When Does Navigation Pretraining Help? The Role of Action Selection

Anonymous authors
Paper under double-blind review

Abstract

Does the way a reinforcement-learning agent chooses actions change the measured benefit of pretraining? We study recurrent PPO agents adapting from local maze grids to directional wall distances and goal-visibility signals. Transfer retains the trained controller behind a fresh input encoder. Two settings compare maze pretraining, starting from scratch and equally long obstacle-field pretraining, with eight training seeds per condition. We evaluate the same saved networks by always taking the most probable action (greedy selection) or drawing from the learned probabilities (sampling). In the original setting, sampling improves scratch more than maze pretraining. The pretrained-minus-scratch difference in average success during adaptation on a zero-to-one scale changes from $+0.053$ to -0.054: a shift of -0.107 (descriptive 95% bootstrap interval $[-0.182, -0.044]$). However, neither ordering establishes superiority, and the task inventories overlap. An exploratory follow-up on larger mazes with structurally separate training and test sets does not reproduce this reversal: pretraining stays ahead on average, and the change in its advantage is uncertain. A further control preserves the probability of leaving the greedy action but chooses the alternatives equally. Sampling improves performance beyond this control, without clearly changing the pretraining advantage. Action selection can change a transfer comparison, but that sensitivity does not necessarily carry over to another task setting.

1 Introduction

Pretraining is useful if it improves subsequent learning. In reinforcement learning, that comparison also depends on how the agent acts. Greedy selection always takes the most probable action; sampling retains the alternatives. Even when pretrained and scratch-trained agents use the same rule, changing that rule may help them unequally. We test whether it changes the measured pretraining benefit across the complete adaptation process.

Our setting is navigation with a changed sensor. Agents learn from a small local grid, then adapt to numerical rays indicating wall distances and visible goals. Both views are partial, not complete maps. Transfer copies the recurrent controller and output heads behind a fresh encoder, the learned transformation of the inputs. We compare maze pretraining with scratch and equal-budget pretraining among scattered obstacles. Prior work compares action rules after training (Jang et al., 2025); we test whether changing the rule affects these training histories equally.

We report two findings:

1. **An action-selection interaction with limited scope.** Sampling helps scratch more in all eight original seed blocks, reversing the mean ordering without establishing superiority. This reversal does not recur in a structural holdout with 32 new agents, where pretraining has higher mean scores under all four tested rules.
2. **A distinction between better navigation and greater transfer benefit.** Learned relative probabilities of non-greedy actions improve performance beyond matching their total probability, but their effect on the pretraining advantage remains uncertain.

Supporting encoder and simple-policy controls limit what either result reveals about spatial knowledge: successful behaviour alone does not identify a representation (Firestone, 2020).

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.

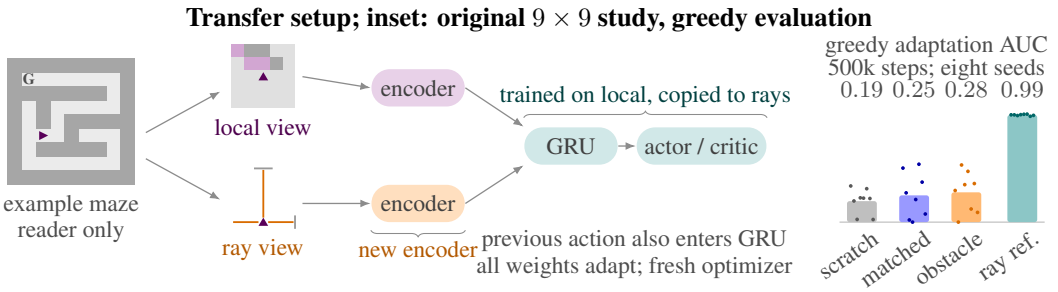


Figure 1: Transferring a recurrent navigation policy between sensors. The same maze state supplies a local patch or numerical rays, not camera images; the full map is shown only for the reader. Local pretraining supplies the GRU and actor/critic behind a fresh ray encoder. All weights then adapt. Right: mean normalized learning-curve area (AUC) under greedy evaluation over 500k target interactions; dots show all eight training seeds per arm. The matched-local advantage over scratch is not established, not ruled out. The ray reference retains prior ray experience and its trained encoder. These original evaluation entries overlap training content. Exact inputs and the illustrative pose appear in Appendix F.

2 Related work

Transfer across observations. Cross-View Policy Learning uses aerial and ground-level views for street navigation (Li et al., 2019). DARLA separates representation learning from policy learning (Higgins et al., 2017); Global Workspace policies use representations trained across modalities (Maytié et al., 2024). BIFROST studies shared history representations for observation and dynamics transfer (Deng & Hanna, 2026). Our unaligned encoder-replacement procedure does not test these methods. Its failure cannot establish that cross-view transfer is generally unavailable.

Interpreting reinforcement-learning comparisons. Implementation details can change apparent algorithmic gains (Engstrom et al., 2020); few independent training runs and evaluation design complicate performance comparisons (Henderson et al., 2018; Agarwal et al., 2021; Patterson et al., 2024). Jang et al. (2025) directly compare post-training action-selection rules for PPO agents across eight Atari environments. Thus changing a saved policy's action rule is not itself our contribution. We instead measure whether the same change alters scratch and pretrained agents equally across their complete adaptation curves. We do not propose stochastic evaluation as universally preferable to greedy evaluation.

Navigation and spatial competence. Memory Maze targets long-term memory in 3D navigation (Pasukonis et al., 2023), whereas POPGym notes that perfect mazes admit wall-following solutions (Morad et al., 2023). Restricted vision alone does not make maze success a selective test of spatial memory. Wijmans et al. (2023) combine behavioural and representational interventions to investigate maps in blind agents, under different sensory and position-related inputs. Our behavioural controls do not support the same representational conclusion. Parameter-reset work (Nikishin et al., 2022; Zilly et al., 2021) motivates adaptation controls, but differs from our one-time encoder replacement and on-policy repair.

3 Controlled navigation study

We use the same four training histories in two settings (Table 1). The first retains the original frozen comparison; the second tests its scope on larger mazes with separate train and test structures. Neither study changes the action rule during training: all action-selection comparisons re-evaluate saved checkpoints without learning. For the structural split, walkable cells are graph nodes and neighbouring cells are connected. Comparing these trees without grid coordinates keeps the same corridor structure from appearing in both training and testing.

Table 1: Two settings, not an isolated intervention on train–test overlap. Both use eight training seeds per condition, one million source interactions for pretrained arms and 500,000 target interactions for every arm.

	Original setting	Structural holdout
Maze size; action cap	9×9; 324	11×11; 484
Training/test separation	Seed IDs differ; task content overlaps	Unlabelled tree structures are disjoint
Test inventory	100 entries, with repeated structures	100 distinct structures
Evaluation rules	Greedy and sampled	Greedy, sampled and two epsilon controls
Status	Frozen greedy criterion; exploratory sampling	Exploratory follow-up

3.1 One task, two restricted observations

The original study uses deterministic 9×9 perfect mazes: connected, acyclic corridors with farthest endpoints as start and goal. A manifest stores the layout and task settings with a checksum, so changing the observation format does not regenerate the task. The realized goal is always $(1, 1)$ in this inventory. The policy does not receive coordinates, but this convention restricts the distribution.

The agent moves forward one cell or turns left or right by $90°$. Walls block movement. Reaching the goal ends the episode; otherwise it ends after 324 actions, including turns and collisions. The reward is $+1$ at the goal and -0.001 otherwise. Training adds 0.01 on the first physical entry into each cell within an episode, under identical rules across conditions. Evaluation has no exploration bonus and makes no updates.

The local observation is a heading-aligned 5×5 numerical grid. A radius-two forward cone and wall occlusion determine visible cells; hidden cells have a distinct code. The input contains 25 values, not an accumulating map. The ray observation contains seven values: free-cell counts left, ahead and right, capped at eight and divided by eight; three corresponding goal-visibility flags; and a constant one. There is no rear ray, global goal bearing, position or heading input. Local cells supply off-axis information; rays reach farther along selected directions (Figure 1). The wall-extruded illustration in Appendix F is not an agent input.

3.2 Pretraining, transfer and independent runs

Our recurrent agent is trained with PPO (Schulman et al., 2017). It uses a linear–tanh encoder with 64 outputs, a 64-unit GRU receiving those features and the previous action, and linear actor and critic heads. The GRU carries state between decisions and resets between episodes. Eight environments collect 125-step rollouts; learning rate is 3×10^{-4} and entropy coefficient 0.01. Appendix E gives the remaining settings.

All four conditions receive 500,000 target-ray interactions (Table 2). Pretrained conditions first receive 1,000,000 source interactions. Local-to-ray transfer copies the GRU and output heads while installing a fresh ray encoder. The direct ray reference retains its whole trained network. Every condition begins target adaptation with an empty optimizer. Scratch therefore has less lifetime experience than pretrained conditions; the direct reference has both prior target-interface experience and encoder retention. It is not an upper bound or an isolated sensing control.

The obstacle-field control matches dimensions, walkable-cell count, sensor, actions, reward, architecture and interaction budget, but uses connected cyclic fields rather than perfect-maze trees. It controls for useful navigation pretraining, not irrelevant experience. Difficulty and reward exposure need not match. Matched denotes the maze family and source inventory, not guaranteed prior exposure to an identical evaluation task.

Independent training seeds 1001–1008 are used in each condition. Their labels form prespecified comparison blocks; this does not imply identical trajectories after different training histories. Train-

Table 2: Four training histories, one target-adaptation budget. New and inherited describe weights at the start of adaptation. All target-network parameters are trainable.

Condition	Source experience	Ray encoder	GRU/heads
Scratch	None	New	New
Matched local	Perfect mazes, local input	New	Inherited
Obstacle-field local	Cyclic obstacle fields, local input	New	Inherited
Ray reference	Perfect mazes, ray input	Inherited	Inherited

ing samples task seeds 0–999 and original evaluation uses 20000–20099. All 32 completed runs are retained.

Different seed labels did not ensure different mazes. A post-hoc audit finds training-inventory matches for 94/100 evaluation grids and 48/100 full grid/start/goal/heading configurations. The 1,000 training entries contain 306 distinct oriented grids and 685 full tasks. These are inventory matches, not evidence that each policy encountered or memorized them. We preserve the frozen results as an in-distribution comparison, not a test of unseen layouts. Later repair tasks have similar overlap (Appendix F). Section 4.3 describes the separate structurally held-out follow-up.

3.3 What counts as a pretraining benefit?

Every 25,000 target interactions, including zero, the frozen policy is evaluated on all 100 original task entries. The primary score averages success across the adaptation budget, rewarding early as well as final performance. For training seed i and action-selection rule m, normalized area under the success curve is

$$A_i^m = \frac{1}{500000} \sum_{j=1}^{20} \frac{s_i^m(t_{j-1}) + s_i^m(t_j)}{2} (t_j - t_{j-1}). \tag{1}$$

It ranges from zero to one and is not a final-success percentage. No checkpoint is selected by performance.

The original evaluation uses the highest-probability action (greedy). Its frozen criterion requires positive matched-minus-scratch and matched-minus-obstacle-field AUC, both 95% paired-bootstrap lower bounds above zero, and both Holm-adjusted two-sided sign-flip p-values below 0.05. The bootstrap uses 10,000 seed-block resamples and RNG 20260902; sign flips enumerate 2^8 assignments. The repository protocol was fixed before these runs, not externally preregistered.

All subsequent diagnostics are exploratory. In the complete-curve re-evaluation, each saved checkpoint receives one greedy and three sampled attempts per task. Sampling draws from the checkpoint's softmax distribution without changing its parameters or temperature. Success is per attempt, not best-of-three. Both rules have the same 324-action budget per attempt. We first calculate success per attempt within each checkpoint, then AUC within each trained seed. The eight training seeds, not their episodes or verification passes, are the independent units. Same weights do not imply the same deployed policy: greedy selection and sampling define different action rules.

4 Results

4.1 Successful pretraining does not establish the specified benefit

Matched-local greedy AUC is 0.2474, versus 0.1940 from scratch and 0.2754 after obstacle-field pretraining (Table 3). Matched minus scratch is 0.0533, with 95% interval $[-0.0432, 0.1383]$ and Holm-adjusted $p = 0.6719$. Matched minus obstacle field is -0.0281, interval $[-0.2395, 0.1785]$, adjusted $p = 0.8203$. Neither advantage criterion passes. This is not an equivalence result.

