RECURRENT STRUCTURE REVEALS HOW GOAL-CONDITIONED POLICIES FAIL

Anonymous authors
Paper under double-blind review

ABSTRACT

Failed trajectories may follow different paths yet converge to the same persistent outcome. We propose a behavioural interpretation of goal-conditioned policy failure whose unit is a recurrent outcome of the frozen policy–environment dynamics, its incoming basin, and its operating probability mass. This identifies which starts share a failure, where behaviour becomes trapped, and how consequential that failure is. In finite deterministic systems, the closed-loop map yields an absorbing goal outcome or non-goal fixed points and cycles.

Across FourRooms and MiniGrid KeyDoor, this representation reveals failure concentration and task-stage bottlenecks. In the larger 16×16 KeyDoor setting (MG16), exact analysis identifies 1,755 goal-conditioned failure structures across 84 right-room goals. Discovery from 32 starts per goal—8.2% of the evaluated start population—recovers 85% of pooled recurrent failure mass and identifies the dominant failure in 98.8% of goal–trial pairs. With 64 starts, mean per-goal coverage reaches 91.3%. Discovery uses sampled executions; exact enumeration supplies evaluation ground truth.

The representation also guides intervention. Sample-selected FourRooms edits redirect incoming basins toward success, while iterative MiniGrid repair addresses downstream failures. Recurrence-guided exploration and replay improve terminal right-goal success over ordinary continuation in all three tested MG16 seeds, with heterogeneous gains. These results support the recurrent outcome, basin, and mass as a recoverable and actionable unit of behavioural interpretation.

1 INTRODUCTION

Success rate and return measure how well a goal-conditioned policy performs, but reveal little about how its failures are organized. The same failure rate can arise from many distinct persistent behaviours or from a few outcomes reached from different initial states. Behavioural interpretability asks what these outcomes are, which starts lead to them, and how much failure they account for.

We address these questions through the recurrent closed-loop behaviour of a frozen policy. In a finite deterministic environment, a deterministic policy induces a functional graph over augmented states. With success represented as absorbing, every trajectory eventually reaches the goal or a non-goal fixed point or cycle. Each outcome has an incoming basin of states that eventually reach it. Apparently different failed trajectories can therefore share the same persistent behaviour.

We propose the recurrent outcome, its incoming basin, and its operating mass as the unit of explanation for persistent goal-conditioned policy failure. This complements explanations of individual action choices and abstract policy summaries by connecting trajectories through their eventual outcomes. We study this representation in 12×12 FourRooms (FR12), compare Medium and Good checkpoints in 11×9 MiniGrid KeyDoor (MG11), and examine a larger 16×16 setting (MG16). In KeyDoor, the representation also locates failures relative to key acquisition, door opening, and right-room navigation. The graph mathematics is established; our contribution is this interpretive construction, its recovery from sampled executions, and its use for diagnosis and intervention.

Exhaustive enumeration limits the practical reach of such analysis. We therefore retrieve recurrent failures from sampled executions, using exact censuses only afterwards to evaluate recovery. In MG16, 32 sampled starts per goal—8.2% of the evaluated start population—recover 85% of pooled

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.

recurrent failure mass; 64 starts recover 91.3% on average across goals. Independent held-out executions assess coverage of a frozen discovered set. Substantial mass recovery need not imply discovery of every rare structure or formal verification.

We test actionability through local repair and continued learning. Sample-selected FR12 edits redirect incoming basins toward success, while iterative MG16 repair exposes downstream failures requiring renewed diagnosis. Recurrence-guided continuation uses discovered failures to focus exploration and replay without supplying correct actions. It improves terminal right-goal success over ordinary continuation in all three tested seeds and exceeds random triggering in two. The gains are heterogeneous, and unequal realized exposure means the comparison evaluates the combined protocol rather than localization alone.

Our contributions are threefold:

1. **Behavioural interpretation.** Recurrent outcomes and basins identify shared persistent failures, quantify their operating mass, and reveal sequential bottlenecks.
2. **Recovery from limited executions.** Sampled discovery recovers consequential failure mass, evaluated against exact structural ground truth.
3. **Actionable diagnosis.** Local repair and continued-learning experiments test whether discovered structures provide useful intervention targets.

2 Related Work

Policy explanations and behavioural structure. Abstracted Policy Graphs and subsequent policy summaries describe learned behaviour through abstract states and transitions (Topin & Veloso, 2019; Climent et al., 2021; Boggess et al., 2022). More recently, SILVER constructs interpretable policies using Shapley-based explanations (Li et al., 2025). Our unit of explanation instead groups distinct trajectories by their eventual recurrent outcome and attaches the operating mass of its incoming basin, revealing which starts share a persistent failure and how consequential it is.

Recurrent policy–environment dynamics. Li et al. (2026) identify stable cyclic structures in recurrent neural policies interacting with their environments, relating their geometry to memory, robustness, and adaptation. Both approaches examine structure arising during closed-loop interaction. Our focus is non-goal recurrence as a persistent failure outcome, its goal-conditioned incoming basin, and the recovery of its mass from sampled executions. We do not analyse hidden-state geometry or require a recurrent neural architecture.

Sampled analysis and verification. Statistical model checking detects recurrent components from simulated executions (Daca et al., 2016). AEGIS-RL combines trajectory-derived explanatory graphs, probabilistic verification, risk-guided falsification, and runtime shielding (Le et al., 2026). It is a close precedent for connecting behavioural explanations to failure discovery and intervention. Our objective is to decompose persistent goal-reaching failure into recurrent outcomes and measure their incoming probability mass, rather than check temporal safety specifications. Held-out assessment concerns coverage of a frozen discovered set; it does not establish exhaustive discovery or formal verification.

Failure-guided intervention. Exploration, curricula, and replay can concentrate learning on selected states or experiences (Ecoffet et al., 2021; Florensa et al., 2017; Andrychowicz et al., 2017). Recent approaches include FEMA, which retrieves stored failure experiences (Miao, 2026), and MEDS, which clusters recurring errors and penalizes prevalent error patterns during LLM reinforcement learning (Liu et al., 2026). Our selection signal is a recurrent class detected through augmented-state repetition under a frozen deterministic policy. Repeated error patterns across episodes need not constitute recurrence in this dynamical sense. We use discovered recurrent locations to guide local edits or exploration and replay, and rediscover them as the policy changes. The continuation results evaluate this combined protocol, without isolating the effect of localization alone.

3 BEHAVIOURAL RECURRENCE IN GOAL-CONDITIONED POLICIES

Let $\mathcal{X}$ be a finite augmented state space containing the variables needed to determine both the environment transition and the policy's action, g a fixed goal, T the deterministic transition function, and π_θ a frozen deterministic policy. Their closed-loop map is

$$F_g(x) = T(x, \pi_\theta(x, g)).$$

Since every augmented state has one successor, F_g induces a functional graph. With success represented as an absorbing goal outcome, every trajectory eventually reaches either the goal or a non-goal fixed point or cycle.

For a recurrent outcome R, its basin $B_g(R)$ contains all states whose trajectories eventually enter R, including the states in R itself. Given an operating start distribution μ_g, its basin mass is

$$m_g(R) = \Pr_{x \sim \mu_g}[x \in B_g(R)].$$

Thus a small recurrent structure can account for substantial failure when many starts flow into it. We call non-goal recurrent outcomes *recurrent failures*; their masses sum to the goal's recurrent failure mass M_g^-. For $M_g^- > 0$, the within-goal share $q_g(R) = m_g(R)/M_g^-$ measures the fraction of recurrent failure attributable to R.

These objects provide complementary interpretive information: R identifies the persistent behaviour, $B_g(R)$ identifies the starts that share it, and $m_g(R)$ quantifies its consequence under the operating distribution. In KeyDoor, augmented-state variables further locate recurrent failures relative to key acquisition, door opening, and entry into the right room.

During execution, goal arrival identifies success; repetition of an augmented state identifies a non-goal fixed point or cycle if success has not occurred. Determinism ensures that the repeated state reproduces the same subsequent behaviour. A trajectory stopped before either event is recorded as unresolved, rather than classified as recurrent failure. Formal definitions, cycle canonicalization, and concentration measures appear in Appendix B.

3.1 RECOVERY FROM SAMPLED EXECUTIONS

For each goal, we sample N starting conditions and execute the frozen policy until success, detected recurrence, or the unresolved cutoff. The recurrent failures encountered form a discovered set $\mathcal{D}_g$; constructing it requires only sampled executions, not an exact functional graph.

Where an exact census is available, we use it only afterwards to score the frozen discovered set. For a goal with $M_g^- > 0$, recovered recurrent failure mass is

$$C_g(\mathcal{D}_g) = \frac{\sum_{R \in \mathcal{D}_g} m_g(R)}{M_g^-}.$$

Mean per-goal coverage weights goals with failures equally; pooled coverage weights them by their recurrent failure mass under the evaluation goal distribution. We also report discovery of the dominant within-goal failure and recall of structures above specified within-goal mass thresholds. These measures assess how much persistent failure the sampled representation accounts for, while distinguishing mass coverage from complete structural recovery. Full metric definitions appear in Appendix B.

4 EXPERIMENTAL DESIGN

We evaluate structural interpretation, sampled recovery, and intervention in three deterministic goal-conditioned settings with complementary experimental roles.

FourRooms (FR12). A 12×12 navigation task provides exact ground truth across five independently trained policies, each evaluated over 125 goals and 124 non-goal starts per goal. We use this setting to evaluate sampled structural recovery and sample-selected local repair, with exact analysis scoring the recovered structures and post-edit outcomes.

