LOOPREPLAY: RELIABILITY- AND DIAMETER-ADAPTIVE SEQUENCE REPLAY FOR RAPID RECOVERY AFTER REWARD RELOCATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Under conditions where rewards are sparse and spatially distributed, three failures arise in sequence replay. Sweeping stalls due to iterative states, low-confidence stochastic edges contaminate propagation values, and a fixed replay horizon becomes mismatched with the scale of the learned graph. LOOPREPLAY introduces an event-based controller. This controller regulates access based on reward outcomes and signed prediction errors, enforces non-cyclic scans prioritized by empirical transition confidence, uses residual depletion as a termination condition, and determines propagation depth based on the diameter of the learned graph. All rules were fixed prior to running the validation experiment, which consisted of 12 conditions. This experiment covered three topologies (ring, bridge, cross), two sizes (9, 17), two slip probabilities (0.08, 0.22), and a balanced reward-shifting geometry. When compared to the stronger of prioritized replay and Dyna under each condition, LOOPREPLAY improved the average success AUC by $+0.071$ (95% hierarchical confidence interval $[0.024, 0.126]$) at matched replay compute per online step. The gain arises after the reward moves, where post-relocation success rose from 0.72 to 0.88. It maintained a difference of within 0.05 AUC from the best baseline across all 12 conditions, and it also outperformed standard Dyna-Q, prioritized sweeping, and backward trajectory replay. The completion rate for backward sweeps exceeded 95% across all conditions, and the selection rate for low-confidence edges was less than 10%. The same rule improved all six conditions at the unseen size of 25. These results indicate that sequence reliability and the learned structural scale are joint control variables in replay-driven credit assignment.

1 INTRODUCTION

Sequence replay propagates sparse rewards through saved trajectories. Since a single replay event carries new value along an ordered transition path, it can concentrate credit assignment in areas that transition-based prioritization or model-based rollouts cannot adequately address. This propagation capability comes with a control burden. A replay system must decide when to start a sweep, in which direction to scan, which edges to trust and include, how far to propagate value, and how to isolate old rewards after a task change.

The movement of sparse rewards makes these decisions observable. Three failure modes with measurable characteristics emerge. First, the replay sweep revisits stationary or iterative states, exhausting the budget and resulting in an incomplete reverse sweep with an advancing-edge fraction of nearly zero. Second, when large TD residuals occur on infrequently observed stochastic edges, the sequence shifts toward a less reliable structure, and the selection rate of low-confidence edges increases. Third, a fixed horizon leads to insufficient propagation in large graphs and excessive propagation in small graphs, introducing a size-horizon interaction in the success AUC while increasing exposure to weak edges in small graphs.

LOOPREPLAY assigns one control operator to each failure mode and adds one invariant, context isolation, that keeps obsolete rewards out of replay. Reward outcomes and signed prediction errors control access to events and determine the scanning direction. A context identifier isolates reward changes. Acyclic scanning eliminates stationary transitions and repeated states. Empirical confidence

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.

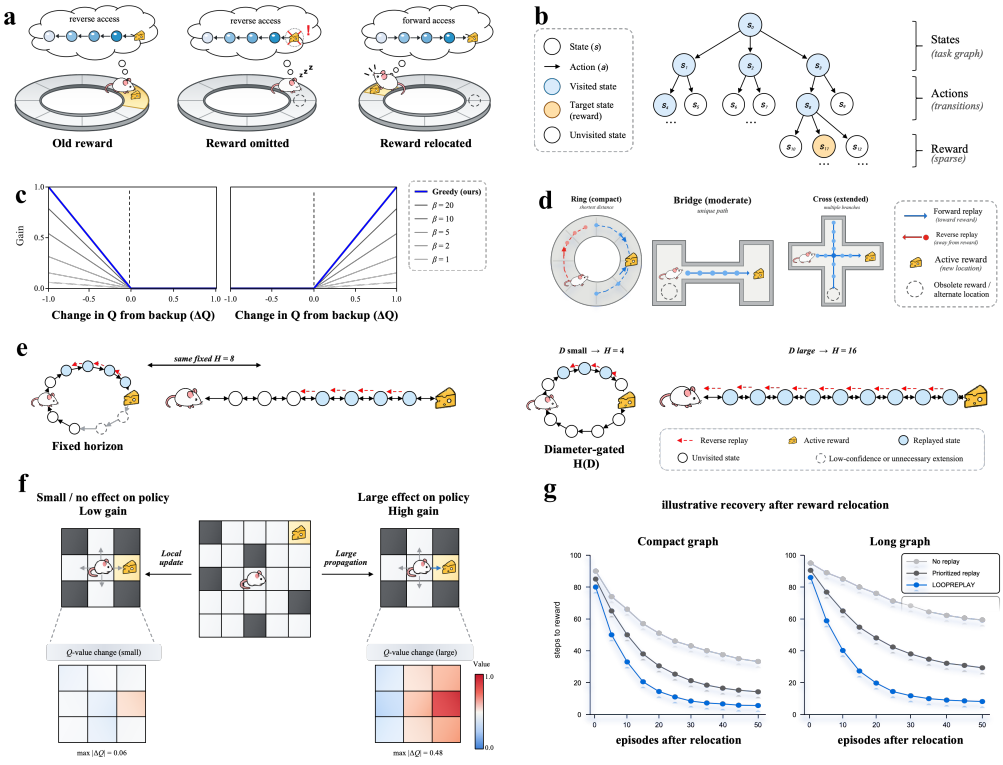


Figure 1: **Replay after a reward moves.** (a) Reverse access at the old reward, reward omission, and forward access after relocation. (b) Task graph with states, transitions, and a sparse reward. (c) Gain of a backup as a function of the change in Q. (d) Ring, bridge, and cross topologies with forward and reverse replay. (e) A fixed horizon versus a diameter-gated horizon $H(D)$. (f) Local update versus large propagation after a backup. (g) Illustrative recovery after reward relocation on compact and long graphs.

determines the ranking of candidate edges. Residual termination ends exhausted sweeps. The learned graph diameter determines both the replay horizon and the replay interval, adjusting the nominal planning capacity according to the structural scale.

This study makes three contributions. (1) Through component interventions, we demonstrate the relationship between acyclic advancement and reverse-sweep completion rates, as well as the relationship between confidence weighting and the suppression of low-confidence edges. For each operator, we validated the internal metrics to be targeted prior to external evaluation. (2) Across 12 held-out conditions, the diameter-calibrated LOOPREPLAY outperforms prioritized replay, Dyna, Dyna-Q, prioritized sweeping, and backward trajectory replay at matched replay compute, and stays within 0.05 AUC of the best conventional baseline in every condition. (3) Fixed-horizon scaling experiments reveal the size-depth interaction resolved by the diameter rule, and the rule transfers unchanged to unseen graph sizes. Hippocampal replay serves as the motivation for the computational operators. Controlled stochastic navigation tasks make the effect of each operator directly measurable.

2 RELATED WORK

In hippocampal replay, sequenced reactivation occurs during rest and deliberation (Wilson & McNaughton, 1994; Foster & Wilson, 2006; Diba & Buzsáki, 2007). Rewards modulate the frequency and content of replay (Singer & Frank, 2009), and future-oriented sequences trace paths leading to memorized goals (Pfeiffer & Foster, 2013). Reverse replay is selectively modulated by changes in reward (Ambrose et al., 2016), and replayed trajectories can extend over long paths or combine segments never traversed as a whole (Davidson et al., 2009; Gupta et al., 2010). Value-based memory theories involve expected utility to formalize selective access (Mattar & Daw, 2018). Recent compu-

tational models link task-optimized recursive dynamics and predictive cognitive maps to replay-like sequences (Stachenfeld et al., 2017; Bono & Clopath, 2023; Jensen et al., 2024). These findings provide a computational motivation for preserving order, accessing based on outcomes, and making sequence content measurable.

Model-based reinforcement learning provides a set of comparable baselines. Dyna alternates between online learning and sampling from an empirical model (Sutton, 1990). Priority sweeping utilizes the structure of subsequent states to concentrate backups on large value changes (Moore & Atkeson, 1993; Peng & Williams, 1993; van Seijen & Sutton, 2013). Priority-based experience replay ranks saved transitions based on estimated learning values (Schaul et al., 2016). Adaptive rollout (Feinberg et al., 2018) and distributed recursive agents (Kapturowski et al., 2019) also extend the replay framework. Replaying stored experience in temporally backward order was proposed early (Lin, 1992) and has been revisited through episodic backward updates and learned backtracking models (Lee et al., 2019; Goyal et al., 2019), while hindsight relabeling and return decomposition address sparse credit assignment without sequence control (Andrychowicz et al., 2017; Arjona-Medina et al., 2019; Harutyunyan et al., 2019). Planning depth trades propagation against model error (Jiang et al., 2015; Janner et al., 2019), and count-based confidence is a standard treatment of uncertain transitions (Strehl & Littman, 2008). Appendix A extends this comparison. LOOPREPLAY addresses control problems that are orthogonal to these. Specifically, it addresses how to progress through replay sequences on uncertain learned structures, how to preserve the reward context, and how to calibrate depth according to the graph.