The ray reference starts at 96.5% mean success and finishes at 99.75%. Scratch and both local-transfer arms start at zero and vary substantially across seeds. Matched-local final success is 55.0%,

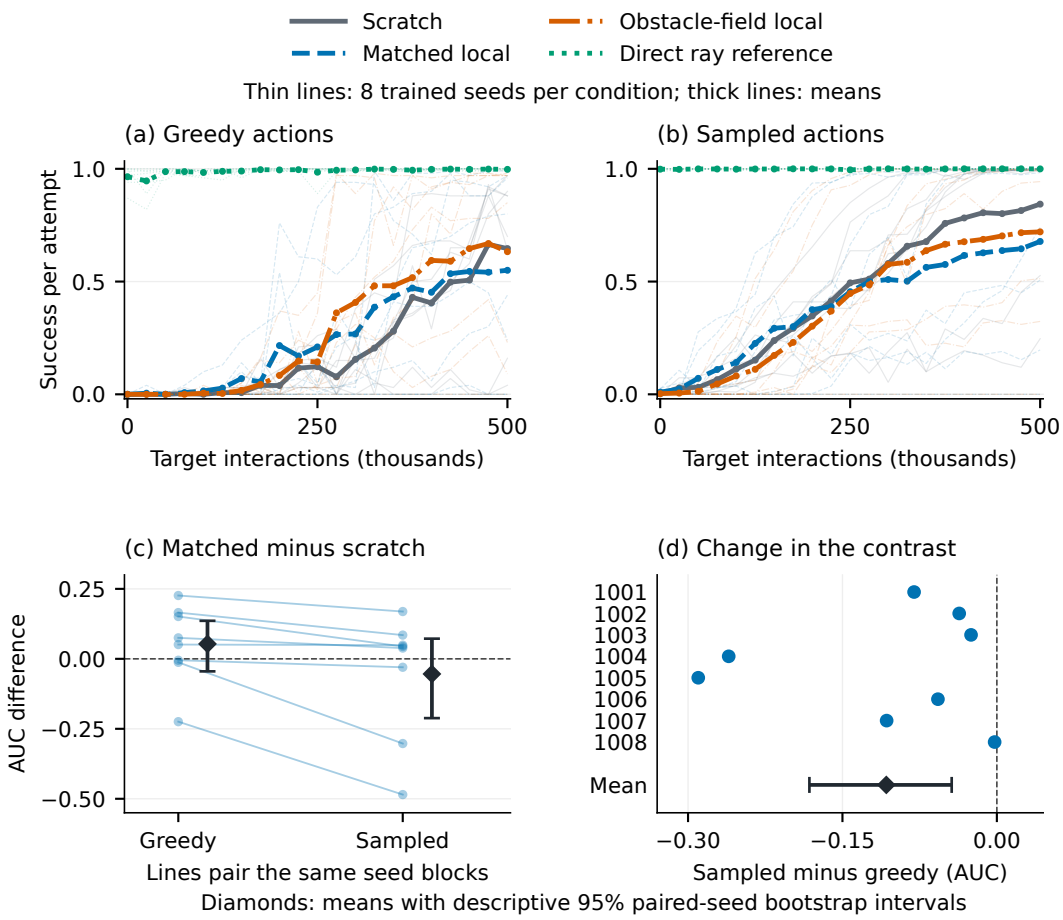


Figure 2: Action selection changes the measured matched-pretraining benefit. (a,b) Success across target adaptation; thin lines show all eight independent training seeds per condition and thick lines their means. (c) Matched-minus-scratch AUC under each rule, paired by seed block. (d) Within-block change in that contrast, sampled minus greedy. Diamonds and bars show means and descriptive 95% percentile intervals from 10,000 paired-block resamples. Each of 21 checkpoints uses 100 task entries, with one greedy or three sampled attempts per entry and a 324-action cap per attempt. Sampling is not best-of-three; checkpoint parameters are unchanged. The direct ray reference has prior target-interface experience. This exploratory comparison retains the original content overlap; the sampled contrast remains uncertain despite all eight changes being negative.

versus 64.625% from scratch: its higher mean AUC and lower final score answer different questions. The complete greedy curves and between-seed variation are retained in Appendix E.

Source learning itself was successful. With their original local encoders, all sixteen matched and obstacle-field source policies solve all 100 common perfect-maze entries; obstacle-field sources also average 99.5% on native fields. Neither inadequate source success nor the claim that the control learned nothing useful explains the target comparison. Equal source scores, however, do not establish equal strategies.

4.2 Action selection changes the estimated pretraining contrast

Does the apparent benefit depend on how the trained policy acts? Re-evaluating all 672 checkpoints changes scratch mean AUC from 0.1940 under greedy selection to 0.4492 under sampling. Matched-local AUC rises from 0.2474 to 0.3951 (Figure 2; Table 3). Sampling improves both means, but improves scratch more.

Table 3: Action selection changes the mean pretraining comparison. Normalized success AUC over 500,000 target interactions, averaged over eight training seeds per condition. Each of 21 checkpoints uses 100 task entries, with one greedy or three sampled attempts per entry. Higher is better. The sampled comparison is exploratory; neither mean ordering by itself establishes superiority. Original task-content overlap remains.

Condition	Greedy AUC	Sampled AUC
Scratch	0.1940	0.4492
Matched local pretraining	0.2474	0.3951
Obstacle-field pretraining	0.2754	0.3936
Direct ray reference	0.9904	0.9992

If sampling helped both training histories equally, their AUC difference would stay the same. We therefore calculate the change in that difference within each seed block:

$$D_i = (A^{\text{sampled}}_{i,\text{matched}} - A^{\text{sampled}}_{i,\text{scratch}}) - (A^{\text{greedy}}_{i,\text{matched}} - A^{\text{greedy}}_{i,\text{scratch}}). \tag{2}$$

The mean contrast changes from $+0.0533$ to -0.0541. Mean D_i is -0.1074, descriptive 95% interval $[-0.1821, -0.0439]$, and all eight block-level changes are negative. Thus sampling increases scratch AUC by 0.1074 more than matched-pretraining AUC. The sampled matched-minus-scratch interval itself is $[-0.2118, 0.0721]$. Consequently, the average ordering reverses, but reliable scratch superiority is not established. The matched-minus-obstacle-field interaction is $+0.0296$, interval $[-0.0734, 0.1298]$; the clearer scratch comparison does not extend to both controls.

These exploratory intervals resample the eight paired blocks 10,000 times with RNG 20260917. They do not replace the original bootstrap, multiplicity correction or failed criterion. No run is excluded: one matched and one obstacle-field policy have zero AUC under both action rules. Two complete inference passes agree, and saved-action replay plus a separate outcome recount reproduce the reported curves and contrasts (Appendix A).

The intervention establishes sensitivity of the measured comparison to the action rule. It does not show that greedy evaluation is erroneous, sampling reveals true competence, or either rule is universally better. PPO samples actions during training, but a training/evaluation difference is not itself an explanation of the outcome. Sampling can alter exploration and repeated action sequences; these results do not isolate a causal mechanism.

4.3 The reversal does not recur on structurally held-out mazes

The original inventory cannot support unseen-layout claims. Merely generating more seed labels is insufficient: a further 10,000-seed scan found no new unlabelled floor-cell tree classes beyond the original training inventory. We therefore construct a separate 11×11 study, partitioning tasks by tree structure rather than seed ID. Its 512 training, 32 validation and 100 reserved test mazes have distinct unlabelled floor-cell trees. The split is outcome-blind and also excludes two previously evaluated 11-by-11 inventories. This changes size and generator weighting as well as overlap, so between-study differences cannot be attributed to overlap alone (Appendix B).

A validation-only pilot passes its fixed learnability gate (Appendix B). The full follow-up then trains all four conditions with eight new seeds, 9201–9208. Budgets, sensors and PPO settings are unchanged, apart from the larger mazes and their 484-action cap. Training and testing both use 11×11 mazes; this is not size extrapolation. All 21 target checkpoints are evaluated on all 100 reserved tasks, with one greedy and three sampled attempts per task, repeated exactly and independently audited.

Matched-local greedy/sampled AUC is 0.2214/0.4054, versus scratch at 0.0424/0.1875 (Figure 4). The paired matched-minus-scratch contrasts are $+0.1790$ $[0.0647, 0.2832]$ and $+0.2179$ $[0.0097, 0.3886]$. Seven of eight seed blocks favour matched pretraining under both rules. Seed 9205 reverses the comparison and remains in every analysis. The primary action-selection interaction is $+0.0388$ $[-0.0847, 0.1537]$, exact two-sided sign-flip $p = 0.5547$. Thus the earlier negative interaction and mean reversal do not recur.

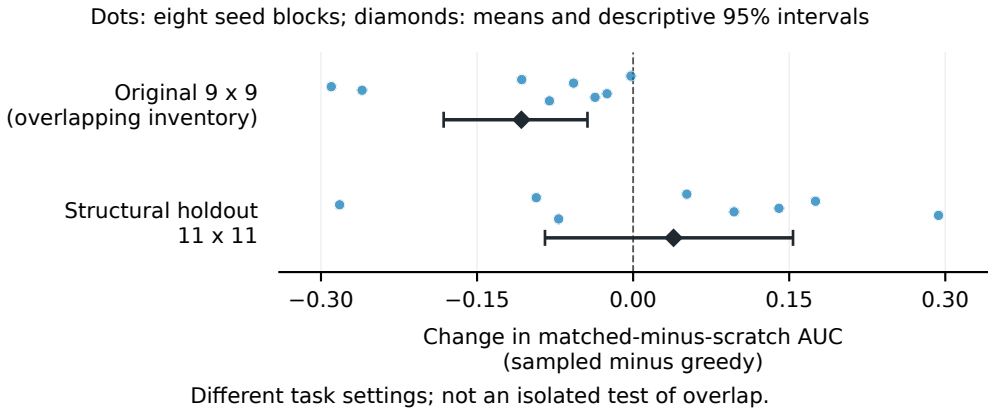


Figure 3: The original action-selection interaction does not recur in the structural holdout. Dots show all eight training-seed blocks per setting; diamonds and bars show means and descriptive 95% intervals from 10,000 within-study paired-block bootstrap draws. The horizontal axis is the change in matched-minus-scratch AUC when sampling replaces greedy selection. Negative values mean sampling helps scratch more. Both comparisons are exploratory. Size, task weighting and overlap change together; this is not an isolated test of overlap.

Matched-minus-obstacle-field is $+0.1120$ $[-0.0091, 0.2208]$ greedy and $+0.1949$ $[0.0268, 0.3738]$ sampled; the greedy comparison remains uncertain. These exploratory intervals use 10,000 paired seed-block resamples, not a new confirmatory success gate.

Figure 3 compares the two action-selection interactions directly. The greater pretraining advantage in the new setting accompanies lower scratch scores: greedy/sampled AUC falls from 0.1940/0.4492 to 0.0424/0.1875, while matched-pretraining means change from 0.2474/0.3951 to 0.2214/0.4054. This is a descriptive comparison, not an explanation of the difference between settings.

4.4 Do occasional random moves explain the sampling benefit?

We evaluate the same checkpoints with two fixed controls, without retraining. Fixed $\epsilon = 0.1$ chooses uniformly among all three actions with probability 0.1 and greedily otherwise. Its actual probability of a non-greedy move is $0.1 \times 2/3$. The more targeted *matched-departure* rule preserves the learned greedy action's probability but divides the remaining mass equally between the other two actions. It changes which alternative is chosen, not the chance of departing from the greedy action at a given state. It does not match entropy, visited states or complete trajectories. For example, action probabilities $(0.6, 0.3, 0.1)$ become $(0.6, 0.2, 0.2)$: the chance of a non-greedy action stays at 0.4, but its identity changes.

Sampling exceeds matched-departure mean AUC by 0.0404 $[0.0116, 0.0757]$ for scratch and 0.0456 $[0.0286, 0.0617]$ for matched pretraining. Learned relative probabilities of alternatives therefore help performance in this intervention. However, sampling changes the matched-minus-scratch contrast by only 0.0052 $[-0.0309, 0.0365]$ relative to matched-departure. This does not establish an effect on the pretraining advantage, equivalence, or a loop-breaking mechanism. Both controls use three attempts per task and all eight seeds; epsilon was not tuned on test outcomes (Appendix C).

4.5 What the supporting controls do and do not explain

Two controls limit the interpretation without explaining the action-selection interaction. First, replacing the input encoder while keeping local sensing unchanged reduces mean AUC from 0.9986 to 0.6276, a paired cost of -0.3710 $[-0.5640, -0.2320]$. Seven policies later recover to at least 97% success. Input replacement can therefore disrupt a successful policy on its own, but this does not apportion the local-to-ray transfer result. Repair and recoding comparisons are detailed in Appendices I–K.

