:} &\quad \pi_v \longrightarrow P_v \longrightarrow (A_{\eta,f}^{\text{out}}, A_{\eta,f}^{\text{in}}) \longrightarrow (L_f, R_f, \Pi_f) \quad [\longrightarrow (G_{\eta,f}^{\text{out}}, G_{\eta,f}^{\text{in}})], \\
\text{Sparse action branch:} &\quad (\xi_v, L_f, R_f, \Pi_f) \xrightarrow{\text{triangular}} Y_0 \xrightarrow{K_v, Z_v} Y_v \longrightarrow \tau_v \quad [\longrightarrow G_{\eta,v}\xi_v], \\
\text{Spectral truth recurrence:} &\quad (Z_v, W_v, C_v, F_v) \longrightarrow F_{v+1} \longrightarrow \mathcal{U}_{\eta,v} \longrightarrow (\beta_v, T_v) \quad [\longrightarrow \operatorname{tr}(G_{\eta,v})], \\
\text{Joint queue branch:} &\quad (\xi_v, \tau_v, \beta_v, T_v) \longrightarrow d_v \longrightarrow p_v,
\end{aligned} \tag{35}$$

Trace identity Dense path realization of G comfortably help extracting the trace in $\mathcal{O}(n)$ diagonal sum. For the sparse path, $G_{\eta,v}$ is not explicitly realized. One can explicitly multiply L and R, taking the outer trace to derive. However, we can figure out the exact (per operator version) update identity for the trace. The trace uses the paired forward and dual actions Z_v and W_v:

Proposition 4 (Exact trace recurrence)**.** *We have*
$$F_{v+1} = F_v + \operatorname{tr}(C_v W_v^\top Z_v). \tag{36}$$

Proof. Take the trace of the Woodbury correction and cycle its factors:
$$\operatorname{tr}(GU_v C_v V_v^\top G) = \operatorname{tr}(C_v V_v^\top G^2 U_v) = \operatorname{tr}(C_v W_v^\top Z_v). \tag{37}$$
□

B.5 Repeated commits

Exactness across commits follows by induction over Operator Versions.

Corollary 1 (Exactness across commits)**.** *For the dense path, let $v_0 < \cdots < v_J$ be admissible Operator Versions. Successive applications of Equation (25) satisfy*
$$G_{\eta,v_j} = ((1+\eta)I - P_{v_j})^{-1}, \qquad j = 0, \ldots, J, \tag{38}$$
in exact arithmetic. The sparse path obeys the same version sequence. Representation reconstruction at fixed v does not alter the exact-arithmetic statement.

Proof. The claim holds at the reference version. Theorem 2 maps an exact inverse at v_j to the exact inverse at v_{j+1}, so induction completes the argument. □

Staging changes the decomposition and execution time of a net shock. It does not change the final resolvent in exact arithmetic. Finite-precision drift is a separate question, treated in Appendix D.

C SPECTRAL TRUST

Spectral trust joins an algebraic matrix breadth to a probabilistic interpretation. Intuitively, the trace of a matrix square is the result's Frobenius norm, thus the variance, look in the lens of its covariate matrix's second moment, can conceivably be extracted. This also survives for kernels whose stationary dynamics no longer support the full-state variance ensemble. This section state the two claims separately and identifies the communication assumption that connects them.

C.1 SIGNAL VARIANCE

Write $\kappa = (1+\eta)^{-1}$. Let $X_0 \sim \mu_v$ and let $g \in L_2(\mu_v)$ satisfy $\mu_v^\top g = 0$. Define

$$Y_g = \sum_{h \geq 0} \kappa^h g(X_h), \qquad \langle x, y \rangle_{\mu_v} = x^\top \operatorname{diag}(\mu_v) y. \tag{39}$$

Proposition 5 (Discounted signal variance)**.**

$$\operatorname{Var}_{\mu_v}(Y_g) = \frac{\|g\|_{2,\mu_v}^2 + 2\sum_{h \geq 1} \kappa^h \langle g, P_v^h g \rangle_{\mu_v}}{1 - \kappa^2}. \tag{40}$$

Proof. Stationarity and centering give $\mathbb{E}[g(X_i)g(X_j)] = \langle g, P_v^{j-i} g \rangle_{\mu_v}$ for $j \geq i$.

Expanding the square product, then grouping terms by lag, and summing $\sum_{i \geq 0} \kappa^{2i} = (1-\kappa^2)^{-1}$ yields Equation (40). □

C.2 TRACE IDENTITY

Let $d \geq 0$ satisfy $d^\top \mathbf{1} = 1$ and define the deflated kernel $X_{d,v} = P_v - \mathbf{1} d^\top$. For $h \geq 1$,

$$\operatorname{tr}(X_{d,v}^h) = \operatorname{tr}(P_v^h) - 1. \tag{41}$$

Define

$$m_{h,v} = \frac{\operatorname{tr}(X_{d,v}^h)}{n-1}, \qquad \Psi_{\eta,v} = \sum_{h \geq 1} \kappa^h m_{h,v}. \tag{42}$$

Assume now that μ_v has full support. Then the centered subspace $\mathcal{H}_v = \{g \in L_2(\mu_v) : \mu_v^\top g = 0\}$ has dimension $n-1$. For an isotropic ensemble in $\mathcal{H}_v$ with coefficient variance $\sigma_{\xi,v}^2$, Proposition 5 gives

$$\mathbb{E}_g[\operatorname{Var}_{\mu_v}(Y_g)] = (n-1)\sigma_{\xi,v}^2 \frac{1 + 2\Psi_{\eta,v}}{1 - \kappa^2}. \tag{43}$$

Theorem 3 (Resolvent trace identity)**.**

$$\Psi_{\eta,v} = \frac{(1+\eta)F_v - n - \eta^{-1}}{n-1}. \tag{44}$$

Proof. The resolvent series implies

$$\kappa^{-1} F_v = n + \sum_{h \geq 1} \kappa^h \operatorname{tr}(P_v^h). \tag{45}$$

Substitute $\operatorname{tr}(P_v^h) = 1 + (n-1)m_{h,v}$, sum the Perron contribution, and use $\kappa/(1-\kappa) = \eta^{-1}$. □

C.3 MODIFICATION FOR NON-COMMUNICATION

Proposition 5 requires stationarity of the Markov induced kernel. The $(n-1)$-dimensional isotropic ensemble used in Equation (43) further requires a stationary measure with full support. Strong connectivity supplies such a measure. In the weak connectivity case, a finite reducible chain decomposes into transient states and closed communicating classes. However, for transient states, stationary measure assigns zero mass, effectively regard them as non-existent. To properly address this, we treat their count separately:

Definition 1 (Terminal state)**.** *A terminal state † is an absorbing MDP state:*

$$\Pr(S_{d+1} = \dagger \mid S_d = \dagger) = 1. \tag{46}$$

Let $P_v \in \mathbb{R}^{N\times N}$ be a substochastic kernel over the ordinary states. Terminal Completion assigns its missing row mass $p_v^\dagger = \mathbf{1} - P_v\mathbf{1}$ to †:

$$\widetilde{P}_v = \begin{bmatrix} P_v & p_v^\dagger \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{47}$$

Terminal Completion resolves the mismatch between episodic planning and the trace identity: terminal outcomes make P_v substochastic, and Equation (47) restores one unit of mass in every row. We use the completed kernel only for the trust reduction; the original MDP states and Bellman updates are unchanged.