3 LOOPREPLAY

3.1 TASK, MEMORY, AND ONLINE UPDATES

The environment is a stochastic grid graph with 4 actions, sparse terminal rewards, and a single reward position shift between episodes. Each transition to be preserved is represented as follows.

$$e_i = (s_i, a_i, r_i, s'_i, z_i, \delta_i, n_i, \ell_i), \tag{1}$$

where z_i is the estimated reward context, δ_i is the online TD error, n_i is novelty, and ℓ_i is the loop closure metric. The same tabular Q-learning policy is used for both online and replay updates.

$$\delta_i = r_i + \gamma \max_a Q(s'_i, a) - Q(s_i, a_i), \qquad Q(s_i, a_i) \leftarrow Q(s_i, a_i) + \alpha\, \delta_i. \tag{2}$$

Under this common update rule, experimental variables are limited to sequence selection and plan utilization.

The experience graph maintains a list of preceding states and a list of succeeding states for each context. Edge reliability is defined as the ratio of observed transition counts for competing outcomes arising from the same state-action pair. When a reward is no longer observed in a state where it was previously present, the context identifier is advanced, and transitions in the current episode are assigned to a new context. Even when connectivity is reused across contexts, reward labels remain localized to the original context to which the transition belongs.

3.2 EVENT ACCESS, DIRECTION, AND HIGH-CONFIDENCE SCANNING

The priority of each candidate transition is determined by the following equation.

$$p_i = |\delta_i|\, c_i + 0.5\, \ell_i + 0.75 \max(r_i, 0), \tag{3}$$

where c_i is the empirical confidence. Candidates are drawn from the 512 most recent transitions of the active context; whenever a rewarded transition is present, the sweep is seeded at the highest-priority rewarded transition. Otherwise replay begins when the maximum priority exceeds a fixed threshold. Positive rewards, sufficiently positive prediction errors, and value decreases due to reward loss trigger backward scanning. All other eligible events trigger forward scanning.

Acyclic advancement is enforced in all sweeps. Static transitions and revisited states are excluded. Backward replay traces preceding states in order of confidence, while forward replay traces succeeding states in order of confidence. Structural fallback reuses connectivity from other contexts and admits only edges without reward or terminal labels, so an obsolete reward cannot enter the active sequence by construction. A backward sweep runs to the full horizon H; a forward sweep stops once the mean absolute residual falls below 0.025 after at least two edges.

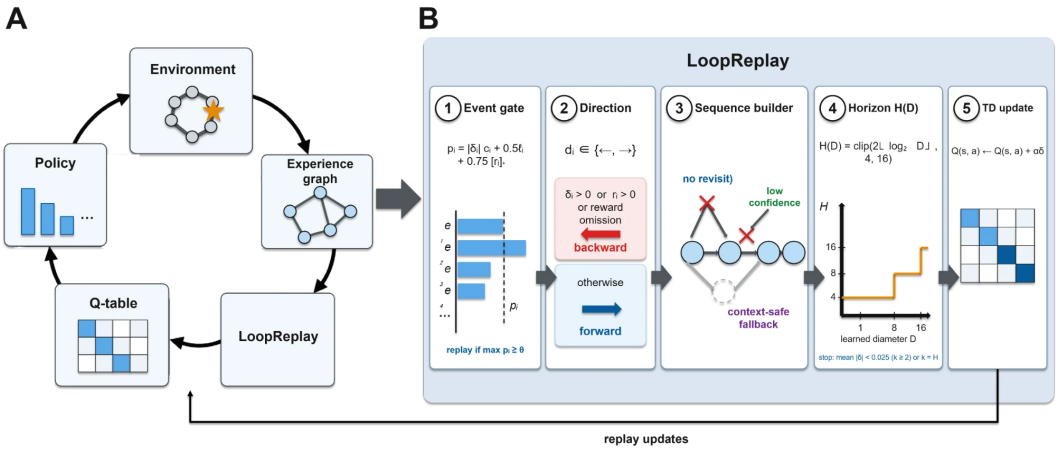


Figure 2: **LOOPREPLAY in the learning loop.** **(A)** The agent acts in the environment, stores transitions in a context-indexed experience graph, and LOOPREPLAY writes replay backups into the same Q-table that drives the policy. **(B)** One replay event: (1) the event gate starts replay from the priority in Equation 3; (2) positive mismatch, reward, or reward omission selects a backward sweep, otherwise a forward sweep; (3) the sequence builder skips revisited states, ranks candidates by confidence, and falls back to context-safe edges; (4) the learned diameter sets the horizon $H(D)$; (5) each selected edge receives the shared TD update.

Algorithm 1 LOOPREPLAY

Require: threshold θ, tolerance ϵ

for $t = 1, 2, \ldots$ **do**
  observe (s, a, r, s') and update Q ▷ Eq. (2)
  if ω **then** $z \leftarrow z + 1$
  $\mathcal{G}_z \leftarrow \mathcal{G}_z \cup \{(s, a, s')\}$
  $H \leftarrow H(D(\mathcal{G}_z))$ ▷ Eq. (4)
  $i \leftarrow \arg\max_j p_j$ ▷ Eq. (3)
  if $p_i \geq \theta \ \vee \ r_i > 0 \ \vee \ \ell_i = 1$ **then**
    $d \leftarrow$ BWD **if** $r_i > 0 \vee \delta_i > 0 \vee \omega$ **else** FWD
    $u \leftarrow s_i$, $V \leftarrow \{u\}$
    for $k = 1, \ldots, H$ **do**
      $C \leftarrow \mathcal{N}_z^d(u) \setminus V$
      if $C = \emptyset$ **then** $C \leftarrow \tilde{\mathcal{N}}_{\neq z}^d(u) \setminus V$
      if $C = \emptyset$ **then break**
      $e \leftarrow \arg\max_{e' \in C} |\delta_{e'}| \, c_{e'}$
      $Q(e) \leftarrow Q(e) + \alpha \, \delta_e$
      $u \leftarrow \text{next}(e)$, $V \leftarrow V \cup \{u\}$
      if $d =$ FWD $\ \wedge \ k \geq 2 \ \wedge \ \overline{|\delta|} < \epsilon$ **then break**

3.3 DIAMETER-BASED DEPTH CALIBRATION

Experiments with a fixed horizon reveal that graph scale is a missing key variable. LOOPREPLAY computes the exact undirected diameter D of the learned transition graph (self-transitions excluded) and sets the horizon by

$$H(D) = \text{clip}\Big(2^{\lfloor \log_2(\max(D,1)) \rfloor}, \, 4, \, 16\Big) . \tag{4}$$

The replay interval is equal to $H(D)$, so the nominal replay rate stays at one update per online step at every scale, the same rate as the baselines.

Figure 2 shows how these operators sit inside the learning loop, and Algorithm 1 gives the full procedure. There, ω marks a reward omitted at a previously rewarded state, $\mathcal{N}_z^d(u)$ denotes the

Table 1: **Design requirements and falsification metrics.** Each of the three failure modes is mapped to a control operator and a target metric. The target metrics allow the contribution of operators to be ruled out independently of aggregate performance.

Failure mode	Controller operator	Falsification signature
Sweep stagnation	Acyclic advancement	Reverse-sweep completion rate, advancing-edge fraction
Low-reliability edges	Confidence ranking	Low-confidence selection rate, high-noise AUC
Scale mismatch	Diameter-calibrated horizon	Size×horizon interaction, compact-graph exposure
Invariant: context-safe fallback keeps obsolete rewards out of replay (recovery after relocation)		

predecessors ($d = \text{BWD}$) or successors ($d = \text{FWD}$) of u in context z, and $\tilde{\mathcal{N}}^d_{\neq z}(u)$ denotes the same edges from other contexts with reward and terminal labels removed.

4 Experimental Design

4.1 Component interventions

Each intervention targets a single failure mode. Context reassignment verifies the isolation of old rewards. Acyclic advancement targets incomplete backward sweeping. Reliability weighting targets the selection of stochastic edges. For a fixed horizon $H \in \{4, 8, 16\}$, we verify the interaction with the structural scale. The diameter rule (Equation 4) and all evaluation criteria were fixed before the confirmation matrix was run on held-out seeds.