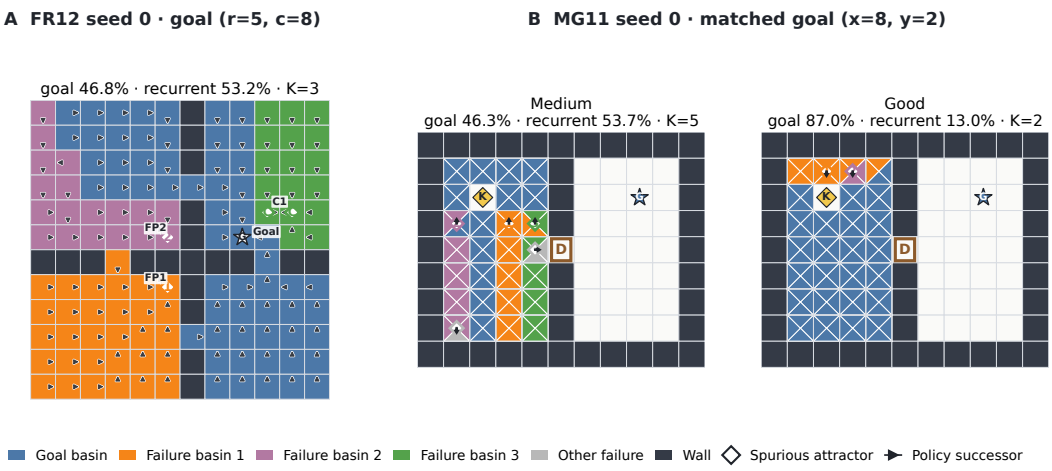

Figure 1: **Recurrent outcomes organize policy failure into interpretable basins. Left:** For one FR12 goal, basin colours identify starts that share an eventual outcome; arrows show policy-induced transitions, and markers locate recurrent failures. **Right:** For a matched MG11 goal, success increases from 46.3% to 87.0% between the Medium and Good checkpoints, while the number of recurrent failure structures falls from five to two. The remaining failure basins show where behaviour still becomes trapped. These panels illustrate individual goals; text reports aggregate census results.

MiniGrid KeyDoor (MG11). An 11×9 task provides Medium and Good checkpoints from the same training run. Their comparison tests how recurrent failure structure and its recoverability change with policy competence; it is not an independent-seed comparison.

MiniGrid KeyDoor (MG16). A larger 16×16 task requires key acquisition, door opening, and navigation to right-room goals. The terminal recurrence-guided $+250$k developmental seed-0 policy supplies the exact structural census and sampled-recovery evaluation over 84 right-room goals and 32,592 starting conditions. Separately, the continuation experiment branches each of three independently trained one-million-step source policies into ordinary, random-trigger, and recurrence-guided arms for an additional 250k training steps. Seed 0 is developmental; seeds 1–2 are confirmatory.

All structural analyses use frozen deterministic greedy policies. Exact census information scores sampled recovery after discovery and is not supplied to the continuation learner. Environment layouts, augmented-state definitions, training configurations, and experiment-specific evaluation populations are detailed in the appendix.

5 Results

5.1 Interpreting and Recovering Recurrent Failure Structure

We first examine what exact recurrent decompositions reveal about policy failure, then test how much of that structure's failure mass can be recovered from sampled executions.

5.1.1 Exact structure reveals shared failure outcomes

Across five FR12 policies, 125 goals per policy, and 124 non-goal starts per goal, the exact census covers 77,500 conditions. Success accounts for 57.12% of operating mass; 39.26% enters non-goal fixed points and 3.62% enters multi-state cycles. These failures comprise 2,376 goal-conditioned recurrent structures. Figure 1 shows how distinct initial states feed the same persistent outcome. The basin representation therefore identifies which starts share a failure and how extensively that failure competes with successful behaviour.

Table 1: Recovery of consequential recurrent failure structure from sampled starts. Coverage is the fraction of exact recurrent failure mass represented by discovered structures, averaged across goals or pooled by failure mass. Dominant discovery and recall of structures carrying at least 5% of within-goal failure mass are reported separately.

Environment / policy	N	Mean per-goal coverage	Pooled coverage	Dominant discovered	Recall $\geq 5\%$
FR12	16	85.3%	87.8%	93.5%	86.4%
FR12	32	95.9%	96.8%	99.0%	97.7%
FR12	64	99.4%	99.5%	99.9%	100.0%
MG11 Medium	16	89.9%	88.9%	87.4%	91.3%
MG11 Medium	32	95.7%	94.6%	95.4%	98.1%
MG11 Medium	64	98.6%	97.5%	100.0%	100.0%
MG11 Good	16	82.9%	83.3%	66.5%	89.5%
MG11 Good	32	91.0%	91.5%	77.3%	98.6%
MG11 Good	64	95.4%	95.9%	87.0%	100.0%
MG16 guided seed 0	16	71.9%	75.0%	94.3%	76.1%
MG16 guided seed 0	32	83.3%	85.0%	98.8%	90.5%
MG16 guided seed 0	64	91.3%	92.1%	99.9%	97.6%

FR12 pools five independently trained seeds and 625 seed–goal policies. MG11 Medium and Good are checkpoints of the same training run over 55 goals, not independent seeds. MG16 uses the terminal recurrence-guided $+250$k developmental seed-0 policy over 84 right-room goals and 32,592 conditions. Sampling starts are drawn uniformly within each goal.

Across the full MG11 evaluation population, success rises from 53.9% to 86.6% between the Medium and Good checkpoints of the same training run, while the number of recurrent failure structures falls from 234 to 108. The decomposition makes the remaining failures explicit even as aggregate competence improves. This is a comparison between two checkpoints, rather than evidence that every individual failure basin contracts during learning.

The exact censuses thus provide both a behavioural interpretation of shared persistent failures and ground truth for evaluating sampled recovery.

5.1.2 Limited sampling recovers most consequential failure mass

Sampling aims to retrieve recurrent failure structures without enumerating the full state space or constructing the complete functional graph. Each sampled execution that repeats an augmented state identifies a fixed point or cycle; matching these outcomes across executions builds a catalogue of discovered failures. Sampling therefore recovers concrete behavioural objects, although it does not enumerate their complete incoming basins. The exact census is consulted only afterwards to measure how much failure mass those objects account for.

In FR12, mean per-goal recovered failure mass rises from 85.3% at $N = 16$ starts per goal to 95.9% at $N = 32$ and 99.4% at $N = 64$ (Table 1). At $N = 32$, the dominant failure is discovered in 99.0% of cases, and recall of structures carrying at least 5% of within-goal failure mass reaches 97.7%. Thus, sampled executions retrieve structures accounting for most persistent failure without requiring a complete catalogue.

MG11 shows why recovery must also be assessed as competence improves. At $N = 32$, mean failure-mass coverage is 95.7% for the Medium checkpoint and 91.0% for Good, while dominant failure discovery falls from 95.4% to 77.3%. Higher policy success therefore does not imply easier discovery of residual failures from uniformly sampled starts. Coverage of consequential failure mass and reliable identification of individual structures remain distinct criteria.

5.1.3 Recovery persists in the larger sequential task

MG16 extends the analysis from 11×9 to 16×16 KeyDoor, with an exact evaluation population of 32,592 start–goal conditions across 84 right-room goals. This is approximately $5.6\times$ the 5,832 conditions evaluated per MG11 checkpoint. These counts describe the evaluation populations, rather than the complete augmented state spaces.

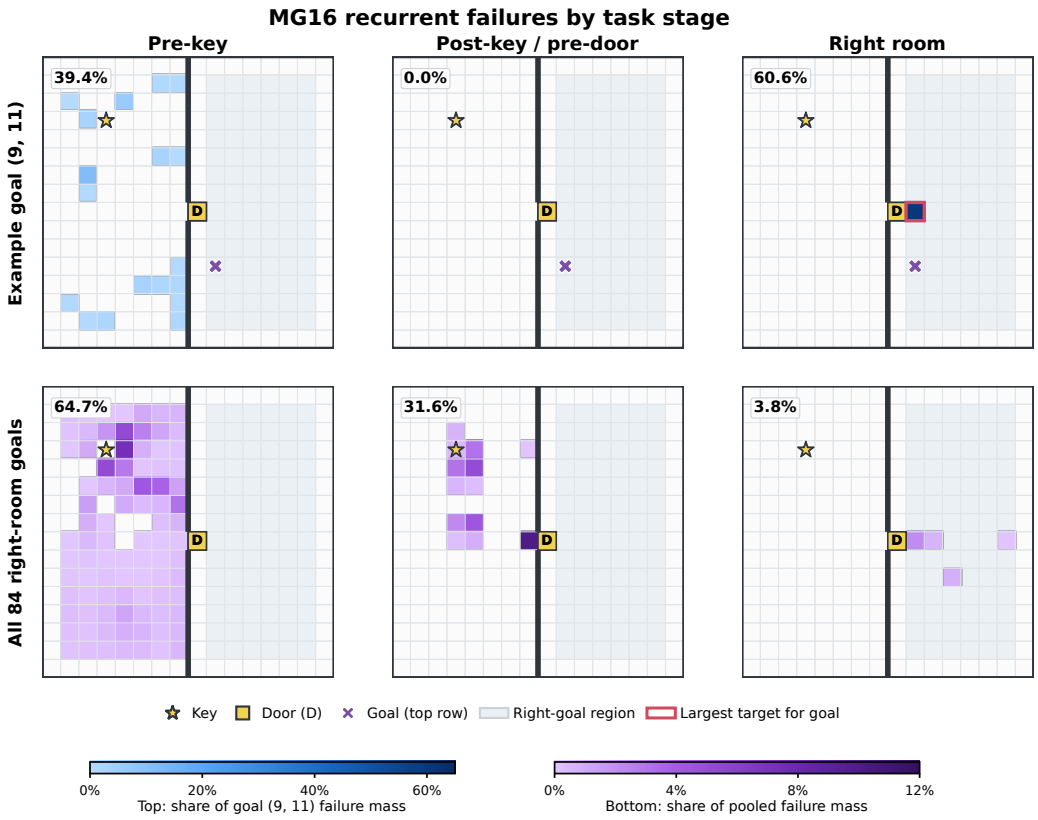

Figure 2: **MG16 recurrent failures organize by sequential task stage. Top:** For example goal $(9, 11)$, recurrent-failure mass is split between pre-key failures (39.4%) and right-room failures (60.6%), with no post-key/pre-door failure mass. Cell intensity denotes the share of that goal's recurrent-failure mass. **Bottom:** Aggregating over all 84 right-room goals gives a different population-level picture: 64.7% of recurrent-failure mass occurs before key acquisition, 31.6% after key acquisition but before door opening, and 3.8% after entering the right room. Cell intensity denotes contribution to pooled recurrent-failure mass. The red outline marks the largest recurrent target for the displayed goal; K, D, and G denote the key, door, and goal.