Write $F_v^{\mathrm{plan}} = \mathrm{tr}(((1+\eta)I_N - P_v)^{-1})$. The completed resolvent is block triangular, so its dimension is $N+1$ and its trace is $\widetilde{F}_v = F_v^{\mathrm{plan}} + \eta^{-1}$. Substitution into Equation (44) yields the amended trust

$$\widetilde{\Psi}_{\eta,v} = \frac{(1+\eta)F_v^{\mathrm{plan}} - N}{N}, \qquad \widetilde{\mathcal{U}}_{\eta,v} = \sqrt{\frac{[1+2\widetilde{\Psi}_{\eta,v}]_+}{1-\kappa^2}}. \tag{48}$$

Terminal Completion changes only the trust reduction. The leading resolvent block remains $((1+\eta)I_N - P_v)^{-1}$, so topology actions with a zero terminal component and the host's terminal Bellman updates are unchanged.

Stage II tasks are completion task, and the task completion state is absorbing, therefore they use $\widetilde{\mathcal{U}}_{\eta,v}$ as an algebraic scheduling statistic.

C.4 Trust Estimator

Normalize Equation (43) by the input scale:

$$R_{\eta,v} = \frac{[1+2\Psi_{\eta,v}]_+}{1-\kappa^2}, \qquad \mathcal{U}_{\eta,v} = \sqrt{R_{\eta,v}}. \tag{49}$$

Thus $\sigma_{\xi,v}\sqrt{n-1}\,\mathcal{U}_{\eta,v}$ is the root-mean-square amplification of the isotropic centered ensemble. This is an average over centered directions. The positive part protects against a slightly negative finite-precision trace reduction, mainly for symbolic preservation.

At sparse factor construction, block solves against the identity give the reference trace without storing a dense inverse. Subsequent commits use Equation (36). Scheduled representation checks refresh the trace from the same streamed inverse columns, while ordinary maintenance reuses the paired inverse actions.

C.5 Priority Coupling

The topology weight and common queue temperature are

$$\beta_v = \beta_1[1+\beta_2 - \sqrt{\lambda_\omega}\mathcal{U}_{\eta,v}]_+, \qquad T_v = \sqrt{n-1}\,\mathcal{U}_{\eta,v}. \tag{50}$$

The temperature follows from the trace-derived scale and the state count. Meanwhile, $\sigma_{\xi,v}$ in Equation (43) specifies the signal ensemble used in the variance analysis. Draw one independent canonical Gumbel for each candidate:

$$g_v(s) \overset{\mathrm{iid}}{\sim} \mathrm{Gumbel}(0,1), \qquad \epsilon_v^{\mathrm{jit}}(s) = T_v g_v(s). \tag{51}$$

For $d_v(s) = \xi_v(s) + \beta_v\tau_v(s)$, the priority is

$$p_v(s) = d_v(s) + T_v g_v(s). \tag{52}$$

Conditioned on a finite nonempty frozen candidate set $\mathcal{C}_v$, scores d_v, and $T_v > 0$, the Gumbel-max identity gives

$$\Pr\left(\arg\max_{s\in\mathcal{C}_v} p_v(s) = i \mid d_v, T_v, \mathcal{C}_v\right) = \frac{\exp(d_v(i)/T_v)}{\sum_{j\in\mathcal{C}_v}\exp(d_v(j)/T_v)}. \tag{53}$$

Gumbel-max identity. Fix the conditioning above and set $a_i = \exp(d_v(i)/T_v)$. The standard Gumbel CDF implies $E_i = e^{-g_v(i)} \overset{\text{iid}}{\sim} \text{Exp}(1)$. Thus $S_i = E_i/a_i = e^{-p_v(i)/T_v}$ are independent exponentials with rates a_i, and

$$\Pr(S_i = \min_{j \in \mathcal{C}_v} S_j) = \int_0^\infty a_i \exp\left(-t \sum_{j \in \mathcal{C}_v} a_j\right) dt = \frac{a_i}{\sum_{j \in \mathcal{C}_v} a_j}. \tag{54}$$

Maximizing p_v minimizes S, proving Equation (53); ties have probability zero. See also Kool et al. (2019, Section 2.3). □

The top c_Q entries form an ordered sample without replacement from these weights (Kool et al., 2019). A larger trust statistic reduces deterministic reliance on topology and raises the sampling temperature. Positive temperature gives positive selection probability for anti-starvation, though an infinite-horizon fairness claim also requires the probability bounds in Theorem 1.

D Factorization Stability

The moving Laplacian and Resolvent maintenance will eventually accumulate finite-precision drift, relative to the algebraic twin constructed from (π_v, p_v). Furthermore, the fill-in effect, in the triangular surrogate, hinders incremental update for L_f/R_f.

This appendix addresses the sparse-dense storage efficiency, in joint of limited precision arithmetic. Our objective is to assert arbitrary-precision guarantee for all operators' residence, in balance with the computation effort – avoiding any superquadratic complexity blowup relative to the operator version frequency.

D.1 Sparse factor anchors

The dense path consume row buffers and reward difference per operator version in an eager fashion. Sparse path, however, should inhibit sparse LU routine, with rebase launches only for stripped, controllable drift. Thus, multiple operator versions within an interval v can retain the same sparse factor anchor at Topology Version f, applying low-rank corrections until physical compaction or certificate recovery initiates a new Topology Version.

A sparse anchor version f fixes one realized reference system. Sequential SuperLU computes a sparse LU factorization with partial pivoting and fill-reducing column permutations (Li et al., 1999; Demmel et al., 1999):

$$\Pi_f^{\text{row}} A_{\eta,f} \Pi_f^{\text{col}} = L_f R_f. \tag{55}$$

The active representation may retain a low-rank deviation from that anchor. Ordinary commits maintain its paired basis actions and core matrix. To evaluate $A_{\eta,v} Y_v = X$, first compute the anchor solve:

$$L_f B = \Pi_f^{\text{row}} X, \qquad R_f C = B, \qquad Y_0 = \Pi_f^{\text{col}} C. \tag{56}$$

We then solve the small $\ell \times \ell$ core system and apply the low-rank correction:

$$K_v Q = V_v^\top Y_0, \qquad \text{ApplyG}_v(X) = Y_v = Y_0 + Z_v Q. \tag{57}$$

Dual actions transpose the substitution order and permutations. Factor construction may also serve physical compaction during a version transition; certificate-driven reconstruction additionally repairs drift from the primitives.

Sparsity and storage stability. Suppose every action-conditioned transition row has at most b successors. The policy-induced kernel has at most $s = \min\{n, mb\}$ nonzeros per row, so $\text{nnz}(P_v) \le ns$ and $\text{nnz}(A_{\eta,v}) \le n(s+1)$. This sparsity is preserved under exact row replacement. Between anchor changes, triangular solves reuse the static factors without introducing fill. With $w_f = \text{nnz}(L_f) + \text{nnz}(R_f)$ and correction width ℓ, representation storage is

$$O(ns + w_f + n\ell + \ell^2), \tag{58}$$

plus $O(nc)$ workspace for c right-hand sides. For sparse graphs with fill-reducing permutations, $w_f = O(n)$ and $\ell \ll n$, yielding strict $O(n)$ storage complexity compared with dense representations.