4.2 Final confirmation matrix

In the verification matrix, for each of the ring, bridge, and cross topologies, combinations of sizes 9 and 17 and slip probabilities of 0.08 and 0.22 were used. The geometry of reward shifts was balanced across 12 conditions. For each condition, 200 episodes were run using 10 held-out seeds, with rewards shifted at the 100th episode. The primary comparison includes prioritized replay using absolute TD error, Dyna using an empirical model, a fixed-horizon ablation of LoopReplay, and the full diameter-calibrated LoopReplay. Standard Dyna-Q, prioritized sweeping (Moore & Atkeson, 1993), and backward trajectory replay (Lin, 1992; Lee et al., 2019) are evaluated on the same cells and seeds (Appendix D).

The primary outcome measure is success AUC (the average binary success rate across 200-episode trajectories). The unit of inference within each condition is the seed. Hierarchical 95% confidence intervals are obtained by simultaneously resampling the task conditions and their corresponding seeds. In the primary comparison, the stronger of the conventional baselines is subtracted for each condition. Secondary measures are the gain over the fixed-horizon ablation, condition coverage, replay updates per online step, and the mechanism metrics of Table 1. The complete protocol and constants are presented in Appendix C.

5 Results

5.1 Performance and planning efficiency

LoopReplay improved the average AUC by $+0.071$ compared to the strongest conventional baseline, with a hierarchical 95% confidence interval of $[0.024, 0.126]$. The improvement over the fixed-horizon ablation is $+0.024$ AUC. The effect was positive in 10 out of 12 conditions, and the difference from the best baseline was kept within 0.05 AUC in all 12 conditions. The maximum effect per condition was $+0.274$ (high noise, size 17 bridge), and the minimum effect was -0.032 (low noise, size 17 bridge).

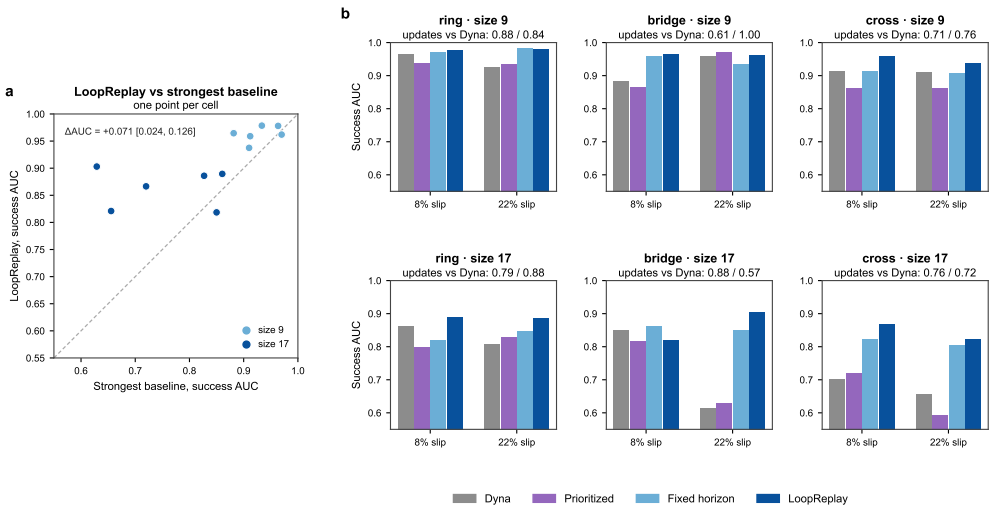


Figure 3: **Held-out validation.** Performance and planning efficiency by method based on success AUC and replay update ratio, overall hierarchical confidence intervals, averages by topology, effects by condition, and the full matrix of topology×size×noise.

The gain comes from faster reassignment of credit. All methods reach the same acquisition performance before relocation (AUC 0.945–0.951); after relocation, LOOPREPLAY reaches 0.877 against 0.722 for the best conventional baseline, and 0.648 against 0.423 in the first 25 episodes. It does so at matched compute: replay updates per online step were 0.97 times those of the strongest baseline (geometric mean), and per episode 0.78 times, because recovered agents reach the goal sooner. The reverse-sweep completion rate exceeded 95% and the low-confidence edge selection rate stayed below 10% in every condition (Figure 3).

Stronger planners do not close the gap. LOOPREPLAY outperformed standard Dyna-Q by $+0.048$ AUC ($[0.013, 0.087]$), prioritized sweeping by $+0.046$ ($[0.007, 0.090]$), and backward trajectory replay by $+0.136$ ($[0.074, 0.206]$), at equal replay budgets (Table S2). Performance was also insensitive to the constants of Equation 3 and the stopping rule: across eight perturbations, mean AUC ranged from 0.900 to 0.917, above every baseline.

5.2 COMPONENT INTERVENTIONS AND TARGET FEATURES

Acyclic advancement altered the targeted internal metrics. In the low-noise validation, the AUC with acyclic advancement was 0.900, exceeding the 0.737 for prioritized replay. The corresponding effect was $+0.163$ (95% confidence interval $[0.053, 0.278]$). The completion rate for backward sweeps rose from approximately 5% to over 99%. Under high-noise conditions, sweep completion alone could not restore performance, and transition reliability was identified as the remaining cause of failure.

The confidence gate addressed this residual failure. In the high-noise 30-seed verification, confidence weighting outperformed the unweighted acyclic controller by $+0.084$ AUC (confidence interval $[0.047, 0.126]$). The effect under low-noise conditions was $+0.067$ (confidence interval $[0.005, 0.131]$). The selection rate for low-confidence edges dropped to 3.5% under low-noise conditions and 6.3% under high-noise conditions. The completion rate remained above 99%, and there were zero omissions. Through various variants of the preceding context processing and stopping rules, we separated the operators responsible for safety from those responsible for performance. All results are shown in Appendix F.

5.3 CALIBRATION OF EFFECTIVE REPLAY DEPTH BY STRUCTURAL SCALE

In the fixed-horizon experiments, we combined ring graph sizes of 9, 13, 17, and 21 with horizons of 4, 8, and 16. The difference in AUC between Horizon 16 and Horizon 4 increased by 0.0061 per unit of size (bootstrap confidence interval $[0.0005, 0.0117]$; Figure 5a). The best measured horizon

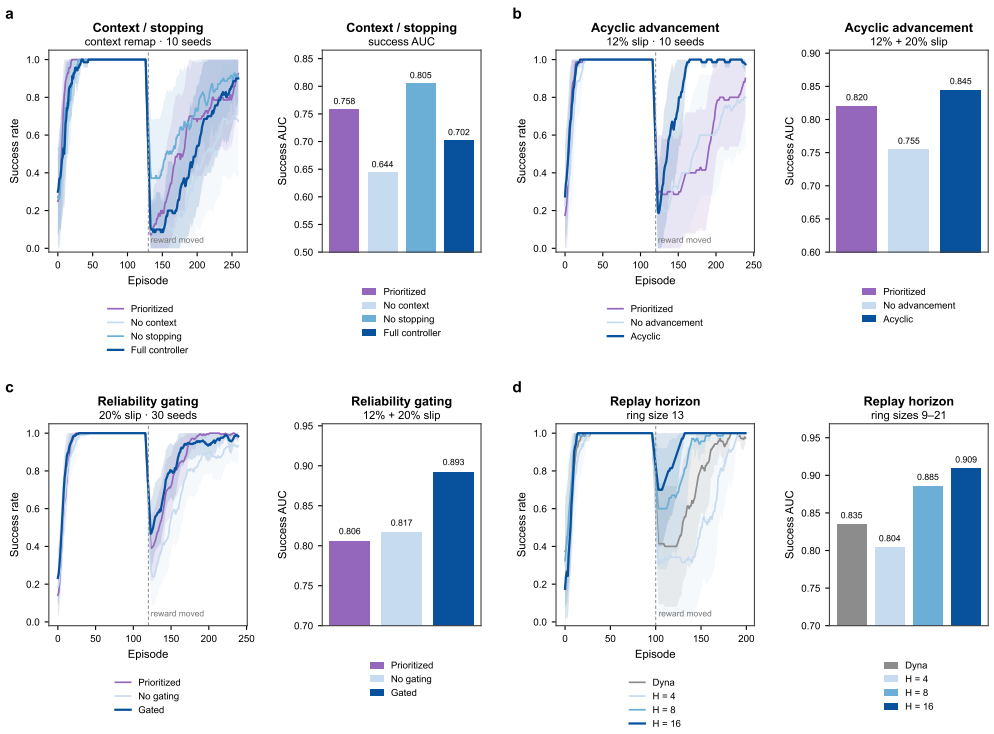


Figure 4: **Evidence of components by target failure mode.** Four blocks: (a) context and stopping, (b) acyclic advancement, (c) reliability gating, and (d) replay horizon. Each block pairs the learning curve (left) with the success AUC of every variant (right).

was 16 at sizes 9, 13, and 21 and 8 at size 17. At size 21, Horizon 16 outperformed Dyna by $+0.178$ AUC and maintained a geometric mean update ratio of 0.764.