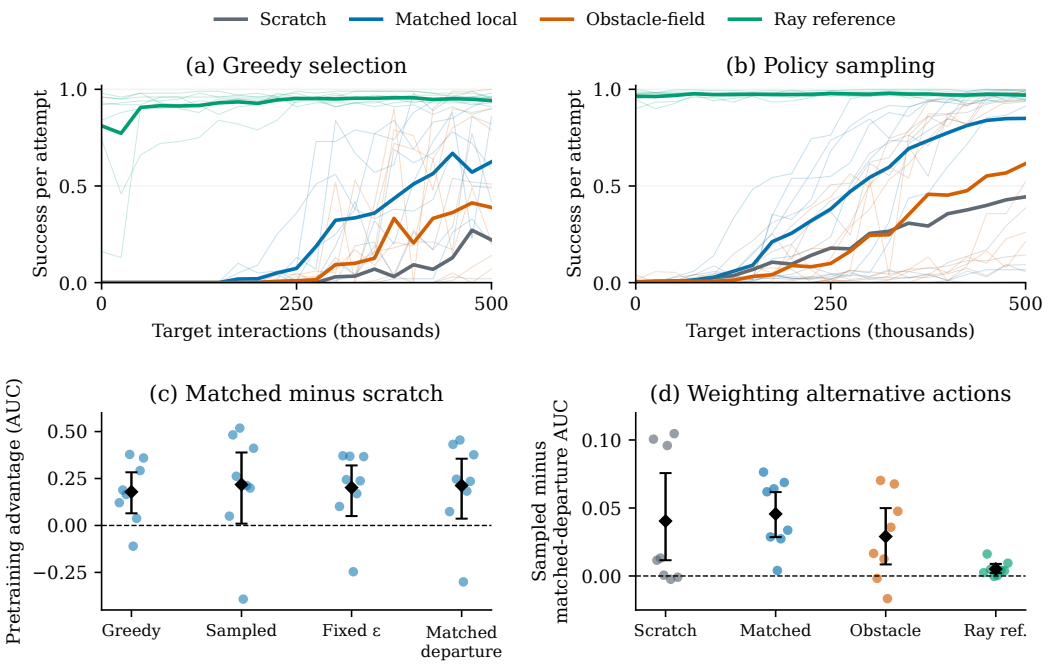


Figure 4: The original reversal does not recur on the new structural holdout. (a,b) Thin lines show all eight trained seeds per condition; thick lines show means. Each checkpoint uses 100 reserved 11×11 tasks, one greedy or three sampled attempts, and a 484-action cap. (c) Matched-minus-scratch AUC under four rules. (d) Sampling minus the control that weights non-greedy alternatives equally while preserving their total probability at each state. Dots are all eight paired seed values; diamonds and bars are means and descriptive 95% intervals from 10,000 seed-block resamples. No smoothing, checkpoint selection or retraining is used; Appendix C gives the analysis details. The study is exploratory and changes size and task weighting as well as overlap.

Second, simple goal-cued policies can solve these mazes without learning. On the new reserved tasks, a ray walker with short action history reaches 99.508% success, versus 0.061% for uniform random actions. These are same-sensor references, not memory-capacity-matched models. Original-policy challenges likewise do not exclude simple stochastic strategies. Thus neither high success nor dependence on recurrent state identifies a spatial map. Appendix D summarizes these separate diagnostic questions and preserves their full results.

5 Discussion and Limitations

In the original setting, sampling helps scratch more than matched pretraining in every paired seed block. Applying the same action rule to all conditions does not therefore guarantee that the estimated pretraining benefit is independent of that choice. The new structural holdout limits this finding: matched pretraining has a larger mean AUC than scratch under every tested rule, and its interaction with action selection is uncertain. Neither study replaces the other. The original failed advantage criterion remains failed.

The epsilon control separates two claims that raw success scores conflate. Learned alternative-action probabilities help performance beyond matching the per-state chance of a non-greedy move. Yet this benefit need not materially change the contrast between training histories. The test does not identify loop breaking or a spatial representation. Structural novelty alone does not make success a selective measure of spatial knowledge: simple goal-cued policies still solve the reserved tasks.

The experiments suggest a practical order for interpreting navigation transfer. First establish source learning and compare against useful, equal-budget pretraining. Specify exactly which network com-

ponents and observations change. Then report the deployed action rule and test whether the claimed benefit survives a relevant alternative. Audit task structure, and compare success against simple policies with access to the same cues. None of these controls substitutes for the others.

The interaction depends on the tested setting; we have not isolated why it differs between studies. Maze size, task weighting, training inventory and seeds change together. We have not tested rendered images, continuous control or robotics. Later diagnostics reuse source policies, and eight trained seeds leave substantial uncertainty in several contrasts. Confidence intervals containing zero do not establish equivalence. The follow-up's matched-minus-scratch intervals are descriptive, and the greedy matched-minus-obstacle-field interval spans zero. Neither a general transfer benefit nor its absence follows from these two studies.

Nor have we identified the contents of recurrent state. Selective spatial-history tasks would need to exclude short-history solutions. Probe control tasks test what the probe itself learns (Hewitt & Liang, 2019); interventions can test whether decoded information affects behaviour (Elazar et al., 2021). In our setting, observation-only, untrained and shuffled controls would provide additional comparisons. Adding a world-model architecture or a rendered environment would test whether the findings extend to those settings. It would not, by itself, show that the agents use spatial representations.

6 Conclusion

Reusing a navigation controller requires a comparison, not just a successful source policy. In the original setting, sampling changes the measured pretraining contrast and reverses its mean ordering. On larger, structurally held-out mazes, that reversal does not recur. Learned alternative-action weights improve mean performance beyond a matched-departure control, without clearly changing the pretraining advantage. Reporting the action rule, complete seed-level adaptation curves and task-content separation makes these distinctions visible. None alone establishes a spatial map or a general law of transfer.

AI use statement

Generative AI tools assisted hypothesis development, experimental design, implementation of agents and procedural maze generation, testing, log analysis and interpretation, literature review, figures, and manuscript drafting and revision. Automated tests and independent record audits checked the implementation and reported results. The human authors take responsibility for the final content, including AI-assisted text, code, analyses and figures.

Ethics statement

The reported experiments involve software agents in procedurally generated mazes, not human participants, patient records or deployed robots. These results do not establish spatial competence or safety in clinical or robotic applications.

Reproducibility statement

Section 3 describes the task, observations, training conditions and estimands. Appendix E records the protocol, execution revision, hyperparameters and evidence provenance. The task-content audit is in Appendix F; complete-curve re-evaluation, the structurally held-out pilot and the complete follow-up are detailed in Appendices A, B and C. The remaining appendices report the diagnostic settings and outcomes.

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*, volume 34, pp. 29304–29320, 2021. URL https://arxiv.org/abs/2108.13264.

Yunfu Deng and Josiah P. Hanna. BIFROST: Bridging invariant feature representation for observation-space sim2real transfer. *arXiv preprint arXiv:2607.01410*, 2026. URL https://arxiv.org/abs/2607.01410.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359. URL https://aclanthology.org/2021.tacl-1.10/.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep RL: A case study on PPO and TRPO. In *International Conference on Learning Representations*, 2020. URL https://iclr.cc/virtual_2020/poster_r1etN1rtPB.html.

Chaz Firestone. Performance vs. competence in human–machine comparisons. *Proceedings of the National Academy of Sciences*, 117(43):26562–26571, 2020. doi: 10.1073/pnas.1905334117. URL https://doi.org/10.1073/pnas.1905334117.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3207–3214, 2018. doi: 10.1609/aaai.v32i1.11694. URL https://ojs.aaai.org/index.php/AAAI/article/view/11694.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 2733–2743, 2019. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. DARLA: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1480–1490, 2017. URL https://proceedings.mlr.press/v70/higgins17a.html.

Sooyoung Jang, Seungho Yang, and Changbeom Choi. Stochastic policies, deterministic minds: A calibrated evaluation protocol and diagnostics for deep reinforcement learning. *International Journal of Advanced Computer Science and Applications*, 16(10):1024–1034, 2025. doi: 10.14569/IJACSA.2025.0161099. URL https://doi.org/10.14569/IJACSA.2025.0161099.

Ang Li, Huiyi Hu, Piotr Mirowski, and Mehrdad Farajtabar. Cross-view policy learning for street navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8100–8109, 2019. URL https://arxiv.org/abs/1906.05930.

Léopold Maytié, Benjamin Devillers, Alexandre Arnold, and Rufin VanRullen. Zero-shot cross-modal transfer of reinforcement learning policies through a global workspace. *Reinforcement Learning Journal*, 3:1410–1426, 2024. URL https://rlj.cs.umass.edu/2024/papers/RLJ_RLC_2024_170.pdf.

Steven Morad, Ryan Kortvelesy, Matteo Bettini, Stephan Liwicki, and Amanda Prorok. POPGym: Benchmarking partially observable reinforcement learning. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2303.01859.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16828–16847, 2022. URL https://proceedings.mlr.press/v162/nikishin22a.html.

Jurgis Pasukonis, Timothy Lillicrap, and Danijar Hafner. Evaluating long-term memory in 3D mazes. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.13383.

Andrew Patterson, Samuel Neumann, Martha White, and Adam White. Empirical design in reinforcement learning. *Journal of Machine Learning Research*, 25(318):1–63, 2024. URL https://jmlr.org/papers/v25/23-0183.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Erik Wijmans, Manolis Savva, Irfan Essa, Stefan Lee, Ari S. Morcos, and Dhruv Batra. Emergence of maps in the memories of blind navigation agents. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2301.13261.

Julian Zilly, Alessandro Achille, Andrea Censi, and Emilio Frazzoli. On plasticity, invariance, and mutually frozen weights in sequential task learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12386–12399, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/6738fc33dd0b3906cd3626397cd247a7-Abstract.html.

APPENDIX GUIDE

Complete action-selection results come first, followed by the structural split and the full held-out comparison. Supporting diagnostics retain all historical outcomes; none replaces the original failed transfer criterion.

Question or resource	Appendix
Original action-selection results	A
Structural split and validation pilot	B
Full held-out results and epsilon rules	C
Encoder and simple-policy interpretation	D
Original protocol and reproduction	E
Exact sensor inputs and content audit	F

A SENSITIVITY TO ACTION SELECTION

Does always selecting the most probable action understate performance? Two exploratory evaluations keep checkpoints, observations, rewards and the 324-action cap fixed, replacing greedy selection with draws from the unchanged softmax probabilities. No temperature is tuned and no model is trained again. For each task and policy, we compare one greedy attempt with ten sampled attempts. Success is averaged per attempt, not best-of-ten. Attempts estimate a policy's performance; the independent units remain eight training seeds per condition. Both studies were specified before their respective outcomes, after the original greedy results were known.

Constructed challenge: original source policies. The first evaluation uses all sixteen original local/ray source policies and the unchanged fifty challenge/control pairs from Appendix M. Local challenge success rises from 36.25% to 49.075%; ray success rises from 48% to 74.15%. All eight policies in each view improve. The mean gains are 12.825 and 26.150 percentage points, with descriptive 95% percentile intervals $[9.600, 16.225]$ and $[24.075, 28.575]$. These use 10,000 resamples of source policies and RNG 20260916, not resamples treating attempts as trained models. Control success rises from 52.75% to 67.575% locally and from 88.25% to 94.125% for rays. On the fixed twenty-one route-matched challenge pairs, sampled success is 39.226%/74.821%.

The sampled policies still fall below their goal-aware walker references at 99.85%/99.87%. Observed periodic trajectory tails decrease and visited-floor coverage increases, but successful early termination changes both measurements. Neither a hidden-state cycle nor causal mediation by looping is identified. The two identical passes contain 35,200 episodes. Complete action-probability and geometry replay, a separate success recount and all 1,600 original greedy trace matches pass. All source scores and provenance are available in `reports/sampled_challenge_2026_09_11.json`.

Original target tasks: final adapted policies. The second evaluation uses all thirty-two original final target policies, after 500,000 target interactions, on the same 100 original target tasks. It does not evaluate intermediate checkpoints. Table 4 reports all condition means. Scratch, matched and obstacle-field policies improve by mean 19.150, 12.4625 and 9.5375 points, respectively. Matched seed 1004 and obstacle-field seed 1002 remain at zero and are retained. The direct reference had prior target-format experience and retained its encoder; it is not an oracle or an equal-experience causal baseline.

The fixed descriptive contrasts pair training-seed labels across conditions. Matched-minus-scratch changes from -9.625 greedy points to -16.3125 sampled points; the change is -6.6875, interval $[-16.8269, 2.5506]$. Matched-minus- obstacle-field changes from -8.250 to -5.325 points; the change is 2.925, interval $[-7.6250, 12.8381]$. These percentile intervals use 10,000 resamples of eight blocks and RNG 20260917. Sampled endpoint contrasts themselves have intervals $[-47.6625, 11.6000]$ and $[-48.5409, 36.8891]$, respectively. Neither an advantage nor equivalence follows. Unchanged mean final rankings do not establish unchanged learning curves. The original failed transfer gate remains unchanged, as do its information mismatch and training-content overlap.