Ordinary action cost and amortization. Evaluating an action $Y_v = G_{\eta,v}X$ via Equation (57) costs $O((w_f + n\ell + \ell^2)c)$. The higher costs of factor construction and reference trace evaluation are charged to initialization, physical compaction, or certificate recovery. Trace updates during ordinary commits use the exact recurrence in Equation (36).

D.2 Inverse certificate

To controllably certify numerical accuracy, we propose the *Inverse Certificate*. The certificate couples the defining primitives to the resident numerical state through one scalar residual only. Its threshold is retained at ϵ_{inv} pending later benchmarking.

For $d \in \{\mathrm{out}, \mathrm{in}\}$, write $A_v^{d,\star} = A^d(\pi_v, p_v)$ for the primitive-induced system, $\widehat{A}_v^d$ for its resident realization, and $\widehat{G}_v^d$ for the maintained inverse representation. Define

$$R_v^{d,\star} = A_v^{d,\star}\widehat{G}_v^d - I, \qquad \widehat{R}_v^d = \widehat{A}_v^d\widehat{G}_v^d - I. \tag{59}$$

The unified Inverse Certificate is

$$\rho_v = \max_{d\in\{\mathrm{out},\mathrm{in}\}} \left\{ \|R_v^{d,\star}\|_\infty, \|\widehat{R}_v^d\|_\infty \right\} \le \epsilon_{\mathrm{inv}}, \tag{60}$$

where $\|\cdot\|_\infty$ is the induced maximum absolute row-sum norm. Both system evaluations reuse each inverse-action block and feed one acceptance decision. Streaming identity columns supplies the complete residual row sums and the represented trace without retaining a dense inverse.

For a nonzero query block X, let $\widehat{Y}_v = \widehat{G}_v X$. Its source residual gives the action diagnostic

$$\varepsilon_{\mathrm{qry}}(X) = \frac{\|A_v^\star \widehat{Y}_v - X\|_F}{\|X\|_F}. \tag{61}$$

This diagnostic covers the exercised block. The full representation certificate in Equation (60) owns the trace and trust bounds. Residual-evaluation and source-assembly rounding must be included when translating these exact-algebra bounds into a floating-point implementation.

Proposition 6 (Inverse-representation precision)**.** *Let the primitive-induced P_v be row-stochastic, $n \ge 2$, and $\eta > 0$. At a checked version, suppose $\rho_v \le \epsilon_{\mathrm{inv}}$. Write $\widehat{G}_v = \widehat{G}_v^{\mathrm{out}}$ and $G_{\eta,v} = (A_v^{\mathrm{out},\star})^{-1}$. Let $\widehat{F}_v = \mathrm{tr}(\widehat{G}_v)$ and let $\widehat{\Psi}_{\eta,v}$ and $\widehat{\mathcal{U}}_{\eta,v}$ be its derivatives under Equation (11). With $d_\eta = 1 - (1+\eta)^{-2}$ and $b_\varepsilon = (1+\eta)n\epsilon_{\mathrm{inv}}/[\eta(n-1)]$, the representation and trace errors satisfy*

$$\|\widehat{G}_v - G_{\eta,v}\|_\infty \le \frac{\epsilon_{\mathrm{inv}}}{\eta}, \qquad |\widehat{F}_v - F_v| \le \frac{n\epsilon_{\mathrm{inv}}}{\eta}. \tag{62}$$

Their trust derivatives obey

$$\begin{aligned} |\widehat{\Psi}_{\eta,v} - \Psi_{\eta,v}| &\le b_\varepsilon, \\ |\widehat{\mathcal{U}}_{\eta,v}^2 - \mathcal{U}_{\eta,v}^2| &\le \frac{2b_\varepsilon}{d_\eta}, \\ |\widehat{\mathcal{U}}_{\eta,v} - \mathcal{U}_{\eta,v}| &\le \min\left\{ \sqrt{\frac{2b_\varepsilon}{d_\eta}}, \frac{2(1+\eta)}{\eta} b_\varepsilon \right\}. \end{aligned} \tag{63}$$

Since $n/(n-1) \le 2$, the trust-derivative bounds depend only on η and ϵ_{inv}, while absolute trace error scales with n. For an individual computed action with source residual $r_v^\star(X) = A_v^{\mathrm{out},\star}\widehat{Y}_v - X$, independently of the representation bound,

$$\widehat{Y}_v - Y_v = G_{\eta,v} r_v^\star(X), \qquad \|\widehat{Y}_v - Y_v\|_\infty \le \eta^{-1}\|r_v^\star(X)\|_\infty. \tag{64}$$

Proof. The stochastic resolvent is nonnegative and has row sum η^{-1}, so $\|G_{\eta,v}\|_\infty = \eta^{-1}$. Multiplying the source residual identity by $G_{\eta,v}$ gives $E_v := \widehat{G}_v - G_{\eta,v} = G_{\eta,v}R_v^{\mathrm{out},\star}$. Hence $\|E_v\|_\infty \le \epsilon_{\mathrm{inv}}/\eta$ and $|\mathrm{tr}(E_v)| \le n\|E_v\|_\infty \le n\epsilon_{\mathrm{inv}}/\eta$. The affine trace reduction gives the Ψ bound. The positive-part map is 1-Lipschitz, yielding the squared-trust bound; $|\sqrt{x} - \sqrt{y}| \le \sqrt{|x-y|}$ for $x, y \ge 0$ gives the first trust bound.

For the linear bound, recall $\kappa = (1+\eta)^{-1}$. Each eigenvalue λ of P_v satisfies $|\lambda| \le 1$, and

$$\operatorname{Re} \frac{1+\kappa\lambda}{1-\kappa\lambda} = \frac{1-\kappa^2|\lambda|^2}{|1-\kappa\lambda|^2} \ge \frac{1-\kappa}{1+\kappa}. \tag{65}$$

The trace reduction averages these terms over the $n-1$ eigenvalues remaining after removal of one Perron eigenvalue, counted with algebraic multiplicity. Thus $1 + 2\Psi_{\eta,v} \ge (1-\kappa)/(1+\kappa)$ and $\mathcal{U}_{\eta,v} \ge 1/(1+\kappa)$, including reducible kernels. Rationalizing the difference of the nonnegative square roots gives

$$|\widehat{\mathcal{U}}_{\eta,v} - \mathcal{U}_{\eta,v}| \le \frac{2b_\varepsilon}{d_\eta \mathcal{U}_{\eta,v}} \le \frac{2b_\varepsilon}{1-\kappa} = \frac{2(1+\eta)}{\eta} b_\varepsilon. \tag{66}$$

Finally, subtract $A_v^{\mathrm{out},\star} Y_v = X$ from the individual source residual and multiply by $G_{\eta,v}$. □

For $P_v = I$ and $\widehat{G}_v = (1+\delta)G_{\eta,v}$, the source residual is δI and the trace error is $n|\delta|/\eta$. Thus the n/η coefficient belongs in the trace bound. The coupled residual alone governs recovery; subsequent quantities inherit the stated bounds with their exogenous parameters.