For the terminal recurrence-guided $+250$k developmental seed-0 policy, exact success is 19.2%; the remaining 80.8% enters 1,755 goal-conditioned recurrent failure structures, all fixed points. The representation identifies where this failure occurs in the task sequence: 64.7% of pooled recurrent-failure mass is pre-key, 31.6% is post-key but pre-door, and 3.8% is in the right room. However, the displayed goal has 60.6% of its failure mass in the right room (Figure 2). Goal-specific decomposition therefore reveals a bottleneck that the pooled diagnosis would obscure.

Despite the large catalogue, failure mass is concentrated within goals. The five largest structures account for 78.7% of recurrent failure mass on average, and the ten largest account for 91.0% (Table 2). This concentration helps explain why limited sampling can recover substantial failure mass without discovering every structure.

At $N = 32$, discovery samples only 8.2% of the 388 evaluated starting conditions per goal. It recovers 83.3% of recurrent failure mass on average across goals and 85.0% when pooled, identifies the dominant failure in 98.8% of goal–trial pairs, and achieves 90.5% recall of structures carrying at least 5% of within-goal failure mass (Table 1). Increasing the budget to $N = 64$, or 16.5% of starting conditions, raises mean and pooled coverage to 91.3% and 92.1%.

Recovery is less complete at a fixed budget than in the smaller settings, but remains effective for identifying dominant failures and representing much of their mass. This demonstrates useful recov-

Table 2: Within-goal concentration of recurrent failure for the developmental guided seed-0 MG16 policy.

Metric	Mean	Median	Range
Failure structures per goal	20.89	16.5	6–46
Largest structure mass	37.59%	32.71%	15.72–82.22%
Top-5 mass	78.74%	81.06%	42.53–99.74%
Top-10 mass	90.99%	94.33%	60.57–100%
Structures for 90% mass	9.76	8	3–29
Mass below 1%	4.06%	3.73%	0–14.18%

All 26,342 recurrent failures in the 32,592-condition census were fixed points.

Table 3: Sample-selected one-step FR12 repair across 500 sampling repetitions per goal and discovery budget N. Success gain is measured in percentage points using exact post-edit evaluation.

N	Diagnosed	Repair available	Success gain	Harmful
8	66.6%	43.5%	+5.9 pp	0%
16	84.7%	71.9%	+10.8 pp	0%
32	95.7%	89.6%	+15.6 pp	0%
64	99.3%	98.2%	+19.6 pp	0%

ery in a larger evaluated system; it does not establish a state-space-independent sampling budget. The fractions above count sampled starts, not the transitions traversed during discovery.

5.2 Recurrent Structure Guides Intervention

5.2.1 Sampled recurrent structures identify actionable repair targets

The basin interpretation suggests that changing an action at a recurrent failure can affect many starts that reach it. We test this prediction through sample-selected one-step policy edits in FR12. Sampled trajectories supply the information used to select edits; the exact census is used only afterwards to evaluate their consequences.

At $N = 32$ sampled starts per goal, recurrent failure is diagnosed in 95.7% of trials and a repair is available in 89.6%, with a reported mean success gain of 15.6 percentage points (Table 3). The gain rises to 19.6 points at $N = 64$. No selected edit is classified as harmful under exact post-edit evaluation. These results support the actionability of the sampled representation: identified recurrent failures provide locations where a local change can improve behaviour across incoming starts.

Figure 3 illustrates the connection between diagnosis and intervention. In FR12, one sample-selected action change breaks a recurrent self-loop and redirects all 45 starts in its incoming basin toward success, increasing the goal basin from $55/124$ to $100/124$. The recurrent state provides a local intervention target, while its basin explains the reach of that edit.

5.2.2 Repair can expose downstream recurrent failures

In MG16, breaking a recurrent failure can redirect its incoming basin into another failure rather than toward the goal. We therefore recompute recurrent structure between repair rounds. Exact right-goal success increases from 1.42% to 24.87% after five rounds of recurrent targeting with the best available exit (Figure 3). Recurrent targeting with random exits reaches 13.40%, whereas random-location repair with the best available exit falls to 0.41%.

These comparisons support recurrent structure as a useful localization signal while showing that exit selection also matters. The behavioural interpretation must be updated after intervention: a repair changes the flow and can expose a downstream bottleneck. Full round-by-round results appear in Appendix Table 18.

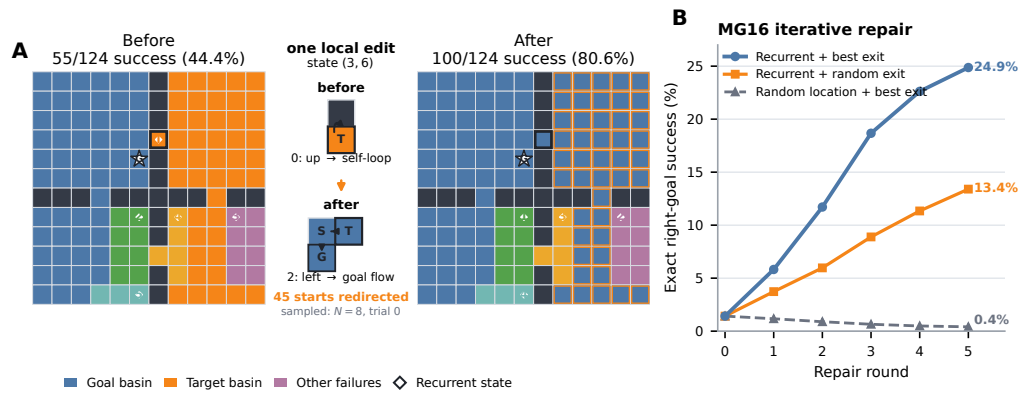

Figure 3: **Recurrent structure provides actionable repair targets. Left:** In FR12, a single sample-selected policy edit breaks a recurrent self-loop and redirects its 45-state incoming basin to successful flow, increasing the goal basin from $55/124$ (44.4%) to $100/124$ (80.6%). **Right:** In MG16, iterative repair at recurrently identified locations progressively increases exact right-goal success. After five rounds, recurrent targeting with the best available exit reaches 24.9% success, recurrent targeting with a random exit reaches 13.4%, while random-location repair reaches 0.4%.

5.2.3 Recurrence-guided continuation improves terminal performance

For each MG16 seed, ordinary, random-trigger, and recurrence-guided continuation branch from the same frozen one-million-step checkpoint and receive 250k additional training environment steps. Seed 0 is developmental; seeds 1–2 use the fixed continuation protocol.

Every 25k training steps, the guided arm diagnoses the current greedy policy using 32 sampled starts for each of 84 right-room goals. Discovered recurrent states and sampled upstream states define trigger regions. Encountering a trigger raises exploration from $\epsilon = 0.12$ to 0.80 for 12 steps; transitions collected in the ensuing 32-step window are retained for dedicated replay. The diagnosis supplies locations for additional learning, without prescribing correct actions. Periodic rediscovery updates these targets as the policy changes.

The random-trigger arm uses the same exploration and replay settings, but selects trigger locations from diagnostic visited states outside the recurrent/upstream target set. Guided and random arms match diagnostic rollout counts; realized trigger encounters and retained targeted transitions differ (Appendix Table 5). Diagnostic transition counts were not logged, so equal total interaction cost is not established. The comparison therefore evaluates the combined localization, exploration, and replay protocol rather than isolating location selection.

Guided continuation achieves the highest terminal overall success in all three seeds and improves right-goal success over ordinary continuation by 11.9, 57.7, and 3.6 percentage points for seeds 0, 1, and 2, respectively (Table 4). It exceeds random triggering on right-goal success in seeds 0 and 1, but not seed 2. These heterogeneous outcomes support directional replication of improvement over ordinary continuation, without establishing population-level superiority.

This experiment connects behavioural interpretation to continued learning: structures identified as persistent failures supply useful targets for additional experience, although the resulting gains depend on the complete intervention protocol.

6 Discussion and Limitations

Recurrent structure connects persistent behaviour to its consequences across initial conditions. A fixed point or cycle identifies a shared failure outcome, its basin identifies the starts that reach it, and its operating mass quantifies its prevalence. Task-stage decompositions further reveal goal-specific bottlenecks that aggregate success can obscure. Local repair and continued learning provide evidence that these behavioural objects can support intervention.

Table 4: Terminal MG16 performance after 250k continuation steps (%). "Random" is the random-trigger control; recurrent failure mass is for right-room goals and is lower when better. Seed 0 is developmental; seeds 1–2 are confirmatory. Full outcomes appear in Appendix Table 20.

Seed	Arm	Overall	Right	Recur.
0	Ordinary	35.1	7.5	92.5
	Random	39.8	8.8	91.3
	Guided	**45.8**	**19.4**	**80.6**
1	Ordinary	35.1	24.0	76.0
	Random	40.7	3.6	96.4
	Guided	**75.2**	**81.7**	**18.3**
2	Ordinary	52.4	44.9	55.1
	Random	52.3	**63.7**	**36.3**
	Guided	**59.6**	48.5	51.5

The interpretation concerns the frozen policy–environment system; it does not explain neural representations or why training produced a particular action choice. We evaluate structural fidelity, sampled recovery, and intervention outcomes, but do not establish improved human understanding through a user study. Our experiments cover deterministic navigation tasks, and the larger MG16 census concerns one developmental policy. They demonstrate a step toward larger systems rather than general scalability.

The functional-graph formulation assumes finite deterministic dynamics and access to sufficient augmented state to detect exact repetition. Stochastic transitions, continuous states, and partial observations require extensions. Long transients can also make recurrence expensive to detect: a small budget of starting conditions need not imply a small transition budget. Executions stopped before success or recurrence remain unresolved.

Failure-mass recovery can leave rare structures undiscovered. Held-out coverage assessment concerns a frozen discovered set under a specified operating distribution, rather than exhaustive verification. Moreover, intervention changes the recurrent landscape, so a diagnosis may need to be recomputed after repair or learning.

The continuation results concern a combined protocol of recurrent localization, exploration, and replay. Differences in realized trigger and replay exposure prevent isolation of localization alone. One developmental and two confirmatory seeds show heterogeneous but directionally consistent gains over ordinary continuation, rather than population-level evidence of superiority. Within these limits, recurrent analysis provides an interpretable connection between failure diagnosis and targeted intervention.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/453fadbd8a1a3af50a9df4df899537b5-Abstract.html.

David Boetius and Stefan Leue. Counterexample-guided repair of reinforcement learning systems using safety critics. *arXiv preprint arXiv:2405.15430*, 2024. doi: 10.48550/arXiv.2405.15430. URL https://arxiv.org/abs/2405.15430.