Table 4: Sampling increases mean final success without establishing a matched pretraining advantage. Success percentages average eight policies per condition, 100 tasks per policy, with one greedy or ten sampled attempts per task and the same 324-action cap. These are per-attempt rates, not best-of-ten scores. Exploratory endpoint sensitivity does not measure adaptation AUC.

Condition	Greedy (%)	Sampled (%)
Scratch	64.62	83.78
Matched local pretraining	55.00	67.46
Obstacle-field pretraining	63.25	72.79
Direct ray reference	99.75	99.91

Two identical passes contain 70,400 episodes. Complete recorded-probability RNG and geometry replay and a separate scalar recount pass. All 3,200 original greedy task metrics match per pass; the original logs lack complete traces, so historical action-sequence equality is not claimed. Verification passes are not extra experimental samples. CPU execution took 604.52 seconds, excluding the subsequent offline audit; no GPU allocation or retraining was used. All thirty-two source summaries, checkpoint hashes, runtime and contrasts are released in `reports/final_sampling_2026_09_11.json`. These checks share the neural implementation; they do not independently validate its learned representation.

A.1 Complete learning-curve sensitivity

The endpoint study cannot establish whether action selection changes adaptation efficiency. We therefore separately fix all 21 adaptation checkpoints from each of the same 32 runs, at 0, 25,000, . . ., 500,000 interactions. On the same 100 tasks, each checkpoint receives one greedy and three unchanged-softmax attempts per task, each capped at 324 actions. No training or temperature tuning occurs. Success is per attempt, not best-of-three. The endpoint is re-evaluated within this design, not pooled with the earlier ten-attempt study.

For each trained seed, we integrate success over all checkpoints by the trapezoidal rule and divide by the 500,000-interaction budget. Table 3 reports these normalized AUCs, not final-success percentages. The descriptive comparisons use eight paired training-seed blocks, 10,000 percentile bootstrap draws and RNG 20260917. This differs from the original confirmatory bootstrap seed; its intervals and gate remain unchanged.

The condition-level AUC table is reported in the main text (Table 3); all per-seed curves remain in the public data.

Matched-minus-scratch changes from 0.0533 to -0.0541 AUC. The change is -0.1074, interval $[-0.1821, -0.0439]$, and is negative in all eight paired blocks. The sampled contrast itself has interval $[-0.2118, 0.0721]$, so a reversed mean ordering does not establish reliable scratch superiority. Matched-minus-obstacle-field changes from -0.0281 to 0.0016; its change is 0.0296, interval $[-0.0734, 0.1298]$. Its sampled contrast has interval $[-0.2975, 0.2954]$. Neither establishes a matched advantage or equivalence. The information mismatch, content overlap and unequal direct-reference experience remain. Action-selection sensitivity does not identify a spatial representation or the mechanism of the original transfer failure.

Both full inference passes agree across all 672 checkpoints: 537,600 episodes in total. One copy of each trace plus its repeated-pass digest is stored. The built-in and a fresh frozen offline audit each replay all 268,800 stored episodes; a separate exporter recounts outcomes and independently reconstructs explicit trapezoidal AUCs and paired intervals. Historical greedy task metrics match at every checkpoint, without implying historical action-trace equality. All 32 sources, including the two zero-AUC policies, remain in the public data. Execution takes 7,267.00 seconds on one CPU thread, excluding the offline audit, with no new GPU allocation. Complete counts, seed curves, provenance and limits are in `reports/curve_sampling_2026_09_12.json`.

Table 5: One scratch pilot on structurally held-out validation mazes. All rows use the same 32 mazes and a 484-action cap. Counts are successful attempts, not distinct mazes or trained replicates. The learned policy uses ray input; reference walkers use short history. No reserved-test or four-condition transfer result is shown in this pilot table.

Final policy or reference	Successful attempts	Success (%)
Scratch, greedy	6/32	18.75
Scratch, sampled	79/96	82.29
Uniform random actions	19/32,000	0.0594
Goal-cued ray walker	31,913/32,000	99.73
Goal-cued local walker	31,935/32,000	99.80

B STRUCTURALLY HELD-OUT VALIDATION PILOT

The new inventory contains 644 distinct unlabelled floor-cell trees, one representative per structure. Candidate seeds 60000–69999 are considered in ascending order. A SHA-256 hash of a fixed prefix and the canonical tree key assigns each structure to one of ten buckets: eight for training, one for validation and one for test. The first 512, 32 and 100 eligible structures are selected, excluding tree classes in the earlier 11-by-11 diagnostic inventories 40000–40099 and 50000–50099. These are explicit exclusions, not a claim of novelty relative to every development artifact. Regenerated manifests and rotation/reflection checks verify separation. The canonical tree ignores its embedding in the grid, so different rotations of the same maze cannot cross roles.

The new mazes contain 49 floor cells, versus 31 in the original study. Deduplication changes the generator's weighting; the role-level oracle route lengths are retained in `configs/disjoint_v1_split.json`. All routes fit the 484-action cap. This establishes solvability, not PPO learnability. Size, task weighting and separation change together, preventing an isolated causal estimate of removing original content overlap.

The pilot trains one scratch ray-input policy, seed 9101, for 500,000 interactions on the 512 training tasks. PPO settings, sensors and training bonus are unchanged except for the 484-action budget. Greedy validation uses all 32 tasks at zero and every 25,000 interactions. Before learning and at the final checkpoint, a separate frozen evaluation uses one greedy and three unchanged-softmax attempts per task. Initial counts are 0/32 and 0/96; final counts are 6/32 and 79/96. Greedy success is zero through 300,000 interactions; the remaining counts at 325,000 through 500,000 are 2, 1, 0, 2, 5, 3, 5 and 6. No sampled intermediate curve is inferred from those greedy checkpoints.

The gate, fixed before training, requires final sampled validation success of at least 20% and an improvement of at least ten percentage points over the initial checkpoint. It passes. This engineering threshold is neither a significance test nor evidence of pretraining benefit. The pilot seed is excluded from the subsequent four-condition comparison. No reserved-test efficacy result is reported in this pilot subsection. This pilot passes its preset learnability threshold, not a transfer criterion. It supplies one policy, not 96 independent replications; sampled intermediate checkpoints were not measured. It does not replicate the pretraining interaction. Initial validation success is zero under both action rules. The local walker uses a different sensor and is included only as context.

The three untrained validation references use 1,000 attempts on each of the same 32 tasks at the same 484-action cap. Uniform random actions solve 19/32,000; the local goal-cued walker solves 31,935/32,000 and the ray walker 31,913/32,000. The ray walker is the same-sensor reference for the pilot; the local walker is separate observation context. All are reported without selecting a favourable learned checkpoint. Reference counts are in `reports/disjoint_gate_calibration_2026_09_13.json`.

All training records, checkpoint identities and validation-task multiplicities pass the audit. Two inference passes reproduce the 256 stored endpoint traces. Fresh Mac and Linux checks reproduce every action sequence and outcome; probability values differ across hosts by at most approximately 2.2×10^{-7}. Each host repeats its own inference exactly. Cross-host bit identity is not claimed, and verification passes are not independent policies. A prior allocation failed during startup unit tests before any pilot training; it is preserved in the execution report rather than counted as a failed

learning seed. Full receipts are `reports/disjoint_pilot_35516327_macos_audit.json` and `reports/disjoint_pilot_35516327_linux_audit.json`.

C Complete structurally held-out comparison and epsilon controls

The full follow-up retains all four conditions and eight independent training seeds 9201–9208. Pretrained arms receive one million source interactions and all arms receive 500,000 target interactions. All training and reserved testing use 11×11 mazes from the split in Appendix B; the 512 training, 32 validation and 100 reserved test mazes have disjoint tree shapes. Between-study differences cannot be attributed to overlap alone. The full plan was fixed before these training runs. Training-validation means were inspected before the epsilon plan was fixed; no reserved-test result was used to select either control. The whole follow-up remains exploratory.

Complete evaluation and reconstruction. All 696 training checkpoints and 160 sealed record shards passed the training audit before reserved inference. Each of 21 adaptation checkpoints uses all 100 test tasks: one greedy attempt and three independent attempts under each of sampling, fixed epsilon and matched-departure. Every attempt has 484 actions. Two inference passes agree exactly on the execution host; this is not a claim of cross-platform bit identity. Independent saved-action audits reconstruct seeded choices, movements, collisions, rewards and termination, then recount success and AUC. Source diagnostics use the 32 validation tasks, with additional native-field diagnostics for obstacle pretraining. The inventory totals 1,352,192 learned inference episodes including repeat passes, saved in 1,376 files. Verification passes are not additional independent replicates.

Action rules. Let p be the policy probabilities and g their first argmax. Fixed epsilon uses $q(g) = 0.9 + 0.1/3$ and $q(a \neq g) = 0.1/3$. Matched-departure uses $q(g) = p(g)$ and $q(a \neq g) = (1-p(g))/2$. This is state-dependent epsilon-greedy with $\epsilon = 3(1-p(g))/2$, not a tuned epsilon schedule. It matches departure probability conditional on a state, not entropy, recurrent state, visitation or trajectories. Every learned and action-rule probability is logged, and the seeded choice is reconstructed. The epsilon RNG identity is `[20260919, run, step, task, control, rollout]`; the control order is fixed epsilon followed by matched-departure. No optimizer is restored or updated.

Analysis. AUC is integrated separately for every training seed. Greedy/sampling contrasts use 10,000 seed-block bootstrap draws with RNG 20260923; epsilon contrasts use RNG 20260919. Their slightly different endpoints for shared comparisons reflect the fixed bootstrap draws, not different observations. All intervals are descriptive, not multiplicity-adjusted claims. The full-study primary contrast is Equation 2; its eight values are $(-0.0716, 0.2935, 0.0970, 0.1402, -0.2821, 0.1753, -0.0931, 0.0516)$, with exact two-sided sign-flip $p = 0.5546875$. The epsilon diagnostic compares matched-minus-scratch under sampling versus matched-departure; its mean is 0.005219 $[-0.030948, 0.036531]$. Fixed epsilon gives a corresponding difference of 0.016250 $[-0.046848, 0.076856]$. Neither identifies a reliable change in the pretraining contrast. Table 6 retains every trained seed; Table 7 reports final success separately from AUC.

Source learning and simple-policy references. Common-maze source validation greedy/sampled success is 99.609%/100% for matched local, 100%/100% for obstacle-field local, and 83.594%/97.656% for the ray reference. Obstacle sources also solve 100% of native-field attempts under both rules. These are validation diagnostics, not reserved-test results. On the reserved test set, each untrained reference makes 1,000 attempts per maze. Uniform actions solve 61/100,000 attempts (0.061%); the goal-cued ray walker solves 99,508/100,000 (99.508%); the local walker solves 99,491/100,000 (99.491%). Both walkers avoid collisions; mean episode lengths are 112.319 and 113.569 actions, versus uniform at 483.954. An independent repeat reconstructs all 300,000 reference episodes and agrees with every task summary. The local walker has a different sensor, and neither walker is a blind random-action policy.

Table 6: All independent training seeds in the held-out follow-up. Normalized success AUC over 500,000 target interactions. No seed is excluded; matched seed 9205 has zero greedy AUC. The ray reference retains prior ray experience and its encoder. Higher AUC is better, but mean ordering alone is not a significance test.