Source-anchored system precision. Let $A_v^\star = A(\pi_v, p_v)$ denote the system induced directly by the admitted primitive owners, and let $\widehat{A}_v$ be its resident incremental realization. For the same certificate block X, set $\widehat{Y}_v = \widehat{G}_v X$ and define

$$r_v^\star(X) = A_v^\star \widehat{Y}_v - X, \qquad \widehat{r}_v(X) = \widehat{A}_v \widehat{Y}_v - X. \tag{67}$$

Subtracting these residuals gives the exact action identity

$$(\widehat{A}_v - A_v^\star)\widehat{Y}_v = \widehat{r}_v(X) - r_v^\star(X). \tag{68}$$

Consequently,

$$\frac{\|(\widehat{A}_v - A_v^\star)\widehat{Y}_v\|_F}{\|X\|_F} \le \frac{\|\widehat{r}_v(X)\|_F + \|r_v^\star(X)\|_F}{\|X\|_F}. \tag{69}$$

If $\widehat{Y}_v$ has full row rank, then

$$\widehat{A}_v - A_v^\star = (\widehat{r}_v(X) - r_v^\star(X))\widehat{Y}_v^\dagger, \qquad \|\widehat{A}_v - A_v^\star\| \le (\|\widehat{r}_v(X)\| + \|r_v^\star(X)\|)\|\widehat{Y}_v^\dagger\|. \tag{70}$$

For $X = I$, the coupled certificate gives

$$\|(\widehat{A}_v - A_v^\star)\widehat{G}_v\|_\infty \le 2\rho_v \le 2\epsilon_{\mathrm{inv}}. \tag{71}$$

Since we can control the tolerance as such $\epsilon_{\mathrm{inv}} < 1$ holds, a passing source representation residual makes $\widehat{G}_v$ invertible. Its pseudoinverse is then $\widehat{G}_v^{-1}$. The source and resident evaluations share one threshold, anchoring G to (π, p) and deriving the precision of moving A.

Source actions need not materialize P_v. For the outgoing system,

$$(A_v^\star \widehat{Y})_s = (1+\eta)\widehat{Y}_s - \sum_a \pi_v(a \mid s) \sum_{s'} p_v(s' \mid s, a)\widehat{Y}_{s'}. \tag{72}$$

Incoming actions use source-derived degrees and the corresponding transpose action. Primitive updates therefore define the reference independently of the resident system's accumulated arithmetic.

Terminal Completion uses the affine multiplier $(1+\eta)/N$ on the planning-trace error in Equation (48), with its absorbing contribution retained analytically.

For a transpose-action residual z, the corresponding dimension-free estimate is $\|\widehat{y} - G_{\eta,v}^\top x\|_1 \le \eta^{-1}\|z\|_1$. The incoming system is included in the same coupled maximum against its own defining matrix.

Scheduled checking. Certification follows completed logical versions, after all physical chunks and before RHS diffusion at checked versions. Starting from a certified v_0, successive passes give checks at $v_0, v_0 + 1, v_0 + 3, v_0 + 6, \ldots$. Recovery resets the interval to one. This inspection policy amortizes checking, with the bounds apply at checked versions.

SCHEDULED COUPLED CERTIFICATE

Require: state certified at v_0; $\epsilon_{\text{inv}} = 5 \times 10^{-4}$

$h \leftarrow 1$, $v_{\text{check}} \leftarrow v_0 + 1$
for each effective candidate version v **do**
  Freeze candidate (π_v, p_v) and coalesce its row changes
  Complete both system and inverse updates across all chunks
  if $v = v_{\text{check}}$ **then**
    Evaluate ρ_v with shared inverse-action blocks
    if $\rho_v > \epsilon_{\text{inv}}$ **then**
      Repair from (π_v, p_v) and recertify internally
      $h \leftarrow 1$
    else
      $h \leftarrow h + 1$
    end if
    Refresh trace from the certified representation
    $v_{\text{check}} \leftarrow v + h$
  end if
  Apply RHS diffusion and derive trust and priority fields
  Publish candidate and check schedule against its parent
end for

E ALGORITHMS AND IMPLEMENTATION

The planner and host divide ownership at a narrow interface, as dictated by the unified host-planner interface 2.1. The host performs environment interaction and value or control updates. The planner maintains the operator, spectral trust, and priority order.

E.1 HOST CONTRACT

The GTA$^+$-PS architecture preserves the canonical host-planner separation of legacy GTA-PS while redesigning the internal topology maintenance and priority synthesis mechanisms. In this contract, the host agent owns environment stepping, experience collection, and value parameter optimization, while the planner maintains the graph transition operator, evaluates spectral trust, and manages the prioritized frontier. Specifically, GTA$^+$-PS maintains full backward semantic compatibility with the host interface routines:

- **Transition and Policy Staging:** Exogenous model adjustments (UPDATEMODELTRANSITION) and smoothed policy updates (GTAPOLICYUPDATE) are staged non-blockingly, replacing per-transition Sherman-Morrison rank-1 updates with coalesced rank-r model update.
- **Reward and Residual Revision:** Exogenous reward shifts (REWARDUPDATE) and Bellman residual notifications (NOTIFYBACKUP) update right-hand sides and residual vectors ξ without triggering spurious topology commits.
- **Spectral Trust replacing SLEM Arnoldi Solves:** GTA$^+$-PS replaces these Arnoldi eigen-solves entirely with the exact resolvent trace recurrence (Equation (36)), computing Spectral Trust $\mathcal{U}_{\eta,v}$ in $O(n\ell + \ell^2)$ ordinary maintenance time, with precision assessed at scheduled certificate checkpoints.
- **Batched Priority Dispatch:** While legacy GTA-PS performed scalar queue polling (POPANDPRIORITIZE), GTA$^+$-PS supports batched priority extraction via Gumbel-perturbed logits (PRIORITYSCORES), decoupling asynchronous candidate selection from host value backup execution.

Table 5: Planner operations and ownership.

Operation	Owner	Effect
REWARDUPDATE	Host then planner	Refresh reward and residual state without advancing v.
UPDATEMODELTRANSITION	Host/model adapter	Stage one action-conditioned row.
COMMITMODELUPDATES	Planner	Coalesce rows and publish one effective topology change.
PRIORITYSCORES / POPANDPRIORITIZE	Planner	Read one frozen priority snapshot.
NOTIFYBACKUP	Host	Return residual and policy consequences after the host update.
REBASE	Operator core	Repair systems and inverse representations for the same admitted primitives.

E.2 COMMIT PROCEDURE

The topology commit acts as an atomic transaction boundary that incorporates staged inference updates into the live planning state. Rather than mutating operators in place, the planner constructs an isolated candidate state and validates its structural invariants against the parent snapshot before publishing, ensuring race-free concurrency and consistent Operator Version progression.