Longer runs also revealed limitations on compact graphs. At Horizon 16, 13.2% of low-confidence edges were selected at size 9 (Figure 5a). At sizes 13, 17, and 21, the selection rate was less than 10%. In the final matrix, Horizon 8 was selected when the learned diameter was approximately 12, and Horizon 16 was selected when it was approximately 29 (Figure 5b). By assigning $H{=}8$ to compact graphs, the diameter rule keeps the low-confidence selection rate below 10% in every condition, including the high-noise compact ring (8.8%).

5.4 Transfer to unseen graph sizes

The diameter rule fixed at sizes 9 and 17 was applied without modification to ring, bridge, and cross graphs of size 25 at both noise levels. LoopReplay outperformed the strongest conventional baseline in all six conditions, by $+0.055$ to $+0.135$ AUC and $+0.091$ on average (hierarchical 95% interval $[0.054, 0.124]$; Figure 5c). Against the best of all five baselines in each condition the gain was $+0.055$ ($[0.016, 0.098]$), while LoopReplay used fewer replay updates per online step (ratio 0.88).

6 Mechanistic Interpretation

The experiments allow us to assign independently measurable roles to each of the three control variables. Advancement converts replay capacity into forward propagation on the graph. Without it, the completion rate of backward sweeps drops below 6%. Reliability guides sequence selection along well-founded transition structures. Without it, the selection rate of low-reliability edges exceeds 30% under high-noise conditions. Diameter aligns the propagation depth with the structural distance. Without it, a fixed horizon leads to insufficient propagation in large graphs and excessive selection of

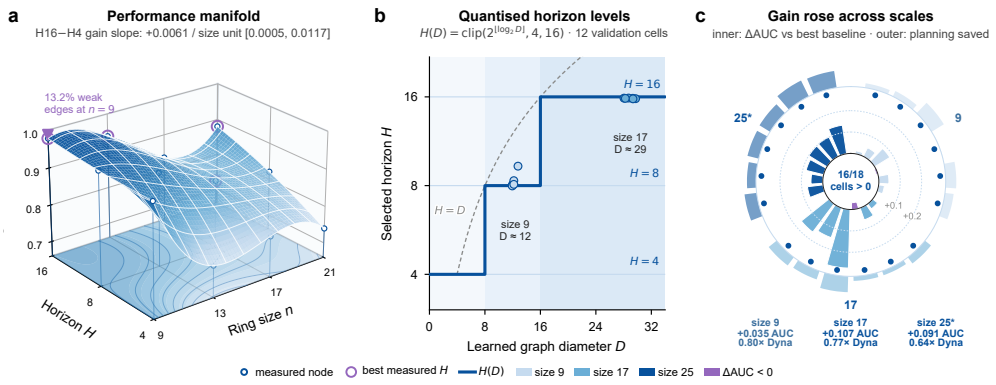


Figure 5: **Calibration of replay depth by structural scale.** **(a)** Success AUC over ring size and fixed horizon; white nodes are the 12 measured conditions, and the surface is a spline interpolation for visual guidance only. Purple circles: best measured H per size; triangle: 13.2% low-confidence selection at n=9, H=16. **(b)** $H(D)$ steps with the mean selected horizon and learned diameter of each validation cell (light: size 9; dark: size 17). **(c)** Per-cell ΔAUC over the stronger baseline (inner; purple < 0) and planning saved, $1 -$ updates/Dyna (outer, floored at 0), for sizes 9, 17, and unseen 25 (*); per-size means below.

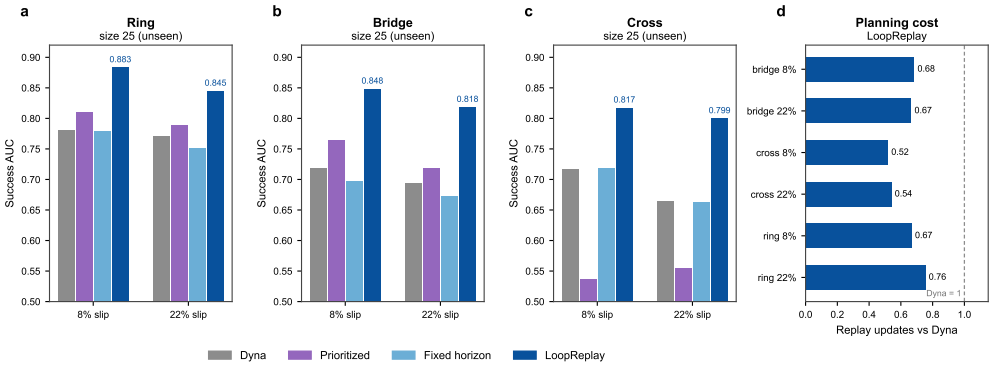


Figure 6: **Transfer to unseen graph scales.** Condition-specific effects, replay-update ratios, and performance and efficiency by topology across six size-25 conditions.

weak edges in compact graphs. Context safety is an invariant rather than a tuned operator: it isolates reward identity while reusing connectivity across reward contexts.

Each variable predicts distinct failure characteristics (incomplete sweeping, increased weak edges, and scale-dependent AUC interactions), and the experiments measure each of these independently of aggregate performance. This separation allows us to falsify the mechanism at the operator level.

7 Biological Correspondences and Testable Predictions

The computational results correspond to hippocampal replay at the algorithmic level. A transition graph analogous to CA3 supports ordered sequence completion, while a mismatch signal analogous to CA1 controls access and directs orientation. Reward outcomes modulate context, and the learned structural scale governs the effective depth of replay. Four testable predictions are derived from this correspondence. (1) After a reward location shifts, access to backward replay increases prior to behavioral recovery. (2) Even after controlling for proximity to the reward and the magnitude of mismatch, low-confidence transitions are less likely to be included in replay. (3) The larger the diameter of the learned path, the longer the effective replay sequence becomes. (4) In stochastic and compact environments, excessively long sequences increase the proportion of weak transitions

without improving selection accuracy. Appendix K distinguishes between these predictions, the implemented operators, and the measured algorithmic evidence.

8 Scope and Extensions

The controlled tabular setting isolates each operator with exact graph diameters and measurable telemetry. The operators themselves require only a transition graph, visit counts, and a distance estimate. With learned latent graphs and function approximation, approximate diameters and learned context estimates take the place of the exact quantities used here.

9 Conclusion

LoopReplay controls replay as an observable graph process. Event gates and signed directions determine access. Acyclic scanning ensures progress. Confidence determines the order of transitions. Context isolates past rewards. Residual error terminates exhausted sweeps. Learned diameter determines propagation depth. Across 12 held-out conditions spanning topology, scale, noise, and reward movement geometry, the method outperformed replay and planning baselines at matched compute, with the largest gains when credit must be reassigned after the reward moves. The same rules generalize to unseen graph sizes and improve all six conditions there.

Reproducibility statement

The supplementary materials include the complete algorithm, constants, environment setup rules, baseline implementations, statistical procedures, and all results by condition. Deliverables include configuration files, source code, seed-level episode logs for all experiments, and scripts for deterministically generating figures.

AI use statement

Generative AI tools were used for language editing, code implementation and debugging, literature search assistance, and layout consideration. The research objectives, algorithms, experimental protocols, evaluation criteria, statistical analyses, and scientific conclusions were determined by the authors. The authors verified the references and validated the code-assisted outputs and numerical claims against the stored artifacts, and they assume responsibility for the entire manuscript and related materials.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, 2021.

R. Ellen Ambrose, Brad E. Pfeiffer, and David J. Foster. Reverse replay of hippocampal place cells is uniquely modulated by changing reward. *Neuron*, 91(5):1124–1136, 2016.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, 2017.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, 2019.

Jacopo Bono and Claudia Clopath. Learning predictive cognitive maps with spiking neurons during behaviour and replays. *eLife*, 12:e80671, 2023.

György Buzsáki. Hippocampal sharp wave-ripple: A cognitive biomarker for episodic memory and planning. *Hippocampus*, 25(10):1073–1188, 2015.

Thomas J. Davidson, Fabian Kloosterman, and Matthew A. Wilson. Hippocampal replay of extended experience. *Neuron*, 63(4):497–507, 2009.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

Kamran Diba and György Buzsáki. Forward and reverse hippocampal place-cell sequences during ripples. *Nature Neuroscience*, 10(10):1241–1242, 2007.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993.