Condition	Seed	Greedy	Sampled	Fixed ϵ	Matched departure
Scratch	9201	0.1085	0.4736	0.2933	0.3730
	9202	0.0000	0.0452	0.0112	0.0337
	9203	0.0030	0.0630	0.0239	0.0498
	9204	0.0000	0.0192	0.0078	0.0187
	9205	0.1103	0.4043	0.2511	0.3084
	9206	0.0000	0.0302	0.0111	0.0327
	9207	0.1175	0.4471	0.2637	0.3424
	9208	0.0000	0.0172	0.0042	0.0181
Matched local	9201	0.2298	0.5232	0.3947	0.4468
	9202	0.1888	0.5275	0.3825	0.4656
	9203	0.1680	0.3250	0.2678	0.2963
	9204	0.3782	0.5377	0.3764	0.4736
	9205	0.0000	0.0120	0.0047	0.0080
	9206	0.0375	0.2430	0.1809	0.2157
	9207	0.4098	0.6462	0.5012	0.5775
	9208	0.3595	0.4283	0.3708	0.3946
Obstacle-field local	9201	0.0000	0.0340	0.0103	0.0174
	9202	0.0000	0.0014	0.0016	0.0032
	9203	0.1087	0.3342	0.2388	0.2639
	9204	0.0067	0.0497	0.0373	0.0372
	9205	0.0080	0.0730	0.0439	0.0897
	9206	0.2605	0.3392	0.2765	0.3033
	9207	0.2587	0.4677	0.3674	0.4000
	9208	0.2330	0.3844	0.2662	0.3368
Ray reference	9201	0.9570	0.9702	0.9571	0.9678
	9202	0.8610	0.9715	0.9303	0.9553
	9203	0.9585	0.9770	0.8577	0.9721
	9204	0.9477	0.9755	0.9580	0.9757
	9205	0.9802	0.9925	0.9631	0.9922
	9206	0.9160	0.9542	0.8444	0.9499
	9207	0.9330	0.9722	0.9470	0.9683
	9208	0.8705	0.9733	0.9316	0.9639

Reproduction and failures. The immutable evaluation revision is `94af1f93891f623128492ea389744ce1c8db1078`. All three evaluation waves and the final trace/reference audit complete with exit zero. Their allocation totals 59.5622 CPU-core-hours, with no GPU or new training; the separate training array used 17.675 allocated GPU-hours. Earlier timing-admission and environment-startup failures are preserved and produced no reserved outcomes. No gate was weakened, no failed learning seed was re-run, and no partial result selected the analysis. Public seed curves, contrasts, reference task counts and output hashes are in `reports/heldout_evaluation_2026_09_26.json`. `scripts/publish_heldout_results.py` independently checks publication arithmetic; `scripts/build_heldout_figure.py` rebuilds Figure 4 without private checkpoints. Raw-record integrity and fresh inference audits remain distinct from these public-summary checks.

Table 7: Held-out final success (%), averaged over eight independent training seeds. Each frozen policy uses 100 tasks, one greedy or three stochastic attempts per task. Sampling is per attempt, not best-of-three. These are endpoints, not learning-curve scores.

Condition	Greedy	Sampled	Fixed ϵ	Matched departure
Scratch	22.125	44.375	36.458	40.417
Matched local	62.500	84.958	73.208	82.667
Obstacle-field local	38.875	61.583	50.125	58.083
Ray reference	94.125	97.000	93.458	96.333

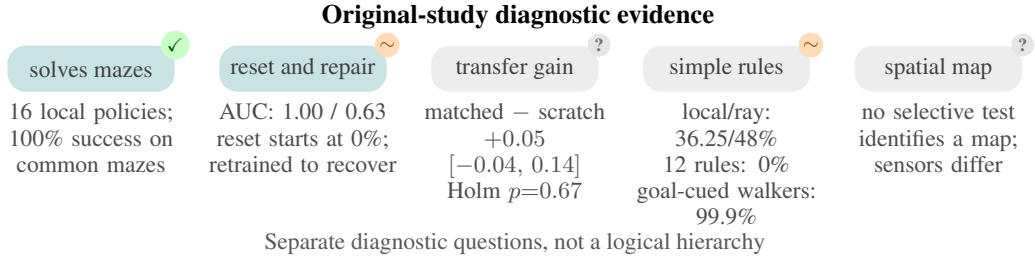


Figure 5: What the diagnostics establish and what they leave open. These are separate questions, not successive tests of a spatial map. Source success uses sixteen local policies on common mazes. Reset-and-repair compares retained versus fresh local encoders: recovery requires further training. Transfer shows the original matched-minus-scratch greedy AUC contrast, its 95% paired bootstrap interval and Holm-adjusted p-value. On challenge mazes, local/ray policies score 36.25/48%; twelve rules score zero, but goal-cued short-history walkers reach approximately 99.9%. The experiments use different task sets; these numbers are not a shared ranking.

D Interpretation controls and original-study summary

D.1 Input replacement is a competing explanation, not a diagnosis

The local-to-ray transition changes both sensing and the encoder. To test whether the encoder change alone can matter, we compare retaining and replacing it while leaving local sensing unchanged. Each pair begins with the same inherited GRU and heads and an empty optimizer; all parameters may adapt. Retained/reset mean AUC is 0.9986/0.6276. The paired reset cost is -0.3710, descriptive 95% interval $[-0.5640, -0.2320]$. Every reset policy scores zero initially and at 25,000 interactions, although seven later reach at least 97% success.

Replacing the encoder can therefore impair a successful policy even when the sensor is unchanged. This control does not measure how much of the local-to-ray result the encoder change explains. A separate same-ray reset is much less costly. Further controls compare which parts of the network need to learn during repair. Updating the whole network outperforms updating only the encoder on average, across two repair starts. Some policies also recover while the fresh encoder remains fixed (Appendices I–K). Recovery alone therefore does not show that the inherited controller contains reusable spatial knowledge, or identify why a particular reset failed.

D.2 Simple navigation provides a demanding reference

Success above a weak baseline need not indicate a complex spatial strategy. On larger perfect mazes, previous-action wall followers solve every tested task, including those where learned ray policies degrade (Appendix L). Clearing learned recurrent state impairs source success, but that intervention can also disrupt short action routines; it does not make spatial memory necessary.

A constructed set of 50 challenges excludes twelve named deterministic rules. Original local and ray source policies solve 36.25% and 48.00%, respectively. Sampling raises these means to 49.08% and 74.15%, still below goal-aware random walkers at 99.85% and 99.87%. These references use visible obstacles and goal cues, with previous-action history and, for the local walker, a one-observation buffer. They are not blind random-action baselines or memory-capacity-matched models. Beating the selected deterministic rules therefore does not establish navigation beyond simple stochastic routines (Appendix M). These separate challenge results contextualize success, not the original pretraining AUC contrast.

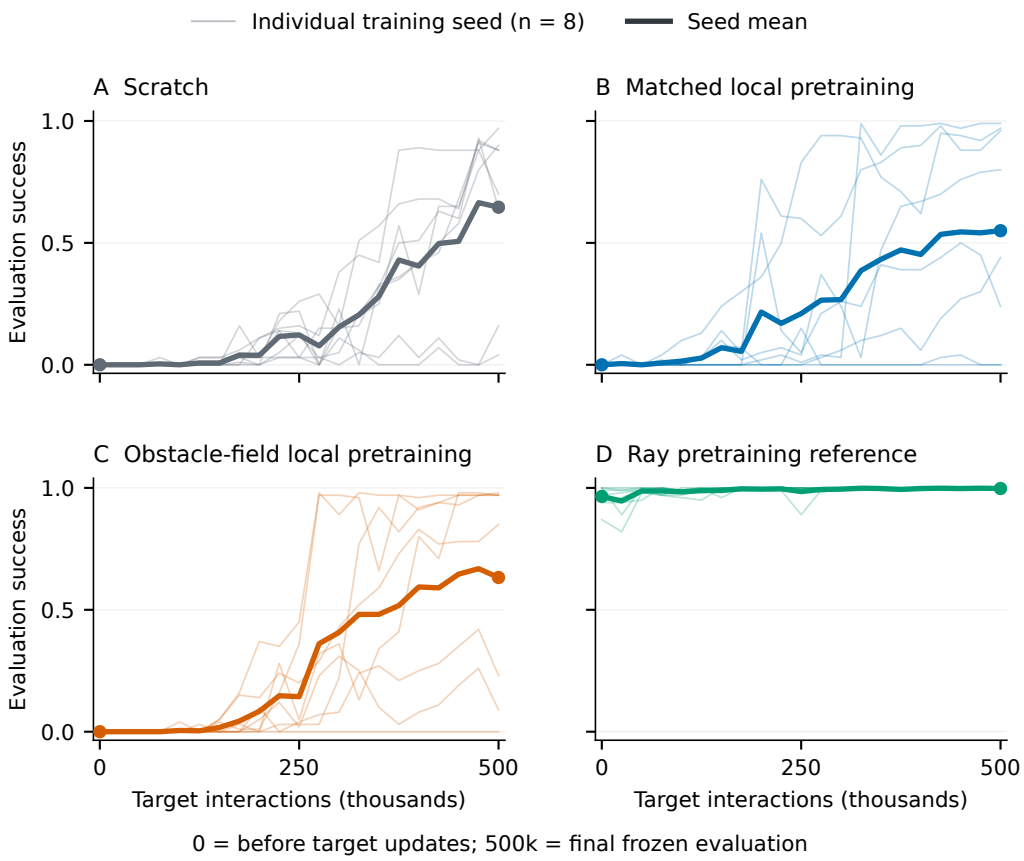


Figure 6: The frozen comparison does not establish a matched-pretraining benefit. Thin lines are all eight training seeds per arm; thick lines are means. Each point evaluates 100 content-overlapping entries. The axis counts only target adaptation. The ray reference has prior target-interface experience and retains its trained encoder, so its high score does not isolate sensing. Neither planned AUC contrast passes the advantage criterion.

Table 8: Frozen outcomes across eight training seeds per arm. Parentheses: sample SD across seeds. Success is a proportion after 500,000 target interactions; normalized AUC is the primary endpoint.

Condition	Final success	AUC
Scratch	0.6463 (0.3556)	0.1940 (0.1150)
Matched local	0.5500 (0.4332)	0.2474 (0.2091)
Obstacle-field local	0.6325 (0.4418)	0.2754 (0.1880)
Ray reference	0.9975 (0.0046)	0.9904 (0.0066)

E Reproduction and evidence provenance

The frozen protocol has identifier `protocol-a16832ea32f0930a`. Its configuration uses schema `m2e.protocol/v1` and version `confirmatory-v1`. The confirmatory training revision is:

`391267aa6e1d1a15c4f45446053bef2bf1311b2e`.

Task manifests bind layouts and settings to checksums. Diagnostics use all source-final checkpoints, not checkpoints selected for their diagnostic scores.

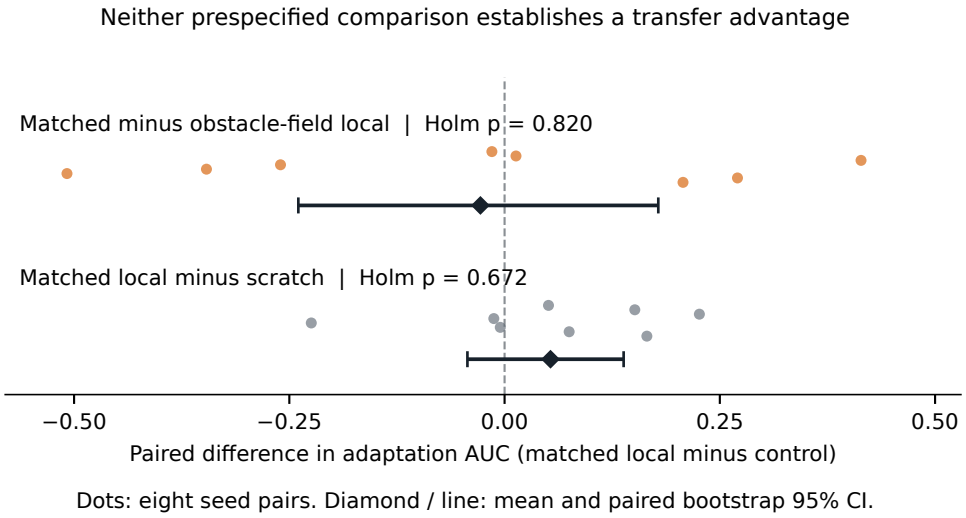


Figure 7: Neither planned AUC contrast establishes an advantage. Small points show all eight paired training-seed differences; large points show their mean. Horizontal intervals are prespecified 95% paired bootstrap intervals; the vertical dashed line is zero. The p-values use exact two-sided sign flips with Holm correction across the two comparisons. Both intervals span zero. Failing the advantage criterion does not establish no effect or equivalence.

The PPO discount is 0.99, generalized-advantage parameter 0.95, policy clipping coefficient 0.2, value-loss coefficient 0.5, and maximum gradient norm 0.5. Each rollout contains 125 transitions from each of eight environments, with four update passes per rollout. The optimizer is Adam. The encoder, 64-unit GRU, and heads remain trainable during adaptation. Hidden states are carried during execution and reset between episodes; evaluation is greedy and frozen. Optimizer moments are not transferred. Target models are initialized before source training; the local-transfer procedure retains the fresh target encoder and copies the trained GRU and actor/critic. The direct reference copies the whole source network. Original adaptation does not separately reseed the action random stream. Policy inputs exclude privileged state returned for logging.

Local cell codes are wall 0, floor 1, goal 2, agent 3, and hidden 4, divided by four after flattening. Of the 25 local slots, 16 are outside the forward cone and always hidden; walls can hide additional slots. Ray depths count free cells before a wall, capped at eight and divided by eight, rather than continuous distance to a wall surface. There is no rear ray. Local and ray encoders have different input parameter counts; the shared GRU and output heads have identical dimensions.