Kayla Boggess, Sarit Kraus, and Lu Feng. Toward policy explanations for multi-agent reinforcement learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 109–115. International Joint Conferences on Artificial Intelligence Organization, 2022. doi: 10.24963/ijcai.2022/16. URL https://doi.org/10.24963/ijcai.2022/16.

Antoni Climent, Dmitry Gnatyshak, and Sergio Alvarez-Napagao. Applying and verifying an explainability method based on policy graphs in the context of reinforcement learning. In *Arti-*

ficial Intelligence Research and Development, pp. 455–464. IOS Press, 2021. doi: 10.3233/FAIA210166. URL https://doi.org/10.3233/FAIA210166.

Przemyslaw Daca, Thomas A. Henzinger, Jan Kretinsky, and Tatjana Petrov. Faster statistical model checking for unbounded temporal properties. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 9636 of *Lecture Notes in Computer Science*, pp. 112–129. Springer, 2016. doi: 10.1007/978-3-662-49674-9_7. URL https://doi.org/10.1007/978-3-662-49674-9_7.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021. doi: 10.1038/s41586-020-03157-9. URL https://doi.org/10.1038/s41586-020-03157-9.

Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1vuO-bCW.

Meng Fang, Tianyi Zhou, Yali Du, Lei Han, and Zhengyou Zhang. Curriculum-guided hindsight experience replay. In *Advances in Neural Information Processing Systems*, volume 32, pp. 12602–12613, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/83715fd4755b33f9c3958e1a9ee221e1-Abstract.html.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 482–495. PMLR, 2017. URL https://proceedings.mlr.press/v78/florensa17a.html.

Tuan Le, Risal Shefin, Debashis Gupta, Thai Le, and Sarra Alqahtani. AEGIS-RL: Abstract, explainable graphs for integrated safety in RL. *Autonomous Agents and Multi-Agent Systems*, 40 (33), 2026. doi: 10.1007/s10458-026-09749-5.

Jin Li, Yue Wu, Mengsha Huang, Yuhao Sun, Hao He, and Xianyuan Zhan. Unraveling the hidden dynamical structure in recurrent neural policies. *arXiv preprint arXiv:2602.01196*, 2026. URL https://arxiv.org/abs/2602.01196.

Peilang Li, Umer Siddique, and Yongcan Cao. From explainability to interpretability: Interpretable reinforcement learning via model explanations. *Reinforcement Learning Journal*, 6:2256–2270, 2025.

Yang Liu, Enxi Wang, Yufei Gao, Weixin Zhang, Bo Wang, Zhiyuan Zeng, Yikai Zhang, Yining Zheng, and Xipeng Qiu. The past is not past: Memory-enhanced dynamic reward shaping. *arXiv preprint arXiv:2604.11297*, 2026. URL https://arxiv.org/abs/2604.11297.

Chenyang Miao. Learning from failures: Efficient reinforcement learning control with episodic memory. *arXiv preprint arXiv:2603.07110*, 2026. URL https://arxiv.org/abs/2603.07110.

Hadi Ravanbakhsh, Sriram Sankaranarayanan, and Sanjit A. Seshia. Formal policy learning from demonstrations for reachability properties. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6037–6043. IEEE, 2019. doi: 10.1109/ICRA.2019.8793828. URL https://doi.org/10.1109/ICRA.2019.8793828.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1312–1320. PMLR, 2015. URL https://proceedings.mlr.press/v37/schaul15.html.

Nicholay Topin and Manuela Veloso. Generation of policy-level explanations for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 2514–2521, 2019. doi: 10.1609/aaai.v33i01.33012514. URL https://doi.org/10.1609/aaai.v33i01.33012514.

Changmin Yu, Neil Burgess, Maneesh Sahani, and Samuel J. Gershman. Successor-predecessor intrinsic exploration. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/e6f2b968c4ee8ba260cd7077e39590dd-Abstract-Conference.html.

A Additional Related Work

Policy-level explanations and graph representations. Abstracted Policy Graphs represent policy behaviour through transitions among abstract states (Topin & Veloso, 2019); subsequent policy graphs and summaries likewise provide compact sequential descriptions (Climent et al., 2021; Boggess et al., 2022). SILVER takes a different route, deriving interpretable policies from Shapley-based explanations (Li et al., 2026). Our representation groups executions by their eventual recurrent outcome. Its basin identifies starts that share that outcome, while basin mass measures its importance under the specified start distribution. These are complementary ways to describe learned behaviour.

Recurrent and dynamical structure. Finite-state recurrence and functional-graph decomposition are established mathematical tools. Li et al. (2026) study cyclic structure in the hidden-state dynamics of recurrent neural policies during environmental interaction. Our recurrent states instead belong to the augmented task state of a frozen policy–environment system for a fixed goal. We use non-goal recurrence to identify persistent failures and associate each outcome with its incoming basin and operating mass; we make no claim about hidden-state geometry or the novelty of recurrent classes themselves.

Sampled recurrence and verification. Statistical model checking has detected recurrent components from simulated executions without enumerating an entire Markov chain (Daca et al., 2016). AEGIS-RL combines trajectory-derived explanatory graphs with probabilistic verification, risk-guided falsification, and runtime shielding (Le et al., 2026). It is a particularly relevant precedent for connecting behavioural explanations with failure discovery and intervention. Our evaluation asks a different question: after sampled discovery has produced a fixed set of goal-conditioned recurrent failures, how much recurrent failure mass does that set represent? Independent held-out executions estimate coverage of the frozen set. They neither discover omitted structures nor establish formal verification or exhaustive coverage.

Failure-directed learning and repair. Go-Explore returns to archived states and explores outward (Ecoffet et al., 2021). Reverse Curriculum Generation adapts starting states to current competence (Florensa et al., 2017), while Leave No Trace uses recoverability in curriculum construction (Eysenbach et al., 2018). Hindsight Experience Replay and curriculum-guided variants use unsuccessful experience in goal-conditioned learning (Andrychowicz et al., 2017; Fang et al., 2019); structural exploration has also used successor–predecessor relationships (Yu et al., 2023). Our intervention uses a different selection signal: recurrent failures detected in the current frozen greedy policy identify locations for local edits or for additional exploration and replay.

Counterexample-guided policy learning and repair similarly alternate between finding problematic executions, modifying a controller, and reassessing it (Ravanbakhsh et al., 2019; Boetius & Leue, 2024). FEMA retrieves stored failure experiences (Miao, 2026), and MEDS groups recurring error patterns to guide subsequent learning (Liu et al., 2026). In our setting, "recurrent" specifically means that an augmented state repeats within a deterministic closed-loop execution; similar errors across separate episodes do not by themselves meet that definition. After an intervention, we recompute the recurrent structure because altered policy flow can expose downstream failures. The MG16 continuation comparison evaluates the full recurrence-guided exploration-and-replay protocol: unequal realized exposure across arms prevents attributing its gains to location selection alone.

Goal-conditioned setting. Universal Value Function Approximators provide the goal-conditioned policy setting used here (Schaul et al., 2015). We introduce neither a new goal-conditioned objective nor a general targeted-exploration method. The contribution tested in this paper is whether recurrent outcomes, their basins, and their operating mass provide a recoverable explanation of persistent failure and useful targets for intervention.

B Framework and Evaluation Details

B.1 Behavioural Recurrence in Goal-Conditioned Policies

We analyse the behaviour induced by a frozen goal-conditioned policy interacting with its environment. The central object is not the neural network in isolation, but the closed-loop dynamics generated jointly by the policy and the environment.

Let $\mathcal{X}$ denote a finite augmented state space containing all variables required for the environment dynamics to be Markovian. Let $g \in \mathcal{G}$ denote a goal, let $T : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$ be a deterministic transition function, and let $\pi_\theta : \mathcal{X} \times \mathcal{G} \to \mathcal{A}$ be a frozen deterministic policy. For a fixed goal g, their interaction defines the closed-loop map

$$F_{\theta,g}(x) = T\left(x, \pi_\theta(x, g)\right).$$

We suppress θ where unambiguous and write F_g. The directed graph with edges

$$x \longrightarrow F_g(x)$$

is a functional graph: every state has exactly one outgoing edge. For episodic tasks, a successful terminal outcome can equivalently be represented by an absorbing goal state. Under this representation, repeated application of F_g from any state in a finite state space eventually reaches a recurrent component.

The resulting recurrence is a property of the *closed-loop policy–environment dynamics*. It does not imply that the policy network is itself an attractor network, nor does it describe stationary points of the optimization objective used to train the policy.

For a fixed goal g, a recurrent structure R is a periodic orbit of F_g. If

$$F_g(x^*) = x^*,$$

then $R = x^*$ is a fixed point. More generally, a cycle of period $k > 1$ is a sequence of distinct states

$$x_0, x_1, \ldots, x_{k-1}$$

such that

$$F_g(x_i) = x_{(i+1) \bmod k}.$$

Cycles that differ only by their starting position in the sequence are treated as the same recurrent structure.

A recurrent structure is *successful* if it corresponds to the goal outcome and a *recurrent failure* otherwise. The basin of a recurrent structure R is

$$B_g(R) = \left\{x \in \mathcal{X} : F_g^t(x) \in R \text{ for some } t \geq 0\right\}.$$

Thus, states in the same basin may follow different transient trajectories but eventually enter the same recurrent outcome.

Let μ_g denote the operating distribution over starting states for goal g. We define the basin mass of R as

$$m_g(R) = \Pr_{x \sim \mu_g}\left[x \in B_g(R)\right].$$

This quantity distinguishes structural existence from behavioural importance. A recurrent structure may exist in the functional graph yet have very small mass under the operating distribution, whereas another structure may collect trajectories from a large fraction of starting conditions.

Let $\mathcal{R}_g^-$ denote the set of non-goal recurrent structures for goal g. The total recurrent failure mass is

$$M_g^- = \sum_{R \in \mathcal{R}_g^-} m_g(R).$$

When all trajectories are resolved into either success or recurrent failure, success mass and recurrent failure mass sum to one.

In computation, recurrence is identified by reaching the goal or revisiting an augmented state. If a sampled rollout is terminated before either event occurs, we retain it as unresolved rather than classifying it as recurrent failure. This prevents finite rollout limits from creating artificial recurrent outcomes.