COHERENT TOPOLOGY COMMIT

Require: immutable parent snapshot S_v, complete staged changes, and admitted storage for every affected row

Ensure: complete publication or unchanged active state

1: $(\mathcal{D}_v, \Delta H_v) \leftarrow$ FREEZESTAGING(S_v)
2: $(U_v, V_v) \leftarrow$ COALESCEROWS($\mathcal{D}_v$)
3: $Z_v \leftarrow G_{\eta,v} U_v$, $W_v \leftarrow G_{\eta,v}^\top V_v$
$C_v \leftarrow (I - V_v^\top Z_v)^{-1}$
4: candidate $\leftarrow$ BUILDOPERATORCANDIDATE($v, U_v, V_v, Z_v, W_v, C_v$)
5: SCHEDULEDCERTIFYANDREPAIR(candidate)
6: REFRESHFIELDS(candidate, ΔH_v)
7: **if** $\neg$VALID(candidate) **then return** rejected
8: **return** COMPAREANDPUBLISH(S_v, candidate)

SCHEDULEDCERTIFYANDREPAIR completes any due coupled check and internal recovery before RHS diffusion, following Section D.2. VALID checks primitive validity and completed numerical work. COMPAREANDPUBLISH atomically swaps the candidate and check schedule against parent S_v, rejecting stale parents.

The version v advances once only if the topology changes. Otherwise it stays fixed, and reward-only updates bypass steps 2–3. Both directional resolvents use the coalesced shock, with the incoming degree correction given by Equation (28).

E.3 QUEUE PROCEDURE

The queue forms one priority field over a frozen eligible stream, implied as the data bundles after inference phase. Each candidate receives one independent draw, reused throughout selection of $\mathcal{B}_v$, so no state is selected twice within one batch, though overlap between batches is possible. Every selected state reads the same value status, and the host stages all Bellman mixes for the next topology commit. Setting $c_Q = 1$ recovers the max-heap pop style.

Top-k vectorized routine on GPU, is in itself, more efficient than the max-heap data structure itself. The Gumbel Trick helps us carpet additive jitters, preserving the vectorization.

FROZEN GUMBEL QUEUE BATCH

Require: ξ_v, τ_v, β_v, T_v, ε_ξ, q_{cf}, and c_Q
Ensure: at most c_Q distinct states from one version-v snapshot
1: $\mathcal{C}_v \leftarrow \{s : \xi_v(s) > \varepsilon_\xi\} \cup \mathrm{TopQ}(\tau_v, q_{\mathrm{cf}})$
2: $d_v(s) \leftarrow \xi_v(s) + \beta_v \tau_v(s)$ for $s \in \mathcal{C}_v$
3: draw independent $g_v(s) \sim \mathrm{Gumbel}(0,1)$ for $s \in \mathcal{C}_v$
4: $p_v(s) \leftarrow d_v(s) + T_v g_v(s)$ for $s \in \mathcal{C}_v$
5: $\mathcal{B}_v \leftarrow \mathrm{TopK}(\{p_v(s) : s \in \mathcal{C}_v\}, \min(c_Q, |\mathcal{C}_v|))$
6: **return** FROZENBATCH$(v, \mathcal{B}_v, p_v|_{\mathcal{B}_v})$

E.4 DDQN REPLAY LOGITS

In Stage II visual latent control, Double DQN (DDQN) serves as a canonical model-free deep reinforcement learning baseline. While standard Prioritized Experience Replay (PER) samples transitions proportionally to local temporal-difference errors ($|\delta_{\mathrm{TD}}|^{\alpha_{\mathrm{PER}}}$), this local heuristic lacks global graph diffusion awareness in structured latent state spaces.

To evaluate whether topological prioritization generalizes to deep off-policy learning, we introduce **DDQN–GTA$^+$–PER** as an experimental upgrade to the replay mechanism. This is also a showmatch for the other GTA-augmented method, described in F.2. DDQN–GTA$^+$–PER shares the identical neural architecture, minibatch budget, optimizer state, and replay capacity as standard DDQN, modifying solely the replay transition sampling distribution by fusing local TD priorities with the global topology signal under the unified Gumbel selection law.

For a common candidate set $\mathcal{C}_v$, let $w_v(c) > 0$ be the PER base priority (e.g., $|\delta_{\mathrm{TD}}(c)| + \epsilon_{\mathrm{PER}}$) and $h_v(c)$ the deterministic GTA score mapped to replay candidate c. Write $\alpha_{\mathrm{PER}} \geq 0$ for the PER exponent and $\beta_{\mathrm{PER}} \geq 0$ for the importance-sampling exponent, distinct from the directional topology coefficient α and the trust weights β_1, β_2. At queue temperature $T_v > 0$, the joint logit is

$$\boxed{\chi_v(c) = \alpha_{\mathrm{PER}} \log w_v(c) + \frac{h_v(c)}{T_v}.} \tag{73}$$

Let $\hat{c}_v$ denote the selected candidate. Normalizing these logits gives its first-draw probability

$$\pi_v(c) = \Pr(\hat{c}_v = c) = \frac{w_v(c)^{\alpha_{\mathrm{PER}}} \exp(h_v(c)/T_v)}{\sum_{u \in \mathcal{C}_v} w_v(u)^{\alpha_{\mathrm{PER}}} \exp(h_v(u)/T_v)}. \tag{74}$$

Thus PER supplies a multiplicative weight within the GTA softmax law. Equivalently, one independent canonical Gumbel per candidate yields the selection rule

$$\boxed{\hat{c}_v = \arg\max_{c \in \mathcal{C}_v} \left[h_v(c) + T_v \alpha_{\mathrm{PER}} \log w_v(c) + T_v g_v(c) \right].} \tag{75}$$

The top entries form an ordered sample without replacement. For $\beta_{\mathrm{PER}} \geq 0$, the learner uses batch-normalized weights proportional to $(|\mathcal{C}_v| \pi_v(c))^{-\beta_{\mathrm{PER}}}$. Both priority terms come from one coherent snapshot, and the Gumbel field is drawn once per selection. At $T_v = 0$, replay follows deterministic h_v order, with importance correction disabled.

E.5 INFERENCE-LEARNING RUNTIME

The inference–learning runtime operationalizes the decoupled producer–consumer lifecycle (Appendix A.5). Along the *actor* axis, parallel generation lanes execute rollouts under a frozen Reference Actor policy at Agent Version v (Appendix A.3) with isolated random sub-seeds. Along the *data stream* axis, generated transitions and environment steps assemble into bounded dispatch buffers, staging candidate experiences without mutating the active policy snapshot.

Along the *prioritization* axis, the planner ranks buffered candidates via topology-fused joint logits before dispatching ordered batches to the learner (Appendix A.4). Along the *learning cadence* axis, the learner consumes prioritized batches across discrete update bursts. Upon completing the burst

schedule, a Coherent Commit advances the model from $v \to v + 1$, publishing refreshed parameters back to the actor pool (Figure 7).