William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting fundamentals of experience replay. In *International Conference on Machine Learning*, 2020.

Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I. Jordan, Joseph E. Gonzalez, and Sergey Levine. Model-based value expansion for efficient model-free reinforcement learning. In *International Conference on Machine Learning*, 2018.

David J. Foster and Matthew A. Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440:680–683, 2006.

Anirudh Goyal, Philemon Brakel, William Fedus, Soumye Singhal, Timothy Lillicrap, Sergey Levine, Hugo Larochelle, and Yoshua Bengio. Recall traces: Backtracking models for efficient reinforcement learning. In *International Conference on Learning Representations*, 2019.

Anoopum S. Gupta, Matthijs A. A. van der Meer, David S. Touretzky, and A. David Redish. Hippocampal replay is not a simple function of experience. *Neuron*, 65(5):695–705, 2010.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado van Hasselt, Greg Wayne, Satinder Singh, Doina Precup, and Rémi Munos. Hindsight credit assignment. In *Advances in Neural Information Processing Systems*, 2019.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, 2019.

Kristopher T. Jensen, Guillaume Hennequin, and Marcelo G. Mattar. A recurrent network model of planning explains hippocampal replay and human behavior. *Nature Neuroscience*, 27:1340–1348, 2024.

Nan Jiang, Alex Kulesza, Satinder Singh, and Richard Lewis. The dependence of effective planning horizon on model accuracy. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 1181–1189, 2015.

Steven Kapturowski, Georg Ostrovski, John Quan, Rémi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *International Conference on Learning Representations*, 2019.

Su Young Lee, Sungik Choi, and Sae-Young Chung. Sample-efficient deep reinforcement learning via episodic backward update. In *Advances in Neural Information Processing Systems*, 2019.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3–4):293–321, 1992.

John E. Lisman and Anthony A. Grace. The hippocampal-VTA loop: controlling the entry of information into long-term memory. *Neuron*, 46(5):703–713, 2005.

Marcelo G. Mattar and Nathaniel D. Daw. Prioritized memory access explains planning and hippocampal replay. *Nature Neuroscience*, 21(11):1609–1617, 2018. doi: 10.1038/s41593-018-0232-z.

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.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 16(1):1–118, 2023.

Ida Momennejad, A. Ross Otto, Nathaniel D. Daw, and Kenneth A. Norman. Offline replay supports planning in human reinforcement learning. *eLife*, 7:e32548, 2018.

Andrew W. Moore and Christopher G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning*, 13:103–130, 1993.

John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171–175, 1971.

H. Freyja Ólafsdóttir, Daniel Bush, and Caswell Barry. The role of hippocampal replay in memory and planning. *Current Biology*, 28(1):R37–R50, 2018.

Jing Peng and Ronald J. Williams. Efficient learning and planning within the Dyna framework. *Adaptive Behavior*, 1(4):437–454, 1993.

Brad E. Pfeiffer and David J. Foster. Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*, 497:74–79, 2013.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

Annabelle C. Singer and Loren M. Frank. Rewarded outcomes enhance reactivation of experience in the hippocampus. *Neuron*, 64(6):910–921, 2009.

Kimberly L. Stachenfeld, Matthew M. Botvinick, and Samuel J. Gershman. The hippocampus as a predictive map. *Nature Neuroscience*, 20(11):1643–1653, 2017.

Alexander L. Strehl and Michael L. Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3 (1):9–44, 1988.

Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning*, pp. 216–224, 1990.

Hado van Hasselt, Matteo Hessel, and John Aslanides. When to use parametric models in reinforcement learning? In *Advances in Neural Information Processing Systems*, 2019.

Harm van Seijen and Richard S. Sutton. Planning by prioritized sweeping with small backups. In *International Conference on Machine Learning*, 2013.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3–4):279–292, 1992.

Matthew A. Wilson and Bruce L. McNaughton. Reactivation of hippocampal ensemble memories during sleep. *Science*, 265(5172):676–679, 1994.

SUPPLEMENTARY MATERIAL

A EXTENDED RELATED WORK

Experience replay and temporal-difference credit assignment. Temporal-difference learning propagates reward information one transition at a time (Sutton, 1988), and tabular Q-learning provides the off-policy backup shared by all conditions in this study (Watkins & Dayan, 1992). Experience replay was introduced to reuse stored transitions (Lin, 1992) and became a core component of deep value-based agents (Mnih et al., 2015). Later analyses showed that replay capacity and the ratio of replay to online updates strongly influence learning (Fedus et al., 2020), and that a replay buffer can act as a nonparametric model whose benefits overlap with those of parametric planning (van Hasselt et al., 2019). LOOPREPLAY reports its replay-update ratio as an explicit cost variable for the same reason: gains in AUC are interpreted jointly with the number of replay backups used to obtain them.

Prioritization and sequence order. Prioritized sweeping propagates value changes to predecessors of states whose values change (Moore & Atkeson, 1993; Peng & Williams, 1993), and small backups reduce the computational cost of each propagation step (van Seijen & Sutton, 2013). Prioritized experience replay applies a related principle to individually stored transitions ranked by TD error (Schaul et al., 2016). Sequence-level methods instead preserve temporal order: backward replay over whole episodes (Lin, 1992; Lee et al., 2019), learned backtracking models that generate predecessor traces toward high-value states (Goyal et al., 2019), and stored recurrent sequences in distributed agents (Kapturowski et al., 2019). These approaches determine *which* transitions or sequences are replayed. LOOPREPLAY adds controls that these methods do not specify jointly: acyclic advancement to guarantee progress along the graph, confidence ordering over competing stochastic successors, a residual stopping rule, and context isolation of reward labels after relocation.

Sparse and delayed credit assignment. Hindsight experience replay relabels goals so that failed trajectories carry learning signal (Andrychowicz et al., 2017). Return decomposition redistributes delayed returns to the steps that caused them (Arjona-Medina et al., 2019), and hindsight credit assignment estimates the contribution of past actions to observed outcomes (Harutyunyan et al., 2019). The successor representation factorizes values into expected future occupancy and reward, allowing rapid revaluation after reward changes (Dayan, 1993). These methods modify the learning target or the value representation. LOOPREPLAY keeps the one-step tabular target fixed and changes only the order, direction, and depth of replayed backups, so the two families are complementary rather than competing.

Model-based planning depth and model uncertainty. Dyna-style agents interleave real experience with simulated backups from a learned model (Sutton, 1990; Moerland et al., 2023). The effective planning horizon interacts with model accuracy: shorter horizons can outperform the nominal horizon when the model is imperfect (Jiang et al., 2015), and short model rollouts branched from real data limit compounding error (Janner et al., 2019). Value expansion (Feinberg et al., 2018) and latent-model control methods (Hansen et al., 2024) likewise treat the rollout length as a design variable. In the tabular setting, count-based confidence intervals over empirical transitions provide a principled treatment of model uncertainty (Strehl & Littman, 2008). LOOPREPLAY uses empirical confidence to rank candidate edges rather than to construct optimistic values, and it sets depth from the learned graph diameter rather than from a fixed rollout length.

Hippocampal replay and planning. Place cells provide a spatial code on which replay operates (O'Keefe & Dostrovsky, 1971). Sequential reactivation occurs during sleep and awake sharp-wave ripples (Wilson & McNaughton, 1994; Foster & Wilson, 2006; Diba & Buzsáki, 2007; Buzsáki, 2015). Replay can cover extended trajectories (Davidson et al., 2009), construct paths not directly experienced (Gupta et al., 2010), and is modulated by reward and by changes in reward (Singer & Frank, 2009; Ambrose et al., 2016); reviews link these properties to both memory and planning (Ólafsdóttir et al., 2018). Novelty-dependent gating of hippocampal encoding has been proposed through a hippocampal–VTA loop (Lisman & Grace, 2005). Normative and network models relate replay to prioritized memory access (Mattar & Daw, 2018), predictive maps (Stachenfeld et al., 2017; Bono & Clopath, 2023), and recurrent planning (Jensen et al., 2024), and offline replay in humans supports planning after task changes (Momennejad et al., 2018). As stated in the main text and

Appendix K, LOOPREPLAY uses these findings only as a computational motivation; no new neural data are analyzed.

B ALGORITHM SPECIFICATION

This section expands the steps of Algorithm 1 in Section 3. At each online step, the agent stores a transition and updates the online value function and the context-indexed predecessor and successor lists. A reward omission at a previously rewarded state advances the active context. For a candidate transition i, the priority is

$$p_i = |\delta_i|\, c_i + 0.5\, \ell_i + 0.75 \max(r_i, 0), \tag{5}$$

where c_i is empirical transition confidence and ℓ_i marks loop closure. When a rewarded transition is present, the sweep is seeded there; otherwise replay starts when the largest priority crosses the event threshold.