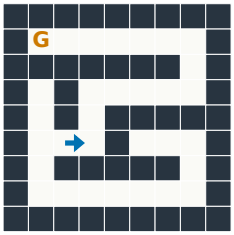
The implementation uses Python, NumPy, Gymnasium and PyTorch; manifests record versions and device identity. Of 696 audited checkpoints, 672 accompany target evaluations and 24 are final-source checkpoints. Hashes and 160 record seals establish artifact integrity, not independent evidence of transfer.

Curated seed-level figure data are stored in `paper/tables/plotted_seed_data.json`. The companion `figure_provenance.json` records relative input names and hashes. The builder checks that all 672 evaluations recover the seed-level AUC summaries and paired effects before plotting. Individual curves are retained; episode-level confidence intervals do not substitute for between-seed uncertainty. Raw checkpoints and generated run directories are separate from the curated paper inputs.

The new diagnostic figures use `paper/tables/diagnostic_plot_data.json` and its hashed provenance ledger. The builder checks the completed encoder audit, all sixteen sealed evaluation shards, 336 evaluation points and paired summaries before curating them. Both figure builders support regeneration from curated inputs without raw run directories. The maze illustration stores its task, pose, observation values and deterministic selector in `paper/tables/maze_illustration.json`.

F TASK-CONTENT AUDIT AND GEOMETRY ILLUSTRATION

(a) Evaluator-only maze



(b) Local grid: 25 values

.25	0	0	1	1
1	.25	.25	0	1
1	1	.75	1	1
1	1	1	1	1
1	1	1	1	1

Forward is up

(c) Ray vector: 7 values

	Left	Front	Right
Depth / 8	0	0.125	0
Goal flag	0	0	0
Constant		1	

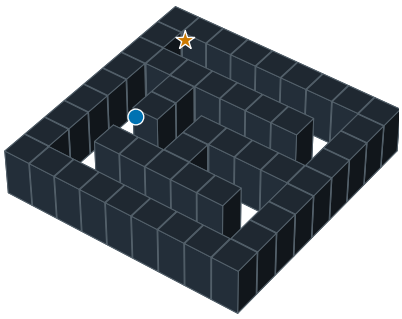
Local codes: wall 0; floor .25; goal .5; agent .75; hidden 1

Figure 8: Exact inputs for the example in Figure 1. The reader-only layout, wall-occluded local cells and numerical rays share the pose after 18 of 37 oracle actions for seed 20000: row 5, column 2, east. This is not a learned trajectory. Rays encode three depths, three goal flags and a constant; the rays drawn in Figure 1 are schematic. The two sensors contain unequal information.

Content equality excludes generation-seed metadata. A grid key contains its shape and row-major occupancy; a full-task key also contains start, goal and initial heading under the shared environment rules. We do not identify rotated or reflected mazes. The original generator and saved inventories were checked against the execution revision; all 32 runs' 1,100 saved task identifiers regenerate exactly. Thus overlap is present in the historical experiments, not introduced by a later generator change.

For example, evaluation seed 20000 and training seed 672 have the same grid, start $(5, 5)$, goal $(1, 1)$ and heading 2. Their manifest identifiers differ because the seed is included in the identifier. The overlap fractions reported in Section 3 count evaluation entries, not unique keys. The audit does not determine which tasks each learner encountered or establish memorization. No entries or frozen results were removed after this discovery.

Same maze with walls raised: geometry illustration only



Illustrated agent pose ★ Goal

Not a policy input; no rendered-3D experiment is reported.

Figure 9: The same maze with walls extruded. Geometry uses exactly the grid and illustrative pose from Figure 1; location markers are overlaid annotations. This drawing is not a rendered policy input. No image-based 3D experiment or physical simulation contributes to the results in this paper.

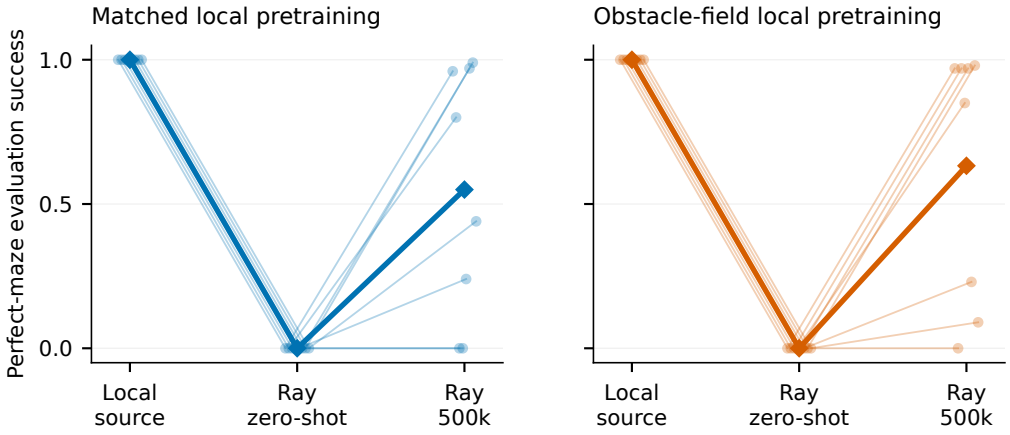


Figure 10: Source success does not ensure immediate expression under rays. Thin lines link each training seed's source-final success on the common local-view evaluation tasks, zero-update ray success and final ray success; diamonds show means. Each score uses 100 entries. All sixteen source policies achieve 100/100 locally and zero immediately after local-to-ray transfer. Source evaluation is exploratory; target scores are from the unchanged frozen comparison. These are content-overlapping tasks. Observation and encoder changes occur together, so the drop does not isolate either cause.

G Frozen recurrent-state diagnostic details

The sixteen source-final checkpoints are evaluated on CPU with one PyTorch thread and deterministic algorithms. Checkpoint hashes and seals are checked before and after evaluation. Each mode uses the same local observations, task seeds 20000–20099 and 324-action budget; output replacement is refused.

Table 9: Exploratory hidden-reset success by training seed (percent). Normal execution is 100% throughout. Hidden-and-action reset has the same per-seed success as hidden-only reset. Trained policies, not their 100 episodes, are the independent descriptive units.

Source training seed	1001	1002	1003	1004	1005	1006	1007	1008
Matched, hidden reset	3	3	0	0	3	0	3	0
Obstacle field, hidden reset	3	3	3	0	3	0	3	0

Mean episode length rises from 42.67 to 319.685 actions for matched sources and from 40.985 to 318.60625 for obstacle-field sources under hidden reset. Normal and intervention evaluations all have zero collisions. These statistics describe failure to finish, not proof of a particular strategy. Short repeated cycles in inspected failures are compatible with disrupted action sequencing. Removing all carried state is not a selective spatial-content intervention.

H Relocated-goal diagnostic

After identifying the fixed realized goal, we evaluated all sixteen frozen local source policies on one relocated goal for each of the 100 common test mazes. Topology, start, heading, observation function, and action budget were unchanged. NumPy PCG64 with seed sequence [20260905, maze seed] selected a relocated cell from row-major walkable candidates, excluding the start and original goal, independently of policy outcomes. This exploratory design was fixed before its outcomes but reused already inspected topologies.

Matched-source success was 99.25%: seven seeds solved all mazes and seed 1004 solved 94%. All obstacle-field sources achieved 100%. However, the unchanged original-goal trajectories already crossed the relocated cell in 93.375% and 94.25% of episodes, respectively. This incidental-crossing reference does not require reacting to the new goal. Mean shortest action distance, including turns, fell from 35.04 to 18.03. Observed mean actions fell from 42.67 to 27.205 for matched sources and 40.985 to 23.93 for obstacle-field sources. Thus high relocated-goal success occurs with closer targets that are often encountered along existing routes. Even additional success can result from continuing beyond the now-nonterminal original goal. It is not decisive evidence of active goal tracking.

An independent repeat reproduced all 32 case files (3,200 episodes) and 200 task manifests byte for byte; original-goal episodes also matched the source reference. A stronger test would balance action distance and require goal-dependent route choice before incidental encounter. These results do not establish arbitrary start-goal generalization or a spatial-map mechanism.

I Completed ray-encoder intervention

This exploratory resolving test was specified after the confirmatory and source-diagnostic results. It uses all eight final ray-source policies from the reference arm. For each policy, two new runs copy identical trained GRU and actor/critic parameters. One keeps the trained ray encoder; the other uses the fresh ray encoder from the same historical seed's scratch zero-update checkpoint. Both have new empty optimizers, new episodes, zero hidden states, and the same new adaptation random seed, equal to the historical training seed plus 20000. Only encoder origin differs at initialization. Equal initial random streams do not imply equal trajectories after different policies select different actions.

All sixteen runs retain the ray observation, perfect-maze generator, training tasks, optimizer settings, and 500,000-interaction adaptation budget. Evaluation follows the same 21-checkpoint schedule. The primary descriptive contrast is reset-minus-retained AUC, paired by source policy, with all eight differences and a 10,000-resample percentile interval using random seed 20260905. Zero-update loss, recovery, and final success are secondary descriptions. Reused historical seeds and tasks make this a post-hoc mechanism study, not a replacement confirmatory test.

All sixteen runs completed with no exclusions. Independent audits verified 336 checkpoints and evaluations, 33,600 evaluation episodes, 42,304 records and 80 sealed shards. Execution revision was

`bce84fbd7a0e6dd442ed21572e1e9cb63ba0d7a5`.

Mean AUC (sample SD across eight source policies) is 0.98740625 (0.00931007) under retention and 0.97221875 (0.00912261) after reset. The paired difference is -0.0151875, with descriptive percentile interval $[-0.0205000, -0.00993671875]$. No confirmatory significance claim is attached to this post-hoc interval. Every source policy has one reset initialization; these eight pairs do not estimate initialization sensitivity conditional on a fixed source controller.

Table 10: Completed encoder-intervention AUC by source training seed. All pairs use unchanged rays and 500,000 new interactions. Negative differences indicate lower AUC after reset; they do not describe persistent failure at the final checkpoint. Values are rounded only for display.

Seed	1001	1002	1003	1004	1005	1006	1007	1008
Retained	.9843	.9908	.9918	.9828	.9980	.9955	.9683	.9880
Reset	.9705	.9870	.9728	.9580	.9720	.9783	.9623	.9770

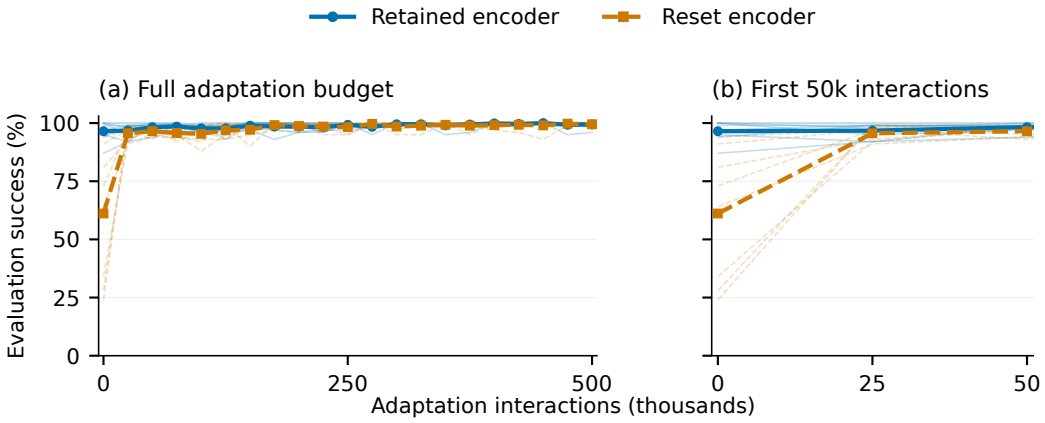


Figure 11: Ray-encoder recovery occurs by the first post-update checkpoint. (a) Full budget; (b) first 50,000 interactions, on the same success scale. Thin lines are eight source-policy pairs; marked lines are means, 100 tasks per checkpoint. Both arms retain ray sensing and train all parameters. This exploratory result does not identify the original local-transfer mechanism.