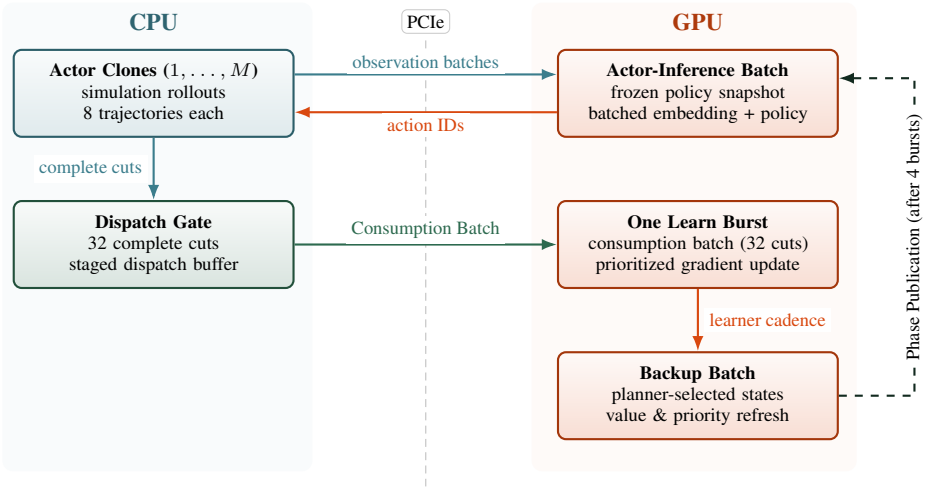

Figure 7: CPU–GPU batching sketch for model-based latent planning. GPU Packing divides ready actor requests into physical packets, whereas the sampling follows the planner's logical selection rule. Replay-only hosts use the learner batch without the state-backup branch.

E.6 Device Mapping

Device mapping operationalizes the inference–planning–learning cycle under a strict locality principle: numerical state remains resident on the device where it is computed, synchronizing across the host–device interconnect only when required by ownership transfer or external dependencies. Host orchestration retains scheduling, environment stepping, and version publication authority, while high-throughput linear algebra, planning updates, and neural optimizations remain entirely on the GPU to eliminate unnecessary memory round-trips.

Planning boundary. During planning, the CPU batches environment-observed transition updates and reward revisions for transfer to the accelerator. Policy changes derived from device-held values is pinned and staged directly on the GPU. Staged primitive rows coalesce into one low-rank shock. The device updates both realized systems and resolvents, completes scheduled coupled certificate and recovery, then applies reward or residual right-hand sides through the diffusion joint.

The planning handoff triggers upon producing a complete candidate batch. Rather than inspecting intermediate numerical state, the CPU host receives only a compact completion token to validate parent identity, allowing candidate priority arrays to pass directly to the learner on-device. Stream dependencies order device execution internally, decoupling host coordination from fine-grained linear algebraic steps.

Role transition. The learner consumes prioritized candidate batches in prescribed order, accumulating parameter gradients within private device memory. At the phase boundary, a Coherent Commit publishes references to the completed state, promoting the new model to the active frozen context without round-tripping parameters through host memory. This separation allows physical packing and inference queries to overlap within a phase while keeping logical trajectories, random streams, and update cadence strictly invariant.

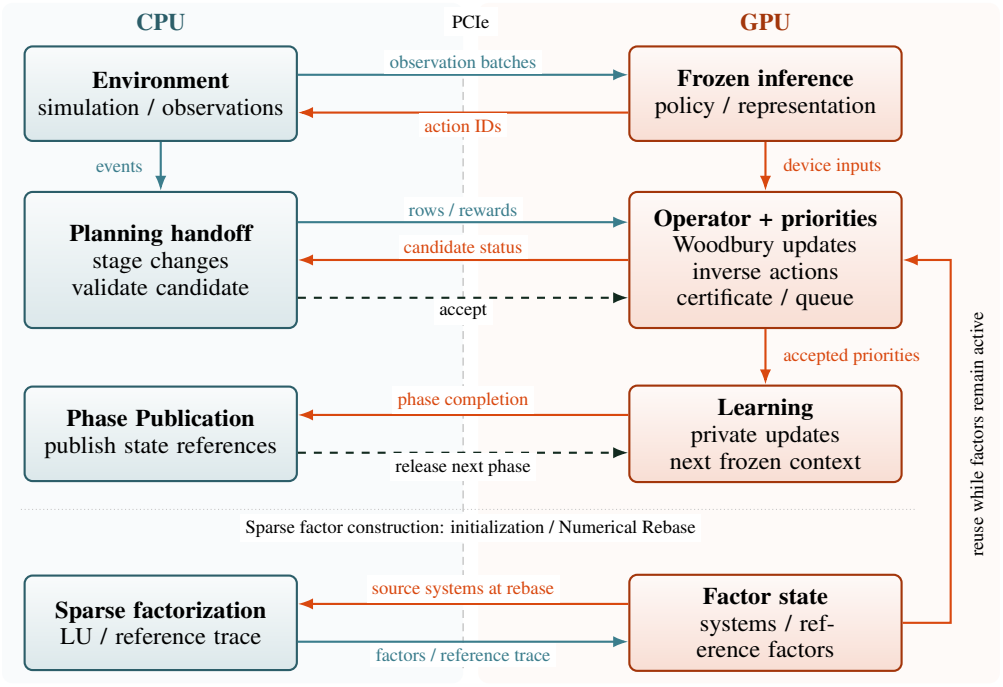

Figure 8: Algorithmic CPU–GPU mapping. Solid arrows carry data; dashed arrows carry control. The GPU planning chain retains its intermediate arrays, while CPU handoffs exchange inputs and compact completion records. Sparse factor construction occupies a separate path below the ordinary phase.

E.7 Backend conformance

Backend conformance requires both representations to preserve the same committed operator semantics. Given the same parent state and row shock, they must agree within declared tolerances on Operator Version, directed inverse actions, maintained trace, candidate publication, RHS revision, and queue ordering. Table 6 summarizes the explicit numerical guards, tolerances, and commit invariants enforced during operator maintenance.

Table 6: Commit invariants and numerical guards for operator maintenance.

Guard / Constraint	Explicit Bound	Enforcement Action
Primitive stochasticity	$\pi_v \mathbf{1} = \mathbf{1}$, $p_v^a \mathbf{1} = \mathbf{1}$	Normalize at the primitive owner; record derived mass error
Shock conservation	$V_v^\top \mathbf{1} = \mathbf{0}$	Coalesce the version-local row shock
Inverse Certificate	$\rho_v \le 5 \times 10^{-4}$ at scheduled checks	Repair and recertify from candidate primitives
Correction capacity	Physical admitted width	Chunk or compact within the v transition; preserve check schedule
Spectral margin	Diagonal shift $\eta > 0$	Bound resolvent norm $\|G_{\eta,v}\|_\infty = \eta^{-1}$
Parent atomicity	Exact snapshot identity	Reject stale candidates or incomplete scheduled recovery

Dense and sparse representations implement the same directed inverse actions and commit equations. Backend choice changes storage and numerical execution; it does not change the residual, directional mixture, candidate eligibility, or sampling law. Table 7 compares the two backends across data structures, storage footprints, and execution paths.

Table 7: Dense and sparse operator backends differences.