The signed event selects reverse or forward traversal. Reverse traversal follows context-matched predecessors after reward, sufficiently positive mismatch, or reward-omission devaluation. Forward traversal follows context-matched successors for other eligible events. Acyclic advancement removes stationary transitions and repeated states. Cross-context fallback can reuse connectivity, but it removes reward and terminal labels before the edge enters the active sequence. Candidate edges are ranked by confidence-weighted mismatch.

The horizon is computed from the diameter of the learned transition graph:

$$H(D) = \mathrm{clip}\Big(2^{\lfloor \log_2(\max(D,1)) \rfloor},\ 4,\ 16\Big). \tag{6}$$

The replay interval equals $H(D)$. Backward sweeps run to the horizon; forward sweeps stop when the mean absolute residual falls below 0.025 after two edges. Every selected transition receives the shared tabular TD update. The implementation records direction, horizon, diameter, confidence, completion, stopping, fallback, and cross-context reward-edge telemetry.

C PROTOCOL AND CONSTANTS

The final external confirmation uses three topologies (ring, bridge, and cross), two graph sizes (9 and 17), two slip probabilities (0.08 and 0.22), two balanced relocation directions, four primary conditions (prioritized replay, empirical-model Dyna, the fixed-horizon ablation, and method), ten held-out seeds per cell, 200 episodes per seed, and relocation at episode 100. Hierarchical intervals resample cells and paired seed indices. No algorithmic parameter was changed after the matrix was run.

Table S1: **Protocol constants.** The baseline replay interval is eight; LOOPREPLAY replaces the interval and budget with the learned $H(D)$.

Constant	Value
Learning rate α	0.180
Discount γ	0.985
Exploration schedule	$0.30 \to 0.02$, linear
Episodes per seed	200; relocation at episode 100
Final-matrix seeds	130–139, held out
Replay budget / interval	8 / 8 for baselines; method sets both to $H(D)$
Conditions	prioritized, Dyna, fixed horizon, method
Event threshold	$\max_i p_i \geq 0.15$
Residual stop	$\overline{\lvert\delta\rvert} < 0.025$ for forward sweeps
Horizon candidates	$\{4, 8, 16\}$
Maximum steps	140, 260, 400 at sizes 9, 17, 25

D BASELINE AND IMPLEMENTATION DETAILS

Shared learner. All conditions use the same tabular Q-learning learner (Watkins & Dayan, 1992) for online and replay updates, with the learning rate, discount, and exploration schedule in Table S1.

Environment construction, held-out seeds, episode count, relocation episode, and step limits are identical across conditions. Replay opportunities occur every replay interval and at loop closure or episode termination. Conditions therefore differ only in which transitions are backed up, in what order, and how many.

Prioritized replay. Transitions among the 512 most recent are sampled with replacement in proportion to $|\delta_i| + 10^{-4}$ (Schaul et al., 2016); eight backups are made every eight online steps, and each replayed transition's priority is refreshed with its new TD error.

Dyna. Dyna samples a stored state–action pair uniformly and backs up one of its stored outcomes (Sutton, 1990), with the same budget and interval.

Dyna-Q. The standard tabular Dyna-Q model stores the most recent outcome $(r, s', \text{terminal})$ of each state–action pair; planning backs up uniformly sampled observed pairs from this model.

Prioritized sweeping. Following Moore & Atkeson (1993), each online TD error above 10^{-4} enters a priority queue; each planning step pops the largest entry, backs it up from the most-recent-outcome model, and pushes the predecessors of its state with their model TD errors.

Backward trajectory replay. Following the backward ordering of Lin (1992) and Lee et al. (2019), each replay opportunity backs up the most recent transitions of the stored trajectory in reverse temporal order.

Fixed-horizon ablation. This ablation contains every LOOPREPLAY operator with a fixed replay budget and interval of eight and a backward-sweep length of four. It isolates the contribution of diameter calibration.

LOOPREPLAY. The replay horizon, budget, and interval are set to $H(D)$ (Equation 4), where D is the exact undirected diameter of the learned transition graph. A transition counts as low-confidence when $c_i < 0.5$. Dyna-Q, prioritized sweeping, and backward replay use the budget and interval of the conventional baselines.

Table S2: **All baselines at equal replay budgets.** Mean success AUC over cells and seeds, split into acquisition (before relocation) and post-relocation phases, and replay updates per online step. Differences are paired by seed; intervals are hierarchical 95% bootstrap intervals over cells and seeds.

	Sizes 9 and 17 (12 cells)				Size 25 (6 cells)	
Method	AUC	Acquisition	Post-relocation	Updates/step	AUC	Updates/step
Prioritized replay	0.817	0.945	0.689	1.14	0.695	1.04
Dyna	0.837	0.951	0.722	1.13	0.724	1.04
Dyna-Q	0.865	0.950	0.780	1.14	0.712	1.04
Prioritized sweeping	0.867	0.958	0.777	1.15	0.773	1.06
Backward trajectory replay	0.778	0.947	0.609	1.12	0.701	1.04
Fixed-horizon ablation	0.890	0.950	0.830	0.72	0.713	0.55
LOOPREPLAY	**0.914**	0.950	**0.877**	1.10	**0.835**	0.92
LOOPREPLAY minus Dyna-Q: $+0.048$ $[0.013, 0.087]$ (sizes 9/17); $+0.124$ $[0.102, 0.150]$ (size 25)						
LOOPREPLAY minus prioritized sweeping: $+0.046$ $[0.007, 0.090]$; $+0.062$ $[0.018, 0.117]$						
LOOPREPLAY minus backward replay: $+0.136$ $[0.074, 0.206]$; $+0.134$ $[0.096, 0.171]$						

Statistical procedure. Within each cell, the seed is the unit of inference. Intervals are hierarchical percentile bootstraps (Efron & Tibshirani, 1993) with 10,000 draws that resample cells and, within each resampled cell, paired seed indices, so that all methods are compared on the same seeds. For each cell, the reference is the stronger baseline in that cell. Reporting interval estimates over tasks and seeds follows recommendations for reliable reinforcement-learning evaluation (Agarwal et al., 2021). The complete validation matrix runs in about seven CPU-minutes.

Table S3: **Sensitivity to constants.** Mean success AUC of LOOPREPLAY over the 12 validation cells and 10 seeds when one constant is changed at a time. The default is 0.914; the best baseline reaches 0.867.

Constant (default)	Lower value		Higher value	
Loop-closure weight (0.5)	0.0	0.906	1.0	0.912
Reward weight (0.75)	0.25	0.914	1.5	0.914
Event threshold (0.15)	0.05	0.917	0.30	0.900
Residual tolerance (0.025)	0.01	0.900	0.05	0.908

E COMPLETE FINAL CONFIRMATION

Figure S1 expands the main-text result into the complete learning-curve matrix. Every cell uses the same axes, method order, rolling window, relocation marker, and uncertainty convention. This repeated small-multiple grammar makes cell-specific acquisition and relocation dynamics directly comparable while Table S4 preserves the exact aggregate values.