The recurrent structures provide a decomposition of persistent failure for each goal. For goals with $M_g^- > 0$, define the within-goal share of a recurrent failure R as

$$q_g(R) = \frac{m_g(R)}{M_g^-}, \qquad R \in \mathcal{R}_g^-.$$

Let

$$q_{g,(1)} \geq q_{g,(2)} \geq \cdots$$

denote these masses in decreasing order. We measure within-goal concentration using

$$C_g(k) = \sum_{j=1}^{\min(k, |\mathcal{R}_g^-|)} q_{g,(j)},$$

the fraction of recurrent failure mass explained by the k largest structures for that goal.

This distinguishes two properties that need not coincide. A policy may contain many distinct recurrent failures across all goals while recurrent failure remains highly concentrated within each individual goal.

For sequential tasks, recurrent structures can additionally be annotated by task stage using variables already present in the augmented state. In KeyDoor, failures can be annotated as pre-key, post-key/pre-door, or right-room failures using task variables and location in the augmented state. Stage labels do not alter the recurrent decomposition; they provide a task-level interpretation of where persistent failure occurs.

B.2 Recovering recurrent structure from sampled executions

Exhaustive evaluation of the specified start–goal population provides exact outcome and basin-mass ground truth for that population, but grows costly as the augmented state and goal spaces increase. We therefore ask how much of the behaviourally important recurrent structure can be recovered using only sampled executions of the frozen policy.

For each goal g, we draw a discovery set of N starting conditions from the operating distribution and execute the deterministic policy until success, recurrence, or the prescribed unresolved cutoff. Let

$$\widehat{\mathcal{R}}_{g,N}^- \subseteq \mathcal{R}_g^-$$

denote the set of recurrent failures observed in these executions. Discovery requires only policy executions and recurrent-outcome identification; it does not require construction of the complete functional graph.

When an exact census is available for evaluation, the fraction of recurrent failure mass recovered for goal g is

$$\rho_g^{\mathrm{fail}}(N) = \frac{\sum_{R \in \widehat{\mathcal{R}}_{g,N}^-} m_g(R)}{M_g^-}.$$

Writing $D_{g,N} = \widehat{\mathcal{R}}_{g,N}^-$, this quantity is $C_g(D_{g,N})$ in the notation of the main paper.

The exact census is used only to evaluate recovery after the discovered set has been frozen; it is not used to select structures.

We report two different aggregations across goals. The macro-average failure coverage is

$$\rho_{\mathrm{macro}}^{\mathrm{fail}}(N) = \frac{1}{|\mathcal{G}^-|} \sum_{g \in \mathcal{G}^-} \rho_g^{\mathrm{fail}}(N),$$

where $\mathcal{G}^-$ contains goals with non-zero recurrent failure mass. This gives each such goal equal weight.

The pooled failure-mass coverage is

$$\rho_{\mathrm{pool}}^{\mathrm{fail}}(N) = \frac{\sum_g \sum_{R \in \widehat{\mathcal{R}}_{g,N}^-} m_g(R)}{\sum_g M_g^-}.$$

Pooled coverage therefore weights goals according to their recurrent failure mass. We report the pooled and macro quantities separately rather than using them interchangeably.

Mass coverage is deliberately distinct from structural completeness. A discovery sample may recover most recurrent failure mass while leaving many low-mass structures undiscovered. We therefore additionally measure whether the dominant within-goal recurrent failure is discovered and the recall of structures exceeding specified within-goal mass thresholds. These quantities distinguish recovery of the behaviourally important head of the distribution from recovery of its residual structural tail.

B.3 Frozen-policy structural analysis

All structural analyses use deterministic greedy policies with fixed parameters. For a given goal and start condition, we execute the policy until one of three outcomes occurs: the goal is reached, an augmented state repeats, or the evaluation horizon is exhausted. A repeated state identifies entry into a recurrent structure. Repetition after one step gives a fixed point; repetition over multiple distinct states gives a cycle. A trajectory that reaches neither the goal nor a repeated state before the cutoff is labelled unresolved rather than recurrent.

Where full enumeration is feasible, we execute the policy from every start–goal condition in the evaluation population and construct an exact census of success and recurrent failure. For every recurrent structure we retain its canonical identity, recurrent type, basin mass, and goal. In sequential environments we additionally assign a failure stage from the augmented state variables.

For each goal we report total success and recurrent failure mass, the number and type of recurrent structures, and the concentration of recurrent failure mass across structures. Concentration statistics are conditional on recurrent failure within that goal. We therefore distinguish global heterogeneity across goals from within-goal concentration.

B.4 Held-out coverage estimation and certification

The sampled-recovery experiments separate *structural discovery* from *coverage estimation*. Discovery determines which recurrent failure structures are known; an independent held-out sample then estimates how much recurrent failure mass is accounted for by that fixed discovered set. No held-out execution is used to add structures to the discovered set.

Target quantity. For a fixed discovery budget, let $\mathcal{D}$ denote the set of goal-conditioned recurrent failure structures observed during discovery. Let an evaluation start–goal pair be drawn from the specified operating distribution and executed under the frozen greedy policy. Conditional on that execution ending in recurrent failure, define

$$p_{\mathcal{D}} = \Pr\left(R \in \mathcal{D} \mid \text{recurrent failure}\right),$$

where R is the recurrent structure reached by the execution. Thus, $p_{\mathcal{D}}$ is the fraction of recurrent failure mass represented by the discovered structure set.

This is a mass-based quantity, not a structural-recall quantity. A discovery set can therefore have high $p_{\mathcal{D}}$ while still missing many individually rare recurrent structures.

Held-out estimator. After $\mathcal{D}$ has been frozen, we generate fresh held-out executions. Among the held-out executions that end in recurrent failure, let n_{fail} denote the number of recurrent failures and let k denote the number whose recurrent structure belongs to $\mathcal{D}$. The held-out estimate is

$$\widehat{p}_{\mathcal{D}} = \frac{k}{n_{\text{fail}}}.$$

When $n_{\text{fail}} = 0$, the estimate and Wilson bound are undefined, so certification cannot occur at that held-out sample size. Successful held-out executions do not enter this conditional denominator. The estimate therefore measures coverage of recurrent failure mass rather than overall policy success or overall behavioural coverage.

Where an exact census is available, we separately compute the true recurrent failure-mass coverage of $\mathcal{D}$. Census information is used only for post-hoc evaluation of the sampling procedure; it is not used to construct $\mathcal{D}$ or to determine whether a held-out execution is covered.

Wilson lower confidence bound. To obtain a conservative assessment of coverage, we compute a one-sided Wilson lower confidence bound for the binomial proportion $p_{\mathcal{D}}$. Let

$$\widehat{p} = \frac{k}{n_{\text{fail}}},$$

and let z denote the normal critical value used for the prespecified confidence level. The lower bound is

$$L_{\text{W}} = \frac{\widehat{p} + \frac{z^2}{2n_{\text{fail}}} - z\sqrt{\frac{\widehat{p}(1-\widehat{p})}{n_{\text{fail}}} + \frac{z^2}{4n_{\text{fail}}^2}}}{1 + \frac{z^2}{n_{\text{fail}}}}.$$

We use a target recurrent failure-mass coverage of

$$\tau = 0.95.$$

The discovered representation passes the coverage rule at a held-out look when

$$L_{\text{W}} \geq \tau.$$

We refer to this event as *certification* for brevity. This terminology denotes passage of the prespecified confidence-based stopping rule; it should not be interpreted as formal verification, exhaustive structural certification, or an exact finite-sample guarantee that all recurrent structures have been discovered.

Repeated held-out looks. Coverage is assessed at seven prespecified held-out sample sizes,

$$m \in \{16, 32, 64, 128, 256, 512, 1000\}$$

draws per goal. Because multiple opportunities to pass the rule are examined, we adjust the confidence level across the seven declared looks. We use a family-wise error level of 0.05 and the one-sided critical value corresponding to

$$\alpha_{\text{look}} = \frac{0.05}{7},$$

giving

$$z = 2.449998.$$

Once a repetition satisfies the coverage rule, it is treated as permanently certified for the purpose of reporting the first-certification statistic.

The variable m denotes the number of held-out starting conditions sampled per goal at a given look. The Wilson calculation itself uses n_{fail}, the number of those executions that actually produce recurrent failure. Consequently, the effective binomial sample size can be smaller than the total number of held-out executions when the policy succeeds from some held-out starts.

Fresh sampling and replacement. Held-out samples are independent of the discovery sample. In MG16, the largest held-out look exceeds the finite number of distinct ordinary start conditions available for an individual goal. Held-out starting conditions are therefore drawn independently with replacement from the evaluation population. Discovery sampling remains without replacement. This preserves the distinction between the finite discovery set and a fresh IID stream used to estimate its operating coverage.

Reported quantities. For each discovery budget, the certification analysis reports:

- held-out recurrent failure coverage $\widehat{p}_{\mathcal{D}}$;
- the corresponding Wilson lower confidence bound;
- whether the coverage rule is satisfied at each declared look;
- the fraction of repeated discovery-and-evaluation trials that certify;
- the first held-out sample size at which certification occurs.

The reported median first-certifying sample size is conditional on certification. If no repetition certifies for a particular discovery budget, this quantity is undefined rather than being assigned the largest held-out sample size.

Where exact census coverage is available, we additionally record whether the confidence-based rule certifies when true recurrent failure-mass coverage is below the target. This provides a direct audit of false certification in the experiments for which exact ground truth is available.

Interpretation. The discovery budget and the held-out budget address different sources of uncertainty. Increasing the discovery budget can reveal recurrent structures that were previously unknown and thereby increase true coverage. Increasing only the held-out budget cannot discover new structures; it can only reduce uncertainty about the mass represented by the already frozen discovered set.

This distinction is particularly important when recurrent failure has a long low-mass tail. A large held-out sample can provide a precise estimate that the discovered set accounts for less than 95% of recurrent failure mass. In that case, failure to certify is the intended outcome: additional evaluation cannot compensate for insufficient structural discovery.

Conversely, passing the rule does not imply exhaustive structural recovery. Rare recurrent structures may remain undiscovered even when the discovered set accounts for at least the target fraction

of recurrent failure under the specified operating distribution. The coverage assessment therefore supports a distribution-conditioned statement about behavioural mass, not a claim that the complete functional graph has been reconstructed.