Aspect	Dense backend	Sparse backend
Data structure	Dense systems, inverses, and update blocks	CSR systems and factors; permutations; dense correction blocks
Representation	Explicit outgoing and incoming inverses	Frozen sparse LU factors plus paired corrections
Initialization	Invert systems; sum outgoing diagonal	CPU factorization and trace solves; device transfer
Inverse actions	Matrix–block products	Permuted triangular solves plus corrections
Commit sequence	Coalesce $\rightarrow$ paired products $\rightarrow$ core solve $\rightarrow$ inverse/trace update $\rightarrow$ publish	Coalesce $\rightarrow$ factor actions $\rightarrow$ extend corrections/core $\rightarrow$ trace update $\rightarrow$ publish
Incoming update	Degree-correct inverse update	Degree-correct factor blocks
RHS change	Multiply by current inverse	Apply factors and corrections
Storage	$O(n^2)$ plus workspace	Factor fill $+ O(n\ell + \ell^2)$ plus workspace
Numerical Rebase	Repair systems, inverses, and trace	Repair systems and factors; refresh trace and clear corrections

The storage comparison depends on more than the sparsity of P_v. Although the sparse backend avoids an explicit inverse, elimination can introduce factor fill, and accumulated corrections grow with ℓ. Its memory advantage therefore depends on both quantities, together with the workspace needed to apply the factors to each RHS block.

F Experiment Apparatus

The empirical evaluation is designed to determine whether graph-topology-augmented planning extends computational viability, theoretically sound, and sample-efficient when scaled far beyond the small, stationary toy problems where GTA-PS was originally introduced. Specifically, our experiment apparatus targets three primary scientific objectives:

1. **Planning efficiency and non-stationary recovery (Stage I):** Quantify value-error decay and Bellman update allocation under controlled, exogenous reward shifts, establishing whether incremental Woodbury resolvent maintenance accelerates planning convergence across large-scale factored graphs.
2. **Spectral trust and queue fairness:** Evaluate whether trace-derived spectral trust ($\mathcal{U}_{\eta,v}$) coupled with Gumbel queue perturbation prevents priority starvation, ensures fair state exploration, and stabilizes stochastic planning across non-communicating or weakly-connected transition topologies.
3. **End-to-end throughput and visual transfer (Stage II):** Establish whether the decoupled asynchronous actor–planner–learner architecture sustains high bare-metal container throughput and superior sample efficiency when deployed on 512-state visual latent MDPs derived from high-dimensional pixel observations (Atari Pong and 3D ViZDoom navigation).

To isolate logical planning efficacy from representation fitting and physical device overheads, the evaluation is structured into two complementary phases: Stage I isolates planning backends on fixed tabular models, while Stage II benchmarks end-to-end visual agents under a strict five-million decision budget.

F.1 Environments and domain scaling matrix

Inherited GTA-PS (Pham & Dam, 2026) evaluated graph prioritization strictly on small, synthetic toy MDPs with maximum state counts of $N_{\max} \leq 200$, where dense resolvent recalculation or explicit matrix inversion $O(n^3)$ remained computationally tractable. In contrast, our benchmark suite establishes a structured domain rubric spanning two reception variation (tabular-vision) and

three distinct scale tiers (512, 1k, 4k states). This represents a $2.5\times$ to $22.5\times$ expansion in graph dimension over the largest legacy GTA-PS test-ground, ablating over both sparse-structure scaling and continuous-to-discrete representation abstractions.

Table 8: Primary evaluation environments and domain scaling matrix across tabular and visual regimes.

Stage	State Scale	Task	Graph / Observation Structure	Factor vs. Legacy
II	$N_{\max} = 512$	Atari Pong (Bellemare et al., 2013)	2D pixel control; frozen tokenizer encoder; 512 discrete codebook states	$2.56\times$
II	$N_{\max} = 512$	ViZDoom My Way Home (Kempka et al., 2016)	3D first-person navigation; pretrained CNN backbone; 512 spatial centroids	$2.56\times$
I	1024 ($n \sim 10^3$)	SysAdmin (Guestrin et al., 2003)	10-machine network (1,024 states); cascading failures; localized reward inversion	$5.12\times$
I	1280 ($n \sim 10^3$)	CargoTaxi (Dietterich, 2000)	5×5 grid (1,280 realized states); multi-passenger routing; destination shifts	$6.40\times$
I	4096 ($n \sim 4$k)	SysAdmin (Guestrin et al., 2003)	12-machine network (4,096 states); high-dimensional directed transitions	$20.48\times$
I	4500 ($n \sim 4$k)	CargoTaxi (Dietterich, 2000)	6×5 grid (4,500 realized states); multi-cargo destination relocation	$22.50\times$

Visual representations. For Pong, the visual backbone is the tokenizer encoder and codebook from the released IRIS `Pong.pt` checkpoint (Micheli et al., 2023).[1] For My Way Home, we use the visual encoder and observation normalizer from the APPO checkpoint published by `MattStammers`, which was trained with Sample Factory 2.0.[2] We retain these representation components and initialize the experimental control modules separately. The ViZDoom encoder inherits control-trained features from its source checkpoint. Where a static abstraction is used, its centroids are fitted for the target task; the representation-update policy is specified separately.

F.2 Baselines

Stage I. Stage I consists of three control methods of Standard PS, GPS-H2 and GTA-PS; in which the former is vanilla formulation of (Dayan, 1993), the latter is vanilla formulation of (Pham & Dam, 2026), and the middle uses the General Prioritized Sweeping of (Wingate & Seppi, 2005). Noticeably, the prioritization score of GPS-H2 is:

$$\text{priority}_{\text{GPS}-\text{H2}}(s) = \xi(s) + \omega V(s) \tag{76}$$

with the parameterization of ω anneals the integral value, avoiding overwhelming stale value in nonstationary setting. The canonical value of ω, in the magitude sense, is $\omega^\star = \gamma$, normalizing one-step residual and infinite-step value breadth.

The other two new variants are of GTA$^+$-PS, codename $\text{GTA} - \text{Trust}$ and $\text{GTA} - \text{full}$ variants whose variance-scaled Gumbel jitter is respectively disabled and enabled. Each comparison shares the MDP instance, value initialization, exogenous shifts, and stopping boundary.

Stage II. Stage II compares five control methods. The three control methods is respectively VaST-PS, DDQN-PER and NEC. VaST supplies model-based latent planning (Corneil et al., 2018). DDQN-PER combines Double Q-learning with prioritized replay (van Hasselt et al., 2016; Schaul et al., 2016), and NEC provides an episodic nonparametric comparator (Pritzel et al., 2017). Our augmentation, VaST-GTA-PS and DDQN-GTA-PER, uses the latest formulation of 4.2 and E.4

[1] `https://huggingface.co/eloialonso/iris`, checkpoint revision `c0539e9`.

[2] `https://huggingface.co/MattStammers/vizdoom_my_way_home`, checkpoint revision `18051bc86a9906e4ef3ad2084c157c3a2e724970`.

respectively. Every within-task comparison uses the same pretrained backbone weights, observation preprocessing, State Capacity, and decision ceiling.

F.3 Seed partitions

Each Stage I suite reserves 10 paired Test seeds. Competing planners share the environment instance, initialization, and shift sequence. Actor Randomness uses a method-specific stream, which prevents GTA jitter draws from advancing any baseline stream.

The Stage II protocol sets a ceiling of five million cumulative environment decisions (5M), including source pretraining, and assigns one Test-seed control run to each method. Read-only evaluation uses a fixed cohort. Pong records complete policy matches, and My Way Home evaluates 32 episodes at each milestone. The resulting interval measure variation within that evaluation cohort.