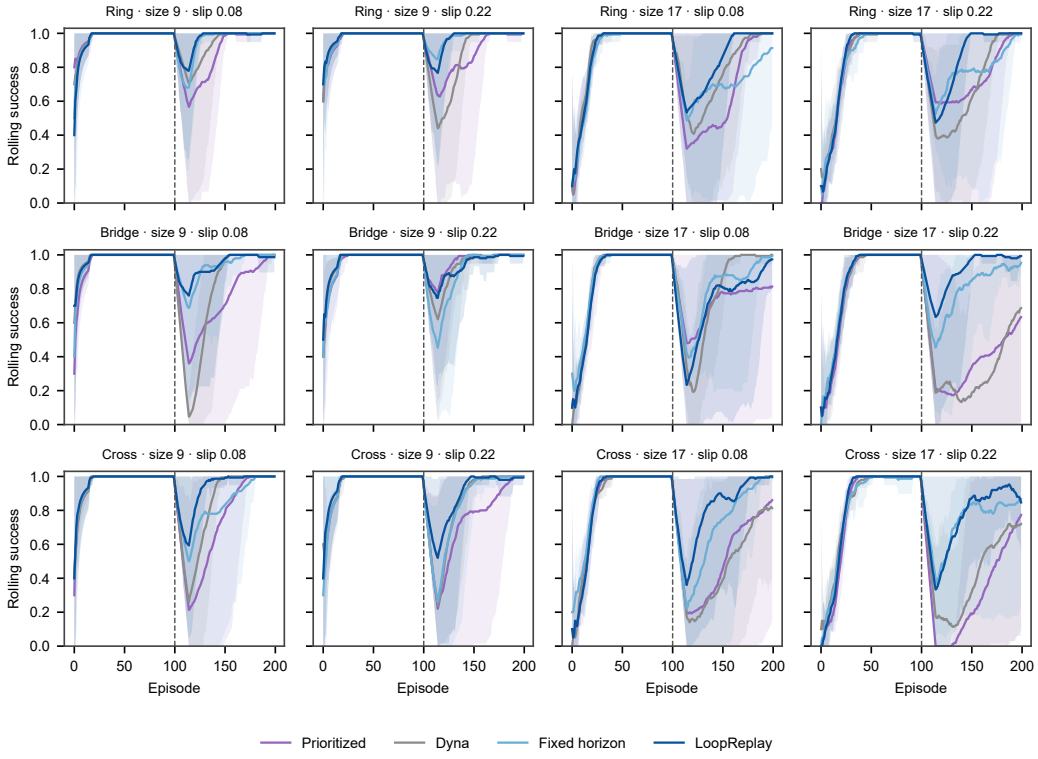


Figure S1: **Full validation matrix.** Each panel is one topology–size–slip–relocation cell. Lines show the mean 15-episode rolling success across ten held-out seeds; shading spans the 2.5th–97.5th seed percentiles. The dashed vertical line marks reward relocation at episode 100. All panels share axes, method colors, and legend order.

Table S4: **Complete cell results.** First block: size-9/17 validation cells; second block: unseen size-25 cells (mechanism telemetry logged for sizes 9 and 17). Fixed: fixed-horizon ablation; Ours: LOOPREPLAY; Ratio: replay updates per episode relative to the stronger baseline.

Cell	P-RP	Dyna	Fixed	Ours	Δbest	Ratio	Low-conf.	Completion	Level
ring_s9_n08_ne	0.936	0.964	0.971	0.978	+0.014	0.881	4.5%	99.8%	seed-level
ring_s9_n22_sw	0.933	0.926	0.983	0.979	+0.045	0.839	8.8%	99.8%	seed-level
ring_s17_n08_sw	0.798	0.861	0.820	0.889	+0.029	0.792	3.1%	99.9%	seed-level
ring_s17_n22_ne	0.827	0.808	0.846	0.886	+0.059	0.880	7.1%	99.8%	seed-level
bridge_s9_n08_sw	0.864	0.881	0.960	0.965	+0.083	0.611	3.5%	98.2%	seed-level
bridge_s9_n22_ne	0.970	0.958	0.935	0.962	-0.008	1.001	7.5%	99.2%	seed-level
bridge_s17_n08_ne	0.815	0.850	0.863	0.819	-0.032	0.876	3.0%	99.8%	seed-level
bridge_s17_n22_sw	0.629	0.614	0.850	0.903	+0.274	0.569	7.0%	98.9%	seed-level
cross_s9_n08_ne	0.861	0.912	0.914	0.959	+0.047	0.712	4.3%	95.9%	seed-level
cross_s9_n22_sw	0.862	0.910	0.908	0.938	+0.027	0.764	8.7%	96.6%	seed-level
cross_s17_n08_sw	0.720	0.701	0.823	0.867	+0.147	0.760	3.7%	99.2%	seed-level
cross_s17_n22_ne	0.594	0.655	0.805	0.821	+0.165	0.724	7.8%	99.6%	seed-level
ring_s25_n08_ne	0.810	0.781	0.779	0.883	+0.073	0.669	–	–	seed-level
ring_s25_n22_ne	0.789	0.770	0.751	0.845	+0.055	0.756	–	–	seed-level
bridge_s25_n08_ne	0.764	0.718	0.696	0.848	+0.084	0.685	–	–	seed-level
bridge_s25_n22_ne	0.719	0.693	0.671	0.818	+0.099	0.666	–	–	seed-level
cross_s25_n08_ne	0.535	0.716	0.719	0.817	+0.101	0.520	–	–	seed-level
cross_s25_n22_ne	0.554	0.664	0.662	0.799	+0.135	0.542	–	–	seed-level

F CONTEXT, DIRECTION, AND STOPPING DIAGNOSTICS

The context-remapping protocol isolates access and stopping operators. Figure S2 contains six ablations that replace one controller component at a time while preserving shared axes, rolling window, relocation marker, and legend.

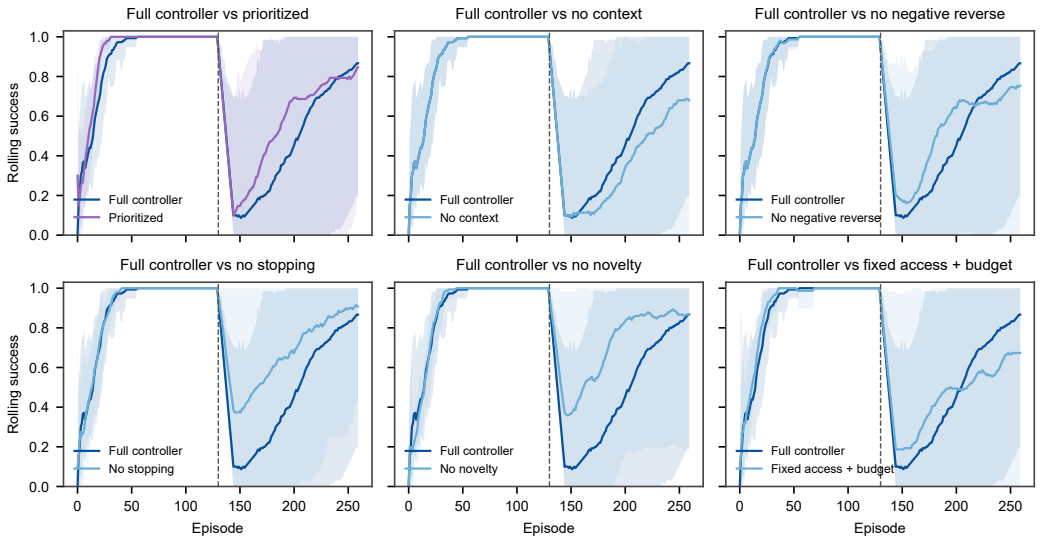


Figure S2: **Context, direction, novelty, and stopping ablations.** The 3×2 matrix compares the full controller with prioritized replay, context removal, negative-reverse removal, stopping removal, novelty removal, and the combined fixed-access/fixed-budget condition. The last condition forces replay access and fills an accepted sweep to the fixed budget; it is not interpreted as a gate-only intervention. Lines are mean 15-episode rolling success across ten seeds; shading spans the 2.5th–97.5th seed percentiles. The dashed line marks reward relocation at episode 130 in this 260-episode protocol.

G Acyclic Advancement Validation

The advancement intervention removes stationary transitions and repeated states from the candidate walk. Figure S3 repeats the same three process curves in the two noise regimes: behavioral recovery, reverse-sweep completion, and advancing-edge fraction. The acyclic controller also retains the reward-safe structural fallback, which is almost never activated, so the completion change is assigned to acyclic advancement.

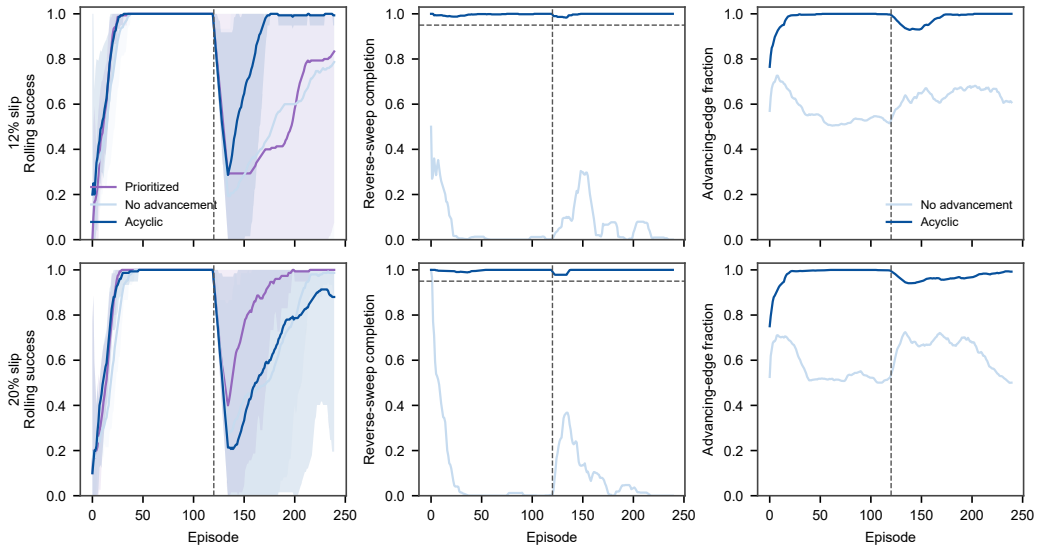


Figure S3: **Acyclic advancement process curves.** Rows are the 12% and 20% slip protocols; columns show 15-episode rolling success, reverse-sweep completion, and advancing-edge fraction. Lines and seed-percentile bands compare prioritized replay where applicable, the controller without advancement, and the acyclic controller. The horizontal dashed line marks the 95% completion gate; the vertical dotted line marks relocation at episode 120 in these 240-episode protocols.