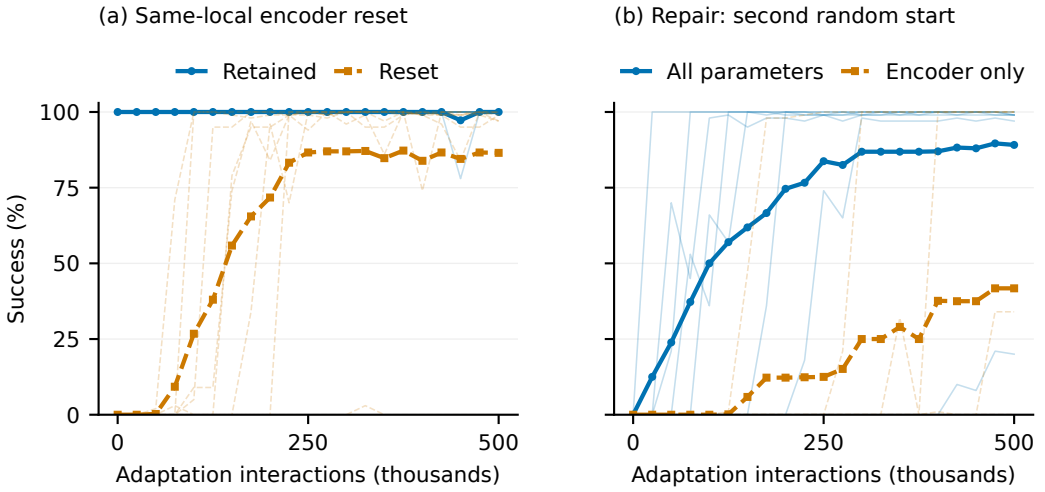


Figure 12: Unchanged local sensing permits disruption and variable repair. (a) Retained versus reset encoder, all parameters trainable. (b) Second repair random start: identical fresh initial networks, all parameters versus encoder only trainable. Thin lines show all eight original source policies per arm; marked lines are means. Each point uses 100 content-overlapping tasks, at the same 21 checkpoints over 500,000 interactions. Panels use different initializations and are not a paired cross-panel contrast. The second start favours whole-network AUC in all eight pairs, but final encoder-only recovery occurs for several policies. These are exploratory local-input experiments.

J LOCAL DISRUPTION AND TWO REPAIR RANDOM STARTS

The same-local retained/reset control uses all eight original matched-local source policies. Fresh encoder initialization is source seed plus 40000; adaptation randomness is source plus 30000. Both arms use task seeds 0–999 for training and 20500–20599 for evaluation, with the unchanged local sensor and 500,000-interaction schedule. The reset-minus-retained AUC interval uses 10,000 source-pair resamples, RNG 20260906. Execution revision:

`984b652b8e6b61cbb985e773ffb0b56f5d0c311f`.

All sixteen runs pass complete audits; retained/reset mean AUC is 0.998625/0.627625, SD 0.003889/0.270928. All retained policies start and end at 100%; all reset policies start at zero, and final reset success averages 86.5%.

The repair studies use fresh contemporaneous arms, not the reset arm above as a historical comparator. Both arms start with identical entire network, optimizer, environment and random state. Whole-network training updates all parameters; encoder-only training freezes the GRU and actor/critic tensors but retains gradient flow through them. The encoder is 25-to-64; sensing, rewards, tasks and budgets remain unchanged. The first repetition uses encoder seed source plus 60000 and adaptation seed source plus 50000. Its execution revision is

`6a002eb53e4e3bcc7254f6a84e57fc85133b7622`.

The second uses offsets 90000 and 100000, respectively, at revision

`bb51fa07a39cd762988983f59144597b5a9cb783`.

Each repetition includes all sixteen runs, 336 checkpoints/evaluations and 33,600 evaluation episodes. Independent artifact audits verify complete sealed records, source origins, actual paired initial-state equality and bit-identical controller tensors at every encoder-only checkpoint. Separate audit executions reproduce their JSON and curve CSV outputs; this repeats the same auditor rather than creating a second implementation. No unsuccessful policy is excluded.

Table 11: Repair AUC for every original source under two random starts. A: all parameters trainable; E: encoder only. Repetitions share pretrained sources. The source-1005 reversal remains visible rather than being replaced.

Source	First A	First E	Second A	Second E
1001	.56550	.00000	.63725	.43550
1002	.72425	.00000	.49225	.02600
1003	.67300	.00000	.78475	.00000
1004	.82900	.00000	.87175	.69650
1005	.00000	.55100	.88250	.00000
1006	.10700	.00000	.02450	.00000
1007	.54350	.00000	.81825	.24100
1008	.46575	.00000	.97500	.00000

In the first repetition, whole-network mean AUC is 0.488500 and encoder-only mean AUC is 0.068875. Final success is 87% and 12.5%, respectively. The encoder-only-minus-all contrast is -0.419625, descriptive 95% interval $[-0.663224219, -0.096864844]$, 10,000 draws, RNG 20260907. Second-repetition AUC is 0.68578125/0.174875 (SD 0.307380/0.264272), final success 89.125%/41.75%, contrast -0.51090625, interval $[-0.739627344, -0.281519531]$. All second-start policies initially score zero; at 25,000 steps only all-parameter source 1008 succeeds, on all 100 tasks.

For the combined comparison, average the two paired contrasts within each source, then resample eight sources, not sixteen source-start observations. The mean is -0.465265625, interval $[-0.61265625, -0.299140625]$. Both new-only and combined intervals use 10,000 draws, RNG 20260911, specified before the second outcomes. They describe these sources and two starts, not a full distribution of initialization effects. Both encoder initialization and adaptation randomness change across starts. The known 93-grid/50-full-task overlap prevents novel-layout claims.

J.1 Repair beyond the training size

We evaluate all 32 final policies from the two encoder-only/whole-network repair starts and all eight original local sources on 100 tasks at each size 9, 11 and 13 (seeds 50000–50099). No training or checkpoint selection occurs. Greedy policies retain the local sensor and 324-action cap, with memory reset only between episodes. All oracle routes and a previous-action local reference succeed within budget. Size-9 entries overlap training in 97 grids and 70 full tasks; larger entries have no exact grid/full-task overlap with the five checked earlier inventories. Size also changes difficulty and inputs, not novelty alone.

Table 12: Whole-network repair's mean advantage extends to larger local mazes. Success percentages average eight sources, each evaluated on 100 tasks per size. Starts share sources; original policies are context, not equal-compute controls. Small encoder-only size losses reflect failure floors, not superior retention.

Policy	9×9 (%)	11×11 (%)	13×13 (%)
Original local	100.000	99.750	99.125
Whole-network, start 1	85.875	82.375	78.875
Encoder-only, start 1	12.500	12.500	12.500
Whole-network, start 2	86.875	80.750	74.875
Encoder-only, start 2	40.500	40.000	39.500

The primary size-13 comparison averages encoder-only-minus-whole-network contrasts within each source across the two starts. Whole-network/encoder-only success is 76.875%/26.0%, with a paired difference of -50.875 percentage points and 95% interval $[-70.8125, -30.9375]$. Smaller size losses in encoder-only policies reflect existing failure floors, not better retention. The analysis resamples those eight source-level differences using 10,000 draws, RNG 20260912, and a descriptive percentile interval. Source-ordered differences (1001–1008) are $-40, -66.5, -90, -41, +1, -35.5, -45, -90$ percentage points. The source-1005 mean of $+1$ combines opposite $+100$ and -98 outcomes, not a stable exception. Whole-network/encoder-only success falls by 9.5/0.5 points from size 9 to 13. We retain unconditional success, including zero-score policies, rather than selecting successful repairs or normalizing by their small-maze scores. The original sources' higher mean does not identify forgetting.

Two CPU passes each evaluate 12,000 episodes; all 120 case files match exactly. A separate checker reconstructs all 24,000 saved action sequences without calling the environment transition implementation. Generation/oracle dependencies remain shared. The fixed plan, all 120 policy-size rows and evidence hashes are in `reports/repair_retention_2026_09_09/`.

J.2 Learning behind a fixed fresh encoder

The new restriction uses all eight original local sources, with encoder seeds source plus 110000 and adaptation seeds source plus 120000. Each pair starts with identical network, optimizer, environment, recurrent and random states. One arm trains all parameters; the other fixes the fresh 25-to-64 encoder while training GRU and action/value heads. Sensors, reward, task inventory and 500,000-interaction budget match. Evaluation remains every 25,000 interactions on tasks 20500–20599. This comparison is not pooled with historical encoder-only starts, and the 93-grid/50-full-task overlap still applies.

Whole-network/controller-only mean AUC is 0.4800/0.4062; final success is 62.25%/75.5%. We retain AUC as primary despite the opposite ordering at the endpoint. The rounded paired AUC interval is $[-0.2495, 0.1271]$; this establishes neither superiority nor equivalence. Recovery with a fixed encoder is possible, not a generally superior method.

The eight C-minus-A AUC differences have mean -0.0738125, sample SD 0.28574751, descriptive 95% interval $[-0.24946875, 0.127125]$, using 10,000 paired-source resamples and RNG 20260913 fixed before outcomes. Arm AUC SDs are 0.40495529 and 0.27576464. Both arms initially score zero and remain there at 25,000 steps. All 336 unsmoothed curve points are published with the 16-policy table in `reports/controller_repair_35146566/`.

Table 13: Recovery with a fixed new encoder is possible but variable. A: all parameters; C: controller only. Each source contributes one matched pair; final success uses all 100 tasks. AUC remains the primary endpoint despite the opposite ordering of final means. All zero-success policies remain included.

Source	A AUC	C AUC	C−A AUC	A final (%)	C final (%)
1001	.66600	.47425	-.19175	100	97
1002	.82800	.32200	-.50600	100	96
1003	.00000	.51050	+.51050	0	100
1004	.70850	.54725	-.16125	98	99
1005	.00000	.01225	+.01225	0	17
1006	.00000	.00000	.00000	0	0
1007	.72050	.63375	-.08675	100	99
1008	.91700	.74950	-.16750	100	96

All sixteen jobs completed, using 4.91917 allocated GPU-hours plus a 0.13306-hour validation smoke. Two complete artifact audits reproduce exactly across all 336 checkpoints/evaluations, 33,600 episodes and 80 sealed streams. Initial payloads match; every controller-only encoder stays bit-identical while controllers update in both arms. A separate scalar checker reconstructs episode metrics and the paired analysis. Unlike the retention check, this is not action-sequence replay; both full audits reuse the same auditor. Successful recovery behind a fixed encoder rejects a universal requirement to update both components, not alternatives involving optimization, capacity or clipping.

K EQUAL-WIDTH INFORMATION-PRESERVING RECODING

For each of three depth counts $d \in \{0, \dots, 8\}$, thermometer coding uses eight bits $b_k = \mathbb{1}[d \geq k]$, $k = 1, \dots, 8$. The three goal flags follow these 24 bits; the original constant one is implicit. The inverse recovers each normalized depth as $\sum_k b_k/8$. Padding instead appends twenty zeros to the original seven values. Exact-inverse trajectory replay and analytical encoder composition are engineering positive controls, not learned alignment methods. Both study arms start with identical fresh 27-to-64 encoders and inherited ray controllers; all parameters adapt.

All eight ray sources receive initialization offset 70000 and adaptation offset 80000. Training uses tasks 0–999; evaluation uses 20500–20599, 100 episodes at each of 21 checkpoints, with the same overlap caveat. The primary descriptive thermometer-minus-padded contrast uses 10,000 paired draws, RNG 20260908. All sixteen runs pass complete audits at execution revision

`707617df18030a1c2fd1238d457d9602d3733cac`.

Table 14: Recoding AUC and paired differences for all sources. Equal width does not match feature scaling, active coordinates or effective optimization.

Source	Padded	Thermometer	Thermometer minus padded
1001	.74000	.00025	-.73975
1002	.67800	.97200	+.29400
1003	.51225	.85150	+.33925
1004	.00000	.81850	+.81850
1005	.52575	.04450	-.48125
1006	.72250	.81850	+.09600
1007	.61500	.00000	-.61500
1008	.69650	.87400	+.17750

Mean final success is 85.75%/70.875%, and mean zero-update success is 1%/0% for padded/thermometer arms. Neither replaces the fixed AUC endpoint. One adaptation random start per source cannot distinguish stable source susceptibility from stochastic variation. No trial selects a preferred encoding per source for its aggregate.