F.4 Environment boundaries

Stage I compares planners under matched scheduled planning allowances: two Poisson-scheduled reward shifts, follow by the exhaustive run until 80% error reduction is achieved in the right tail. Initialization, queue work, topology refresh, operator routine, and backups are all captured within the boundary; wall time is observed as container level.

Stage II evaluates visual latent control under a cumulative decision-censored protocol targeting an approximate ceiling of five million (5M) environment decisions. Because experience generation operates across parallel actor clones with complete trajectory dispatches, execution terminates at the first phase boundary that satisfies the cumulative budget, recording the exact reached decision count without in-flight trajectory truncation.

Special note. Pong adds auxiliary reward of *0.02* per paddle contact only during pretraining to warm-start learning. Continuation uses task rewards with zero contact augmentation, while historical replay retains the rewards observed when its trajectories were collected. The reference checkpoint preserves learner state across this explicit reward-policy handoff.

F.5 Metrics and intervals

To operationalize and validate the scientific objectives established in Appendix F, our empirical evaluation establishes a three-tier **Metric Hierarchy** along with complete run provenance (Table 9). Together, they provide the necessary observational basis to certify planning performance, control applicability, and bare-metal scalability:

1. **Task Completion:** Evaluates the environment-set task outcomes and convergence speed, demonstrating *task superiority* of our methods.

2. **Control Substrate:** Measures internal operator, learner, and policy dynamics, capturing active state coverage, priority gain, objective loss, policy momentum, numerical accuracy, and spectral trust health ($\mathcal{U}_{\eta,v}$), certifying *control longevity* and anti-starvation fairness across long-horizon planning regimes.

3. **Bare-Metal Utilization Metrics Tier (Scalability Certification):** Quantifies physical system efficiency on the GPU residence apparatus, measuring true Container Throughput in real wall time, GPU kernel activity and affinity, host–device synchronization idleness, and memory footprint, demonstrating *bare-metal hardware utilization and scalability*.

Stage I scalar summaries report the mean over the fixed 10-seed set with a declared 95% interval. Stage II may bootstrap paired differences over the common evaluation cohort, provided that the interval is identified as conditional on one trained policy. Duration aggregates use SGM10 for positive elapsed times. Returns and success rates retain their native statistics.

Table 9: Metric Hierarchy tiers, measured quantities, and reporting locations.

Tier	Measures	Reported in
Task Metrics	Value error decay and non-stationary recovery in Stage I; Pong episodic return; Goal Success Rate, Navigation Performance, training duration, and milestone success in Stage II	Section 6
Control SubstrateMetrics	Active states, cumulative reward, objective loss, priority gain, numerical accuracy, spectral trust $\mathcal{U}_{\eta,v}$, and learner trajectory momentum	Appendix G
Bare-Metal Utilization Metrics	Container Throughput (wall-clock), timing decomposition, GPU affinity, GPU activity, host–device transfer overhead, and synchronization idleness	Appendix G
Provenance (Run Manifest)	Software git revision, representation checkpoints, compute hardware, random seed partitions, and termination criteria	Appendix H

F.6 Hyperparameters

We declare the crucial shared protocol invariants, numerical tolerances, pipeline dimensions, lifecycle schedules, and A100 container resource envelopes for both Stage I and Stage II in Table 10. Method-specific exploration schedules and baseline optimizer configurations follow their standard published references and are provided in the supplementary repository.

Table 10 specifies the common comparison protocol. The stationary variance interpretation and spectral trust bounds require the communication assumptions in Appendix C, holding independently of whether dense or sparse GPU execution backends are active.

Table 10: Crucial shared protocol hyperparameters, lifecycle invariants, and container resource envelopes across the Stage I and Stage II benchmark suites.

Quantity / Invariant	Value / Operational Bound
Stage I State scale matrix	1k ($n \in [1024, 1280]$) and 4k ($n \in [4096, 4500]$) states
Stage I Planning allowance	Poisson mean 2,000 - total limit 12k) for 1k environments. mean 5,000 (limit 30k) for 4k environments
Stage I Evaluation seeds	10 paired Test seeds per comparison suite
Stage I Backup batch	8-state queue cutoff; 32 counted backups per learning
Stage II State Capacity	Maximum 512 active discrete latent states
Stage II Decision ceiling	5M cumulative decisions (1M pretraining +4M continuation)
Stage II Actor Clones	4 parallel actors, 8 trajectories/actor (32 per dispatch)
Stage II Learning Volume	128 trajectories (4 burst cohorts) under frozen Agent Version v
Stage II Pong contact augmentation	$+0.02$ auxiliary reward in pretraining; 0.0 in continuation
Stage II Read-only evaluation	Fixed 32-seed cohort; milestones at 25K, 50K, 100K, 250K, 500K, 750K, 1M, 2M, 3M, 4M, 5M decisions
Control discount γ	0.98
Resolvent diagonal η	0.05 (bounds $\|G_{\eta,v}\|_\infty \leq 20$)
Inverse certificate tolerance	$\epsilon_{\text{inv}} = 5 \times 10^{-4}$
Woodbury rank capacity ℓ	≤ 32 columns per coalesced update shock
Profile resource	4 vCPUs, 10 GB GPU VRAM, *unbounded* CPU RAM

G Additional Results

This section is reserved for the full tables, trajectories, profiles, ablations, and failure analyses that underlie Section 6 once the evaluation suite closes.

H Additional Discussion

H.1 Hyperparameter Selection

Baseline hyperparameters follow published canonical defaults for VaST, DDQN-PER, and NEC. For GTA$^+$-augmented methods, hyperparameters are tuned on reserved validation seeds disjoint from the test evaluation cohort. Spectral margin η and directional weight α are selected via grid search on Stage I tasks to minimize Bellman residual norm. Stage II inherits frozen visual encoder backbones and shared batch budgets, tuning queue temperature scaling offline to preserve stable softmax entropy without distorting priority ranking.

H.2 Evaluation Management

To guarantee robustness against container timeouts and preemption on remote accelerators, the runtime executes periodic hash-verified snapshotting at completed phase boundaries. If a container reaches its wall-time ceiling, execution halts cleanly; subsequent runs resume from the verified provisional snapshot, restoring exact parent versions, replay indices, and random streams without duplicating environment decisions or corrupting metric provenance.

H.3 Limitations

The efficiency of GTA$^+$-PS depends on the structured variation rush. For a too rapid reception drift, our model might fail operationally. The knowledge of jitters and queue statistics' elasticity is still underexamined. The variance interpretation of Spectral Trust nonetheless strictly requires a communicating stationary kernel, and or Terminal Completion fix are just as an algebraic surrogate. Finally, our tabular study evaluates transfer under fixed visual abstractions and paired execution profiles, demonstrating conditional planning efficiency rather than end-to-end incremental learning.

H.4 Declaration

Our paper, attached with the supplement, conceive truthful and sufficient material for third-party reproduction:

- Appendices A–D provide the formal assumptions and derivations.
- Appendix E specifies the executable procedures and backend invariants.
- Appendix F records the environments, baselines, seed partitions, timing boundaries, metrics, batching rules, and hyperparameters.
- Section 6 defines the comparison boundaries.

Together, these materials specify the premises, implementation contract, and experimental realization needed to reproduce the reported claims.