C Experimental Details

This section provides the environment, policy-training, structural-evaluation, and intervention details omitted from the main text. Statistical definitions, sampled-recovery metrics, and held-out coverage certification are described separately in Appendix B. Unless otherwise stated, all structural analyses use deterministic greedy policies with exploration disabled.

C.1 FR12

Environment. FR12 is a deterministic 12×12 FourRooms environment. Internal walls divide the grid into four connected rooms. There are 125 valid navigable cells, all of which may serve as goals. For each goal, the remaining 124 valid cells form the non-goal starting population. The episode and structural-evaluation horizon is $H = 32$.

Policy and training. Policies are scalar Value-UVFAs with hidden dimension 128. State and goal coordinates are provided jointly to the value network. Training uses one-step temporal-difference learning with discount factor $\gamma = 0.99$, Adam optimization, minibatches of 128 transitions, and ϵ-greedy exploration.

Exploration begins at $\epsilon = 1.0$, decays during training, and is bounded below by 0.05. Policies are trained for 1,000 episodes with uniformly sampled goals. Structural evaluation always uses the deterministic greedy policy.

We analyse five independently trained corrected policies, corresponding to seeds 0–4. The corrected implementation fixes the transition and goal-handling issues identified during the earlier exploratory implementation while retaining the same experimental role and evaluation protocol.

Exact structural census. For each frozen policy and each of the 125 goals, we evaluate all 124 non-goal starting states. The full census therefore contains

$$5 \times 125 \times 124 = 77{,}500$$

start–goal executions.

Each execution is followed until success, entry into a recurrent structure, or the evaluation cutoff. The resulting census records goal success, recurrent failure, recurrent-structure identity, fixed-point or cycle type, and basin membership.

Across the five policies, the exact census is used as evaluation ground truth for the sampling experiments. It is not used to select structures during sampled discovery.

Sampled structural recovery. To emulate a setting in which exhaustive enumeration is unavailable, we sample a subset of initial states independently for each goal. Discovery starts are sampled uniformly without replacement from the 124 non-goal states.

The principal discovery budgets are

$$N \in \{16, 32, 64\}$$

starts per goal.

For each sampling repetition, recurrent structures are identified only from the sampled trajectories. After the discovered set has been frozen, the exact census is used to measure the true recurrent failure mass represented by that set.

Held-out certification. A separate held-out procedure tests whether the operating failure mass represented by the discovered structure set can be estimated without access to the exact census. Discovery and held-out executions use independent samples, and held-out samples do not add new structures to the discovery set.

The confidence rule, sequential held-out looks, Wilson lower bound, target coverage, and treatment of recurrent-failure conditioning are specified in Appendix B.4.

Local structural repair. FR12 also provides a setting in which recurrent diagnosis can be tested through direct policy intervention.

Repair operates on a frozen policy. Identified recurrent failures define candidate local intervention sites. A policy edit changes only the selected local action assignment; policy parameters are otherwise left unchanged. The complete exact census is then recomputed to determine the global behavioural consequences of that intervention.

We distinguish several possible outcomes. An edit can remove the targeted recurrent structure and redirect its incoming basin to the goal; it can remove the targeted structure while redirecting trajectories into another recurrent failure; or it can alter trajectories outside the originally targeted basin.

We therefore evaluate repair using post-edit success, remaining recurrent failure mass, elimination of the targeted structure, basin redistribution, number of local edits, and collateral changes to previously successful or otherwise unaffected start–goal conditions.

Exact sequential repair is used to characterize the repairability of the structural failures when complete structural information is available. The sampled repair experiments instead select intervention targets from structures recovered by limited sampling and therefore test whether approximate structural diagnosis remains actionable.

C.2 MG11 KeyDoor

Environment and policy. MG11 is a fixed-layout 11×9 two-room KeyDoor task, rather than a randomized DoorKey layout. The key is at $(2, 2)$ and a locked door lies in the central divider. The action set is left, right, forward, pickup, and toggle; ineffective pickup and toggle actions remain available and can induce self-loops. Structural analysis uses the augmented state (x, y, d, k, o, l), comprising position, orientation, key possession, door-open status, and door-locked status. The horizon is 256 steps.

The policy is induced by a scalar goal-conditioned Value-UVFA with two 128-unit ReLU hidden layers and a 14-dimensional fully observable input. Greedy control scores deterministic successor states, with explicit handling of actions that enter the goal. Training uses one-step TD updates, $\gamma = 0.99$, Adam with learning rate 10^{-3}, minibatches of 128, and eight replay updates per episode. Ordinary replay is supplemented with protected experience around key acquisition and door opening. Goal-gated potential-based shaping uses $\beta_{\text{key}} = \beta_{\text{door}} = 0.25$. Exploration starts at $\epsilon = 1$, decays by a factor of 0.999 per episode, and has a floor of 0.05.

Checkpoint comparison and evaluation. Medium and Good come from the same seed-0 training run, not independent seeds. Medium is the episode-700 checkpoint selected by the prespecified competence rule; Good is the terminal episode-5000 checkpoint of the fixed run, rather than the best intermediate evaluation. Both are assessed with exploration disabled on the same 55-goal operating population, comprising 5,832 start–goal–orientation conditions per checkpoint. Exact evaluation follows each trajectory to the goal, a repeated complete augmented state, or the horizon, and identifies recurrent structures and their incoming mass. Sampled recovery draws starts for each goal and compares discovered structures with this exact census; the principal matched budgets are $N \in \{16, 32, 64\}$.

C.3 MG16 KeyDoor

Environment. MG16 is a deterministic 16×16 MiniGrid KeyDoor environment. Boundary walls surround the grid and a vertical interior divider separates the left and right rooms. The divider contains a locked yellow door at coordinate $(8, 8)$, and the yellow key is located at $(3, 3)$.

Episodes begin in the left room. The agent has one of four orientations. Goals may occur in either room. Reaching a right-room goal typically requires the sequential completion of several behavioural stages: reaching and acquiring the key, navigating to the door, opening the door, entering the right room, and navigating to the requested goal.

The episode horizon is

$$H = 256.$$

Augmented Markov state. For structural analysis, the state must include task variables that affect future transitions. We therefore represent the behavioural state as

$$x = (p_x, p_y, d, k, o, l),$$

where (p_x, p_y) is the agent position, d is orientation, k denotes possession of the key, o denotes whether the door is open, and l denotes whether the door is locked.

Recurrence is detected in this augmented state rather than in agent position alone. Two visits to the same grid cell with different orientations or different key/door status are therefore distinct states.

Action space. The policy selects among five actions:

$$\{\text{left}, \text{right}, \text{forward}, \text{pickup}, \text{toggle}\}.$$

Pickup and toggle remain available even when they are not applicable. Invalid or ineffective actions may therefore leave the augmented state unchanged. Such choices generate genuine self-loops under the policy and can appear as fixed-point recurrent failures.

Value-UVFA representation. The policy uses a scalar Value-UVFA. Its 14-dimensional input contains normalized agent coordinates, a four-dimensional one-hot encoding of orientation, key possession, door-open and door-locked indicators, normalized goal coordinates, relative goal displacement, and an indicator for whether the requested goal lies in the right room.

The network uses hidden dimension 128 and outputs a scalar state–goal value.

Base training configuration. MG16 policies are trained using online one-step TD learning with discount factor

$$\gamma = 0.99.$$

Optimization uses Adam with learning rate 10^{-3} and minibatches of 128 transitions. The ordinary replay buffer has capacity 500,000 transitions. After each completed episode, eight replay-update iterations are performed.

Additional milestone replay buffers retain transitions associated with progress through the KeyDoor task, including key acquisition and door opening.

Potential-based reward shaping is used during training with coefficients

$$\beta_{\text{key}} = 0.25, \qquad \beta_{\text{door}} = 0.25.$$

These shaping terms are part of the original training environment. The later recurrence-guided continuation experiments do not introduce an additional recurrence-specific reward.

Exploration schedule. Initial training uses ϵ-greedy exploration beginning at

$$\epsilon_0 = 1.0.$$

After each episode,

$$\epsilon \leftarrow \max(0.05, 0.9995\,\epsilon).$$

Thus exploration decays throughout training without being reset during the original one-million-step run.

Goal curriculum. Training uses a staged goal distribution. The schedule used in the executed developmental and confirmatory runs is

$$0\text{–}150{,}000: \quad \text{ordinary training distribution},$$

$$150{,}000\text{–}500{,}000: \quad 90\% \text{ right-room goals}, \quad 10\% \text{ left-room goals},$$

and

$$500{,}000\text{–}1{,}000{,}000: \quad 75\% \text{ right-room goals}, \quad 25\% \text{ left-room goals}.$$

The executed training code uses the 500,000-transition boundary between the 90:10 specialization phase and the 75:25 consolidation phase. A later protocol manifest incorrectly recorded this boundary as 300,000 transitions; the training implementation and executed-run records use 500,000.

Seed roles. We analyse three independently trained MG16 policies.

Seed 0 is developmental: it was available while the structural analysis and continuation intervention were being developed. Seeds 1 and 2 are confirmatory replications trained and evaluated using the frozen protocol without seed-specific intervention tuning.

The three seeds are reported individually. They are used to assess whether the observed direction of the continuation effect replicates across independently trained policies, rather than to support population-level statistical inference from three observations.

Greedy evaluation. All reported policy-performance and structural evaluations disable exploration and use deterministic greedy action selection.

For learning-curve evaluation, each seed uses a fixed panel of 1,000 start–goal–orientation conditions. The same panel is used for all continuation arms originating from that seed. We report overall success, left-goal success, right-goal success, key acquisition, door opening, right-room entry, and recurrent failure on right-goal conditions.

Terminal structural census. The terminal structural analysis is performed separately from the 1,000-condition learning-curve panel.

For the terminal recurrence-guided $+250\text{k}$ developmental seed-0 policy, we enumerate the predefined right-goal structural population over 84 right-room goals, yielding

$$32{,}592$$

goal-conditioned start conditions per seed.

For every condition we execute the greedy policy until success, state recurrence, or the horizon. Recurrent structures are canonicalized by their complete augmented-state representation.

For interpretability, each recurrent failure is additionally assigned to a task stage:

$$\text{pre-key}, \qquad \text{post-key/pre-door}, \qquad \text{right-room}.$$

These stage labels are descriptive annotations applied after recurrence detection. They do not modify the functional-graph definition of a recurrent structure.