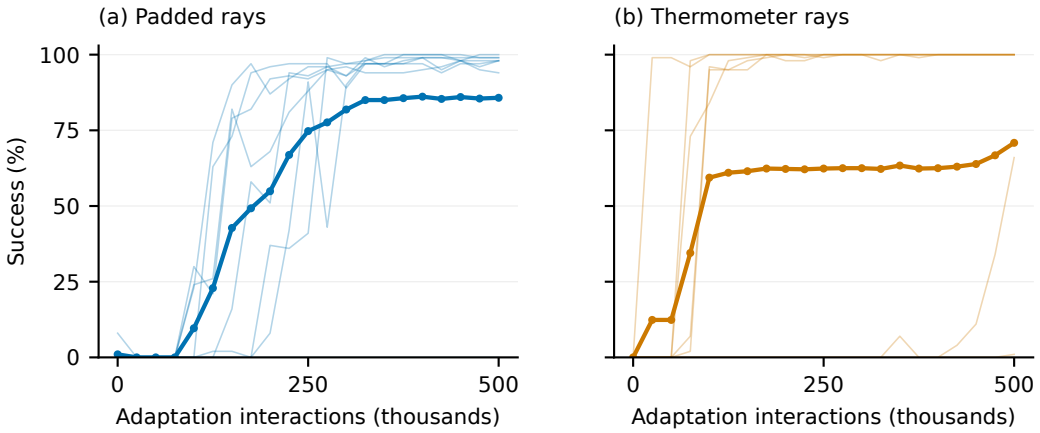


Figure 13: Equal information and input width do not remove variable adaptation. (a) Padded rays; (b) lossless thermometer rays. Thin lines show every original source policy, thick marked lines the eight-source means. Both arms have fresh paired 27-input encoders, with all parameters trainable. Each checkpoint uses 100 content-overlapping tasks. The wide paired interval spans zero; these exploratory curves establish neither a mean winner nor equivalence.

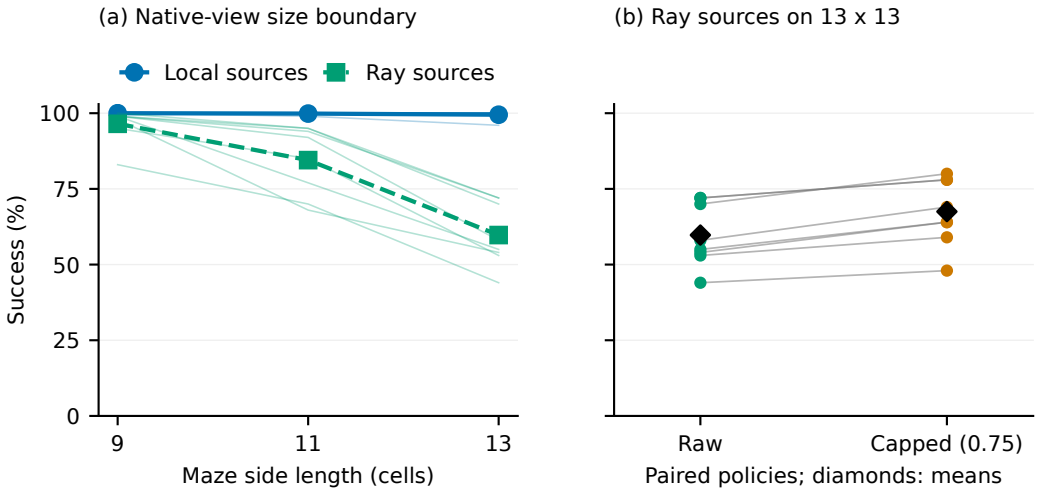


Figure 14: Ray-source size degradation is only partly repaired by depth capping. (a) All eight sources per native view, thin lines; source means, marked lines. Different training histories prevent a causal local-versus-ray interpretation. (b) The same eight frozen ray policies on 13-by-13 tasks, raw versus depth-only cap 0.75; links pair sources and black diamonds mark means. Each point uses 100 tasks and a 324-action limit without learning. All source means improve, but some individual episodes worsen. The cap is lossy, and these exploratory size tests do not identify spatial maps.

L SIZE, DEPTH SUPPORT AND SHORT-HISTORY REFERENCES

The frozen size diagnostic uses task seeds 40000–40099 separately at sizes 9, 11 and 13. Sixteen original source-final policies remain in their native views with no retraining. Each size uses 324 actions; secondary allowances are 484 at size 11 and 676 at size 13. In all 3,200 fixed/scaled episode pairs, success and distinct floor cells visited are unchanged. The four prespecified within-view

size differences use 10,000 source-paired bootstrap draws and RNG 20260909. Ray 13-minus-9 success is -36.75 percentage points, interval $[-41.375, -31.625]$; local is -0.5, interval $[-1.5, 0]$. The 8,000-episode diagnostic repeats identically and passes an independent episode-consistency and summary audit. It does not independently replay actions.

The depth-cap diagnostic records complete trajectories for eight ray policies, three sizes, two input arms and 100 tasks each: 4,800 episodes per pass. Enumeration of 124,000 floor-position/heading states in the 1,000 original training tasks verifies a maximum normalized depth 0.75. Only larger depth values are capped. All size-nine trajectories are identical; raw metrics reproduce the size diagnostic. At size 11, raw/capped success is 84.5%/85.125%, paired gain 0.625 points, interval $[-0.625, 2.125]$; 13 episodes are rescued and eight harmed. At size 13, gain is 7.75 points, interval $[6.125, 9.375]$; 66 are rescued and four harmed. These descriptive intervals use 10,000 source resamples, RNG 20260910. All source means improve only at size 13.

Two complete inference passes reproduce every trace and metric. A separate audit replays actions in the maze, reconstructs observations and the cap, checks termination, rewards and summary arithmetic, and verifies normalized probabilities and greedy actions. It is not a second network implementation. Of 322 raw size-13 failures, 55 never encounter an above-bound depth; familiar marginal values can still occur in unfamiliar combinations or histories.

The left-hand reference uses current rays and the previous action; the right reference reflects its directional inputs, actions and previous-action state. Both solve all 300 tasks at 324 actions, raw or capped. These are achieved short-history performances, not a proof of minimal memory or an optimal ceiling. Along learned trajectories, each reference receives the learned agent's actual previous action. A junction has at least three floor neighbours; arrival is the decision immediately after a successful forward move into it. Turns while occupying a junction are not additional arrivals.

Agreement pools decisions within source, then averages eight source rates; long failures can dominate, so this is not task-balanced agreement. For the 55 unexposed size-13 failures, left/right agreement is 84.65%/86.05% over all actions and 39.86%/59.65% at junction arrivals. Both directions and all sources are retained. Outcome-defined subsets are descriptive, not randomized treatments. The same analysis repeats exactly on all 4,800 saved traces; independent aggregation and a separately written reflected-rule truth table check arithmetic and reflection, not strategy identity.

M Named rules and stochastic navigation references

After constructing and freezing fifty challenge/control pairs using hand-coded rules only, we evaluate all eight original local and eight original ray policies. Each challenge defeats all twelve named rules but is reachable within the 324-action budget. Controls share geometry but change goal placement; median optimal routes are ten versus seven actions. This is a deliberately selected stress test, not a random sample or an isolated topology manipulation. All sixteen policies solve some challenges, whereas each named rule solves zero. This excludes the tested rules as complete behavioural accounts; more elaborate reactive or history-dependent routines remain possible. Different information and training histories prevent causal local-versus-ray interpretation.

Table 15: Original-policy successes out of fifty tasks per role. Every source is retained. L: local; R: ray; challenge and control differ in goal placement and difficulty. Repeated deterministic passes are checks, not independent policies.

Seed	L challenge	L control	R challenge	R control
1001	18	24	26	41
1002	14	23	22	43
1003	18	25	20	48
1004	24	31	17	44
1005	20	29	30	42
1006	18	25	22	43
1007	17	29	27	45
1008	16	25	28	47

Mean challenge/control success is 36.25%/52.75% locally and 48%/88.25% for rays. On the fixed twenty-one pairs whose optimal routes differ by at most two actions, rates are 23.81%/61.31% and 46.43%/92.86%, respectively. Matching route lengths does not remove selection or goal-difficulty confounds. Two complete CPU passes reproduce all 3,200 episodes, including probabilities. Saved-action replay checks transitions and outcomes; a separate scalar recount checks aggregation. These share the released simulator, not a second policy implementation. Complete source/rule comparisons and provenance are in `reports/shortcut_evaluation_2026_09_10/`.

Stochastic calibration. The twelve rules are deterministic. To test whether exceeding them establishes navigation beyond simple stochastic routines, we fix three references before their outcomes are computed. Two choose uniformly among open directions while retaining the corresponding follower's turning, dead-end handling and visible goal cue. The ray walker stores the previous action; the local walker also buffers one observation. These are not maps, blind random actions or a memory-capacity match to the learned recurrent network. A third reference chooses forward, left or right uniformly without observing the maze.

Table 16: Simple goal-aware walkers outperform the learned policies on the frozen challenge. Success percentages under the same 324-action cap; references use 1,000 rollouts per task, learned rates average eight sources. Each role contains 50 tasks; the route-matched subset contains 21 pairs. This selected inventory does not estimate generalization, and the views contain different information.

	Challenge	Control	Matched challenge	Matched control
Local walker	99.85	100.00	99.67	100.00
Ray walker	99.87	100.00	99.74	100.00
Uniform actions	37.25	54.12	41.05	44.05
Learned local	36.25	52.75	23.81	61.31
Learned ray	48.00	88.25	46.43	92.86

Every source falls below its view's walker reference range on both roles and matched subsets. All eight local sources lie within the uniform-action range on the full challenge and control halves. These central 95% ranges describe hypothetical independent task successes at the reference's per-task rates, not sampling distributions for learned-policy scores or equivalence tests. Similar scores do not identify similar strategies. Goal-cue dependence is untested; cycles, goal placement and difficulty remain unresolved explanations.

This post-hoc calibration does not change H2. Its design was committed before reference outcomes. Linux and Windows runs produced identical released results and analyses; each platform re-simulated all 300,000 episodes. A reported 6,000-episode sampled replay used H2's evaluator without MazeEnv; the committed regression check covers 600 episodes. Analysis reconstruction, file-hash and sampled-replay tests were also checked locally, without repeating the full runs. Public counts, analysis and verification metadata are in `reports/random_walk_reference_2026_09_11/`.

N Recovery with inherited or fresh controllers

Eight contemporaneous blocks compare inherited and fresh controllers behind the same fixed new local encoder. Controller includes the recurrent network and action/value heads. Initial optimizer, environment and recurrent states match; only controller tensors differ. Both arms receive 500,000 adaptation interactions, not equal lifetime training: inherited controllers had one million prior steps. There is one fresh start per block. All sixteen runs and all twenty-one evaluation checkpoints remain, including failures. Evaluation overlaps training in 93/100 grids and 50/100 full tasks; this is not unseen-layout transfer.

Primary normalized AUC is 0.3951875 inherited versus 0.20340625 fresh. The paired difference is 0.19178125, descriptive 95% percentile interval $[-0.02240859, 0.40325156]$ using 10,000 draws over eight blocks, RNG 20260914. Four pairs favour inheritance. Final success is 91%/74.875%; six fresh controllers reach 95–100%, while the other two reach 6% and zero. Both arms start at zero and remain at zero at 25,000 steps. Recovery alone therefore does not establish that inherited training

was necessary. Neither superiority nor equivalence follows from the uncertain primary comparison. Complete curves and all-source values are in `reports/inheritance_35284051/`. All 336 checkpoints, 33,600 evaluation episodes and eighty sealed streams passed state/scalar audits twice; a separate parser reconstructs the estimates. These training audits are not saved-action replay.

An exploratory follow-up selected after these ordinary-maze outcomes evaluates all sixteen final controllers on the unchanged challenge inventory. No checkpoint is chosen by performance. Inherited/fresh challenge success is 34%/30.75%; the fixed primary paired difference is 3.25 percentage points, descriptive 95% interval $[-5.5, 14.0]$, with four positive, three negative and one tied block. The bootstrap uses 10,000 draws and RNG 20260915. Control success is 55%/50.25%; length-matched challenge rates are 27.38%/24.40%. All sixteen policies solve some challenges, but prior controller training does not establish a reliable advantage here. The two complete passes reproduce 3,200 episodes and pass saved-action replay. This uses the shared simulator; there is no separate scalar recount for this follow-up. Safe complete source counts and paired analysis: `reports/inheritance_challenge_2026_09_11.json`. No result in either study identifies spatial maps or reverses the original failed transfer gate.

O CURATED EVIDENCE FOR THE ADDITIONAL FIGURES

The two-study interaction plot (Figure 3) is generated by `scripts/build_study_comparison.py` from the hash-pinned public original and held-out summaries. It reconstructs all checkpoint AUCs, all eight within-study interactions and their descriptive bootstrap intervals before plotting. It neither pools studies nor tests a causal effect of removing overlap.

`scripts/build_sprint_figures.py` reads hash-pinned completed audits and summaries for local disruption, repair replication, recoding, size and clipping. It reconstructs displayed means and normalized curve areas before writing `paper/tables/sprint_plot_data.json`. Every source is included. The companion `sprint_figure_provenance.json` records input and output hashes, plotting-code identity and runtime. The `--from-curated` route rebuilds the figures from a clean clone without private checkpoints; it validates the curated data, not the underlying training. Same-runtime redraw checks are separate from the completed raw-run audits. Cross-platform font/rendering differences may change file bytes without changing the plotted numbers.