H Reliability-Gating Confirmation

The reliability intervention multiplies mismatch priority by empirical edge confidence and retains the same graph-walk safeguards. Figure S4 shows, for each slip regime, the behavioral curve above the reliability-telemetry curve. This pairs the outcome with the operator-specific signal in the same visual unit.

I External Breadth Before Diameter Calibration

The fixed-horizon external matrix precedes diameter calibration and isolates the remaining scale-related boundary. Figure S5 places performance and planning compute on one shared curve. The horizontal position records cumulative mean replay updates; the vertical position records the rolling success aggregated across the 12 cells.

J Diameter Experiment and Unseen Graph Size

The fixed-horizon experiment isolates the structural-scale interaction before the diameter rule is fixed. Figure S6 uses three aligned scaling panels: fixed-horizon calibration, learned diameter-to-horizon selection, and transfer to unseen size-25 topologies. The common line grammar carries the structural variable from calibration to deployment.

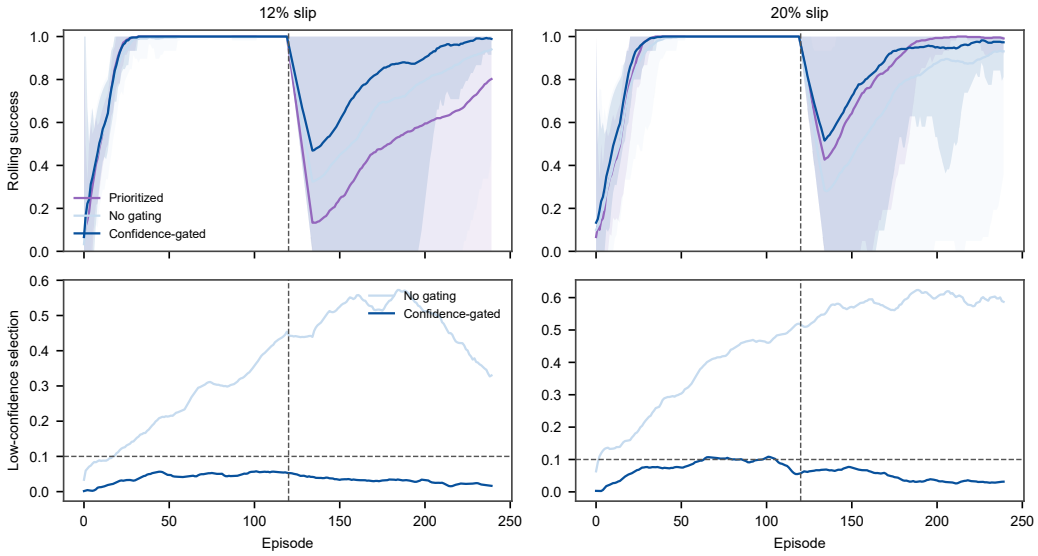


Figure S4: **Thirty-seed reliability component blocks.** Columns are the 12% and 20% slip regimes. Top: 15-episode rolling success with seed-percentile bands. Bottom: rolling low-confidence selection rate. Dashed lines mark relocation at episode 120 or the 10% low-confidence gate.

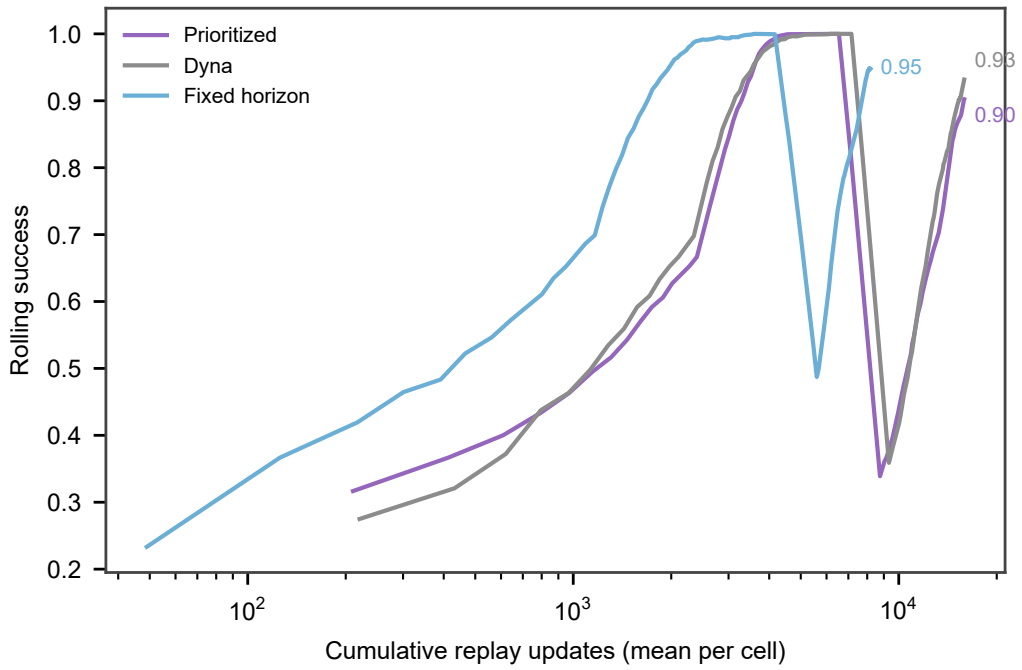


Figure S5: **External-matrix performance–planning trajectories.** Curves show the 15-episode rolling success aggregated across all 12 fixed-horizon external-matrix cells against cumulative mean planning updates on a log scale. Online experience advances concurrently, so this is a descriptive performance–planning position rather than a controlled compute-scaling experiment. Endpoints are labelled with final rolling success.

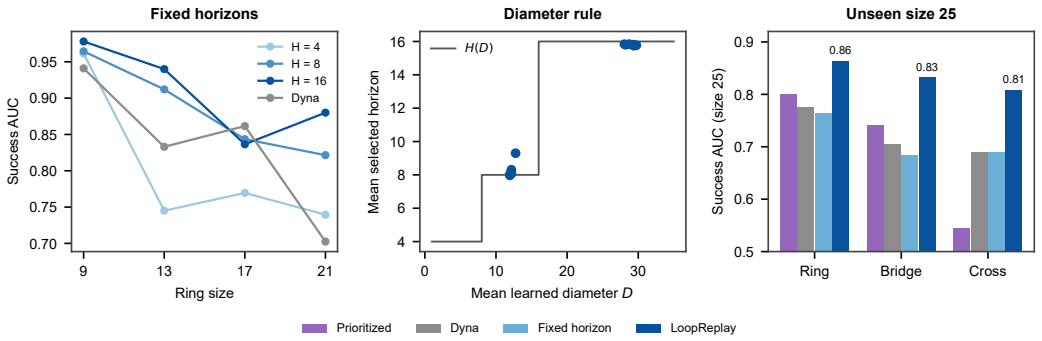


Figure S6: **Structural scaling and unseen-size transfer.** Left: fixed-horizon AUC across ring sizes. Middle: mean learned structural diameter versus mean selected replay horizon in each validation cell, with the $H(D)$ staircase; averaging over replay events places points near, rather than necessarily exactly on, the integer step. Right: mean aggregate success AUC over the two available slip cells for each unseen size-25 topology and method. Error bars: 95% intervals over seeds.

K Algorithmic Hippocampal Mapping

The biological mapping is an interpretation layer over the implemented operators. Ordered graph traversal corresponds to sequence replay; mismatch contributes to event access and signed direction; reward omission advances context; empirical confidence controls transition reliability; and learned graph diameter controls propagation depth. These correspondences generate measurable algorithmic signatures (direction counts, advancing completion, selected-edge confidence, reward leakage, and horizon selection) without introducing a new animal experiment or a new empirical claim.

L Reproducibility and Data Boundary

The appendix figures and tables are generated from the stored CSV/JSON records and analysis scripts distributed with the project artifact. No experiment is rerun during figure generation. The size-9/17 matrix contains ten held-out seeds per cell; the reliability confirmations contain thirty seeds per slip protocol; and the size-25 transfer contains ten seeds per cell.