Table 5: Exposure matching between recurrence-guided and random-trigger continuation. Each entry is the guided-to-random ratio; 1 denotes exact matching.

Seed	Steps	Episodes	Updates	Diagnostics	Selected	Encounters	Targeted trans.	Replay samples
0	1.000	0.915	0.915	1.000	1.028	0.808	0.765	0.913
1	1.000	1.054	1.054	1.000	0.865	0.627	0.601	1.054
2	1.000	1.047	1.046	1.000	1.040	0.681	0.673	1.046

Training-step budgets and diagnostic-rollout counts are effectively identical; diagnostic environment-transition counts were not logged. Selected-state counts and replay intensity are close, but guided runs realize fewer trigger encounters and retain fewer targeted transitions. We therefore classify exposure as *moderately different*: the comparison supports the complete recurrence-guided protocol, not location selection in isolation. Seed 0 is developmental; seeds 1–2 are confirmatory.

The terminal census is used to measure recurrent failure mass, number of distinct recurrent structures, fixed-point versus cycle composition, stage distribution, within-goal mass concentration, and exact ground truth for the MG16 sampled-recovery analysis.

C.4 MG16 CONTINUATION PROTOCOLS AND EXPOSURE ACCOUNTING

We next test a less direct use of recurrent structure: using recurrent failures to determine where additional learning effort is allocated while leaving the policy free to discover improved actions through ordinary reinforcement learning.

For each MG16 seed, the same frozen one-million-step checkpoint is continued for an additional 250,000 training environment steps. Ordinary continuation provides the baseline. The recurrence-guided arm periodically analyses the current greedy policy, identifies recurrent failures, and uses these structures to localize additional exploration and guided replay. Policy parameters remain trainable throughout, so the recurrent landscape can change as learning proceeds and is periodically rediscovered.

The guidance supplies *where* additional learning effort should be concentrated, but not *what action* should be taken. It does not provide oracle actions or trajectories, mask actions, or modify the task reward. Because the intervention combines recurrent localization, local exploration, and guided replay, we evaluate the complete structure-guided protocol and do not attribute any improvement to recurrence localization alone.

The primary continuation comparison is terminal greedy policy performance after the matched additional interaction budget. Intermediate checkpoints are retained to characterize learning trajectories, but the curves are not assumed to be monotonic. Results are reported separately for all three seeds rather than pooling them into a population-level significance claim.

Common branch point. For each seed, all continuation arms begin from the same approximately one-million-environment-step policy checkpoint.

At the branch point, policy parameters and the associated training state required for continuation are copied so that the arms begin from the same trained policy. Each arm then receives approximately 250,000 additional training environment steps. Diagnostic rollouts are accounted for separately.

The continuation horizon, reward function, shaping function, TD update rule, network architecture, optimizer configuration, replay capacities, and evaluation conditions are held fixed across arms.

During continuation, right-room goals are sampled with probability 0.75. The ordinary exploration level is

$$\epsilon_{\text{base}} = 0.12.$$

Terminal performance after the additional 250,000 interactions is the prespecified primary endpoint.

Evaluation schedule. All arms are evaluated at the branch checkpoint and every 25,000 additional environment steps:

$$0,\ 25k,\ 50k,\ \ldots,\ 250k.$$

The same fixed 1,000-condition greedy evaluation panel is used across arms within a seed.

Intermediate checkpoints are retained to show the trajectory of continued learning. They are not used to choose a retrospectively optimal checkpoint, and continuation curves are not assumed to improve monotonically.

Ordinary continuation. The ordinary arm continues the existing TD learner using the continuation goal distribution and base exploration rate

$$\epsilon = 0.12.$$

It receives no recurrent-structure diagnosis, no structure-triggered exploration bursts, and no dedicated recurrence-guided replay.

This arm measures what is achieved by simply continuing ordinary training for the same additional interaction budget.

Recurrence-guided continuation. The recurrence-guided arm periodically analyses the current greedy policy to determine where persistent recurrent failures remain.

Structural rediscovery occurs every 25,000 continuation environment steps. At each rediscovery point, the frozen current greedy policy is evaluated from

$$N = 32$$

sampled starts for each right-room goal.

Observed recurrent structures are ranked within each goal according to their empirical recurrence frequency. Structures are retained until they account for 80% of the sampled recurrent outcomes for that goal, with a maximum of three selected structures per goal.

For each selected recurrent structure, a trigger region is constructed from the recurrent state or states together with up to four upstream states observed on sampled trajectories leading into the structure.

During subsequent online training, entry into a selected trigger region initiates a temporary local exploration burst. Exploration is increased from

$$\epsilon_{\text{base}} = 0.12$$

to

$$\epsilon_{\text{local}} = 0.80$$

for 12 environment steps.

The intervention does not specify the action that should be taken. Actions during the burst remain ϵ-greedy. The method therefore supplies a location for additional exploration but no oracle action, trajectory, action mask, or task-specific repair instruction.

Guided replay. Transitions generated in and immediately downstream of a triggered region are retained for up to 32 subsequent environment steps in a dedicated guided-replay buffer.

When sufficient guided transitions are available, 25% of each replay minibatch is drawn from this buffer. The remainder is supplied by the ordinary replay mechanism.

The recurrence-guided intervention therefore consists of three coupled components:

$$\text{recurrent localization} \rightarrow \text{local exploration} \rightarrow \text{guided replay}.$$

Our causal claim concerns this complete protocol. The experiment does not by itself identify how much of the observed improvement is attributable separately to localization, temporary exploration, or replay.

Online rediscovery. Targets are recomputed throughout continuation rather than being fixed from the initial checkpoint.

This is necessary because successful learning can remove a previously observed fixed point, alter its basin, or expose a new downstream recurrent failure. The structural analysis therefore tracks the recurrent behaviour of the *current* greedy policy at each rediscovery point.

Exact terminal census information is never supplied to the learner. Training targets are constructed only from the sampled diagnostic trajectories available at that continuation stage.

Random-trigger control. The random-trigger arm tests whether similar gains arise when the same general intervention machinery is used without recurrence-based localization.

It uses the same local exploration value,

$$\epsilon_{\text{local}} = 0.80,$$

the same 12-step exploration burst, 32-step downstream transition retention, 25% dedicated replay share, rediscovery cadence, diagnostic-rollout budget, continuation goal distribution, update rule, and evaluation schedule as the recurrence-guided arm.

Instead of selecting recurrent structures and their upstream states, trigger locations are sampled from states visited during the corresponding diagnostic trajectories. States belonging to the recurrence-guided recurrent/upstream neighbourhood are excluded from the random trigger pool.

The random arm therefore preserves localized additional exploration and dedicated replay while replacing recurrent localization with a random visited-state localization rule.

Exposure matching and audit. Equal nominal environment budgets do not guarantee equal realized exposure after the arms begin to follow different policies. Episode lengths can differ; selected trigger regions can be encountered at different frequencies; and this can change the number of targeted transitions entering the dedicated replay buffer.

We therefore audit realized exposure rather than assuming it is identical.

For each seed and continuation arm we record total training environment interactions, completed episodes, mean episode length, gradient updates, left- and right-goal training episodes, diagnostic policy executions, number of selected trigger locations, realized trigger encounters, targeted transitions, dedicated replay samples, and successful right-goal training episodes.

Diagnostic interactions are also accounted for separately from ordinary training interactions.

The guided and random arms are designed to match the main intervention settings and diagnostic budget, but realized trigger and replay exposure need not be numerically identical. Consequently, the random-trigger comparison is interpreted as a control for the complete random-localization protocol, not as a perfectly isolated causal estimate of recurrence-based location selection.

Frozen-policy identity. Before comparing continuation arms, we verify that the branches associated with a given seed originate from the same frozen one-million-step policy. Frozen-checkpoint greedy evaluation is repeated using the common evaluation conditions, and the resulting baseline predictions and metrics are checked for identity across arm archives.

This validation prevents differences introduced during checkpoint loading, evaluation-condition construction, or archival transfer from being mistaken for continuation effects.

Primary comparison. The primary intervention comparison is terminal greedy performance after the matched additional 250,000-step continuation budget.

Table 6: Structural complexity and finite-sample recovery across the three empirical settings. FR12 provides multi-seed controlled evaluation, MG11 compares checkpoints of differing competence, and MG16 evaluates the larger sequential task. Recovery quantities use $N = 32$ IID starts per goal.

Environment / policy	Role	Goals	Conditions	Success mass	Recurrent failure	Failure structures	N=32 mean/goal	N=32 pooled	N=32 dominant	N=32 recall $\geq$5%
FR12 seed 0	independent seed	125	15,500	48.3%	51.7%	606	95.7%	96.2%	99.5%	97.3%
FR12 seed 1	independent seed	125	15,500	58.6%	41.4%	440	95.7%	96.8%	98.8%	97.8%
FR12 seed 2	independent seed	125	15,500	60.7%	39.3%	410	95.7%	97.3%	98.2%	98.1%
FR12 seed 3	independent seed	125	15,500	64.2%	35.8%	408	95.8%	97.1%	98.6%	97.6%
FR12 seed 4	independent seed	125	15,500	53.8%	46.2%	512	96.7%	96.9%	99.7%	97.7%
MG11 Medium	checkpoint comparison	55	5,832	53.9%	46.1%	234	95.7%	94.6%	95.4%	98.1%
MG11 Good	checkpoint comparison	55	5,832	86.6%	13.4%	108	91.0%	91.5%	77.3%	98.6%
MG16 guided seed 0	developmental	84	32,592	19.2%	80.8%	1,755	83.3%	85.0%	98.8%	90.5%

FR12 rows are independently trained policies. MG11 Medium and Good are checkpoint-quality comparisons, not independent seeds. The MG16 row uses the terminal recurrence-guided +250k developmental seed-0 policy over 84 right goals; the sampled-recovery analysis reported here uses this developmental seed-0 checkpoint. Failure structures are goal-conditioned recurrent outcomes. Mean/goal and pooled failure-mass coverage are distinct.

Table 7: Exact MG11 census under medium and good checkpoints.

Metric	Medium	Good	Change
Success mass	53.9%	86.6%	+32.7 pp
Recurrent failure mass	46.1%	13.4%	-32.7 pp
Failure structures	234	108	-126
Largest competing basin	22.8%	10.3%	-12.5 pp
Conditional failure entropy	0.961	0.456	-0.505
Effective failure outcomes	2.98	1.86	-1.12
Fixed-point / cycle mass	46.1% / 0%	13.4% / 0%	-32.7 pp / 0

Checkpoint robustness, not seed-level uncertainty. Complete-census values weight 5,832 conditions per checkpoint.

Ordinary continuation asks whether further training alone improves the policy. Recurrence-guided continuation asks whether allocating additional exploration and replay around recurrent failures improves learning under the same nominal interaction budget. The random-trigger arm asks whether comparable localized intervention without recurrence-based targeting produces the same effect.

Results are presented by seed and arm. Seed 0 is identified as developmental, while seeds 1 and 2 are identified as confirmatory throughout the main text, tables, and appendix.

D Additional Structural Recovery Results

D.1 Exact censuses and policy-quality comparisons

Table 6 compares exact census outcomes and sampled recovery across policies. Tables 7 and 8 give the MG11 checkpoint comparison in detail.

D.2 Full sampled-recovery results

Tables 9 and 10 report recovery across discovery budgets in FR12 and MG16.

D.3 MG11 sampling ablations

The first ablation compares equal-allocation sampling across task milestones with IID starts at the same total discovery budget N. The hybrid reduces recovery at every tested budget.

In the follow-up, Stage 1 uses $N = 32$ IID starts per right-room goal. Goals with an observed pre-key failure receive 16 additional starts, selected either IID or near key access. Table 12 reports outcomes per activated goal where applicable.

Table 8: MG11 sampled recovery under medium and good checkpoints. Values average 20 paired repetitions.

Policy	N	Failure mass recovered	Dominant discovered	Recall $\geq$5%
Medium	16	89.9%	87.4%	91.3%
Good	16	82.9%	66.5%	89.5%
Medium	32	95.7%	95.4%	98.1%
Good	32	91.0%	77.3%	98.6%
Medium	64	98.6%	100.0%	100.0%
Good	64	95.4%	87.0%	100.0%

Better training shrinks residual failure but makes some goal-specific dominant structures harder to observe.

Table 9: FR12 sampled recovery across discovery budgets. Values are means over 625 seed–goal policies, with 100 repeated discovery/held-out splits per policy and budget.

N	Failure mass	Dominant found	Recall $\geq$5%	Recall $\geq$10%	Held-out coverage	True unseen coverage	Held-out MAE
4	45.8%	57.1%	43.3%	54.4%	75.7%	75.8%	4.65 pp
8	66.8%	78.6%	65.7%	77.4%	86.3%	86.3%	3.53 pp
16	85.3%	93.5%	86.4%	93.9%	93.9%	93.9%	2.09 pp
32	95.9%	99.0%	97.7%	99.6%	98.1%	98.1%	0.85 pp
64	99.4%	99.9%	100.0%	100.0%	99.6%	99.6%	0.23 pp

D.4 Rare structures and goal-level variation

D.4.1 Rare structures form a long tail

The gap between dominant-structure discovery and total failure-mass recovery arises from a long tail of increasingly rare recurrent failures.

In FR12, structures carrying less than 5% of within-goal recurrent failure mass constitute a substantial fraction of the recurrent catalogue while accounting for only a small fraction of recurrent failure mass. The number of distinct structures therefore grows faster than their aggregate behavioral importance.

The effect is especially clear in MG16. Within each goal, the largest recurrent structure accounts on average for 37.59% of recurrent failure mass. The top five structures account for 78.74%, and the top ten for 90.99%. On average, only 2.58 structures are required to explain 50% of within-goal recurrent failure mass, 5.36 to explain 75%, and 9.76 to explain 90%. Reaching 95% requires 12.87 structures on average.

Conversely, recurrent structures individually carrying less than 1% of within-goal failure mass account collectively for only 4.06% of failure mass. Thus the policy can contain many distinct recurrent outcomes while the majority of its persistent failure behavior remains concentrated in a comparatively small head.

The pooled MG16 stage distribution summarizes where recurrent failure mass concentrates across the full set of right-room goals, but individual goals can exhibit markedly different structural profiles. Figure 4 provides two additional exact decompositions from the frozen terminal guided policy.

For goal $(14, 13)$, recurrent failure is overwhelmingly concentrated before key acquisition: 92.5% of recurrent-failure mass is pre-key, compared with 4.1% in the post-key/pre-door stage and 3.4% in the right room. In contrast, goal $(14, 6)$ is dominated by the intermediate task stage: 87.4% of recurrent-failure mass occurs after key acquisition but before door opening, while 12.6% is pre-key and no recurrent-failure mass is observed in the right room.

These examples reinforce two points from the main text. First, the aggregate MG16 failure profile should not be interpreted as a common template shared by all goals. Second, recurrent decomposition preserves goal-specific information about where sequential behavior becomes trapped. The same frozen policy can therefore exhibit qualitatively different failure bottlenecks depending on the requested goal, even when those goals lie in the same broad task region.

Table 10: IID structural recovery for the developmental guided seed-0 MG16 policy.

N	Pooled coverage	Mean goal coverage	Dominant found	Recall $\geq$1%	Recall $\geq$5%	Structures / goal
16	74.97%	71.92%	94.35%	47.18%	76.13%	5.96
32	84.99%	83.25%	98.75%	63.79%	90.54%	8.63
64	92.14%	91.29%	99.93%	79.94%	97.57%	11.83

No repetition achieved 95% true pooled coverage, and the Wilson procedure issued no certification.

Table 11: Prespecified MG11 milestone-stratified discovery versus matched-budget IID sampling.

N	Method	Failure mass	Dominant found	Recall $<1\%$	Recall 1–5%
16	IID	84.2%	54.8%	14.6%	21.7%
16	Hybrid	66.8%	34.1%	3.9%	5.0%
32	IID	93.0%	71.6%	31.1%	40.6%
32	Hybrid	75.2%	43.9%	5.0%	11.7%
64	IID	95.9%	81.8%	57.1%	66.7%
64	Hybrid	87.5%	59.5%	12.5%	22.8%

The equal-allocation hybrid is counterproductive at every budget; this is a limitation result.

E Additional Policy-Repair Results

The FR12 tables report sample-selected repair, targeted-success sampling, and exact repairability audits. The MG16 results then show how success changes across iterative repair rounds.

E.1 MG16 iterative repair

Table 18 reports exact right-goal success after each repair round for recurrent and random intervention locations.

F Additional Continued-Learning Results

This section reports the common starting checkpoints, terminal outcomes, and realized training exposure for the three continuation arms.

F.1 Frozen checkpoints and full terminal outcomes

Table 19 checks that each seed's arms start from the same checkpoint; Table 20 gives their terminal results.

F.2 Exposure audit and random-trigger control

Table 21 compares the realized training and diagnostic exposure across arms, including trigger encounters and targeted replay.

Table 12: MG11 geography-guided Stage-2 follow-up with 16 additional starts per activated goal. Means and 95% Student-t intervals use 20 repetitions.

Metric	Additional IID	Key access	Difference
New structures / active goal	0.145 [0.105, 0.186]	0.188 [0.130, 0.247]	+0.043 [-0.015, 0.101]
New operating failure mass	0.233% [0.107, 0.359]	0.325% [0.168, 0.483]	+0.093 pp [0.010, 0.175]
Total failure mass recovered	92.20% [90.68, 93.72]	92.80% [91.23, 94.36]	+0.60 pp [-0.63, 1.83]
Dominant failure discovered	95.57% [93.21, 97.94]	96.07% [93.75, 98.40]	+0.50 pp [-0.55, 1.55]
Recall $\geq$1%	85.66%	89.35%	+3.69 pp [0.69, 6.70]
Recall $\geq$5%	100%	100%	0

The gain is narrow: low-mass recall improves, but dominant discovery and total recovery do not materially change.

Table 13: FR12 failure structures form a long tail. Counts and masses pool 2,376 non-goal recurrent structures across 625 seed–goal policies.

Mass threshold	Structures below	Count share	Failure mass share
2.5%	250	10.5%	1.4%
5%	602	25.3%	6.7%
10%	1,384	58.2%	30.6%
20%	2,076	87.4%	66.0%

"Below" is strict. The complement gives the number and aggregate mass of modes at or above the threshold; for example, 41.8% of structures have at least 10% mass and account for 69.4% of recurrent failure mass.

Table 14: Sample-selected one-step repair in FR12 across fixed trajectory budgets. Means pool 125 goals and 500 repeated samples per goal–budget pair.

Budget	Failure diagnosed	Repair available	Failure repaired	Success gain	Post-edit success	Abstain	Harmful
8	66.6%	43.5%	12.3%	+5.9 pp	54.2%	56.5%	0.0%
16	84.7%	71.9%	22.8%	+10.8 pp	59.1%	28.1%	0.0%
32	95.7%	89.6%	32.7%	+15.6 pp	63.9%	10.4%	0.0%
64	99.3%	98.2%	40.3%	+19.6 pp	67.9%	1.8%	0.0%

Candidate edits use sampled trajectories only; full-graph quantities score outcomes after selection.

Table 15: FR12 targeted-success sampling for sample-selected repair.

Budget	Strategy	Failure diagnosed	Repair available	Failure repaired	Success gain	Post-edit success
32	Uniform	95.8%	89.6%	32.7%	+15.6 pp	63.9%
32	Goal-near	85.6%	86.6%	29.1%	+15.7 pp	64.0%
32	Success frontier	86.5%	87.0%	28.6%	+15.5 pp	63.8%
64	Uniform	99.3%	98.3%	40.3%	+19.6 pp	67.9%
64	Goal-near	90.7%	96.2%	36.5%	+19.0 pp	67.3%
64	Success frontier	96.5%	99.3%	43.0%	+21.6 pp	69.9%

Success-frontier sampling helps at budget 64 but not 32; all harmful-edit rates were zero.

Table 16: Exact FR12 structural-repair audit for the corrected seed-0 checkpoint.

Metric	Value
Initial non-goal-start success	48.30%
Failed start–goal pairs	8,014
Failure mass: fixed points / cycles	85.4% / 14.6%
Non-goal attractors: fixed / cycles	578 / 28
Attractors breakable by one edit	606 / 606
Immediately goal-redirectable	149 / 606
Failure pairs recovered by useful local edits	2,462
Success after all immediately useful edits	64.47%

This is an oracle-scored audit. Breaking a recurrent set is weaker than redirecting its basin to the goal.

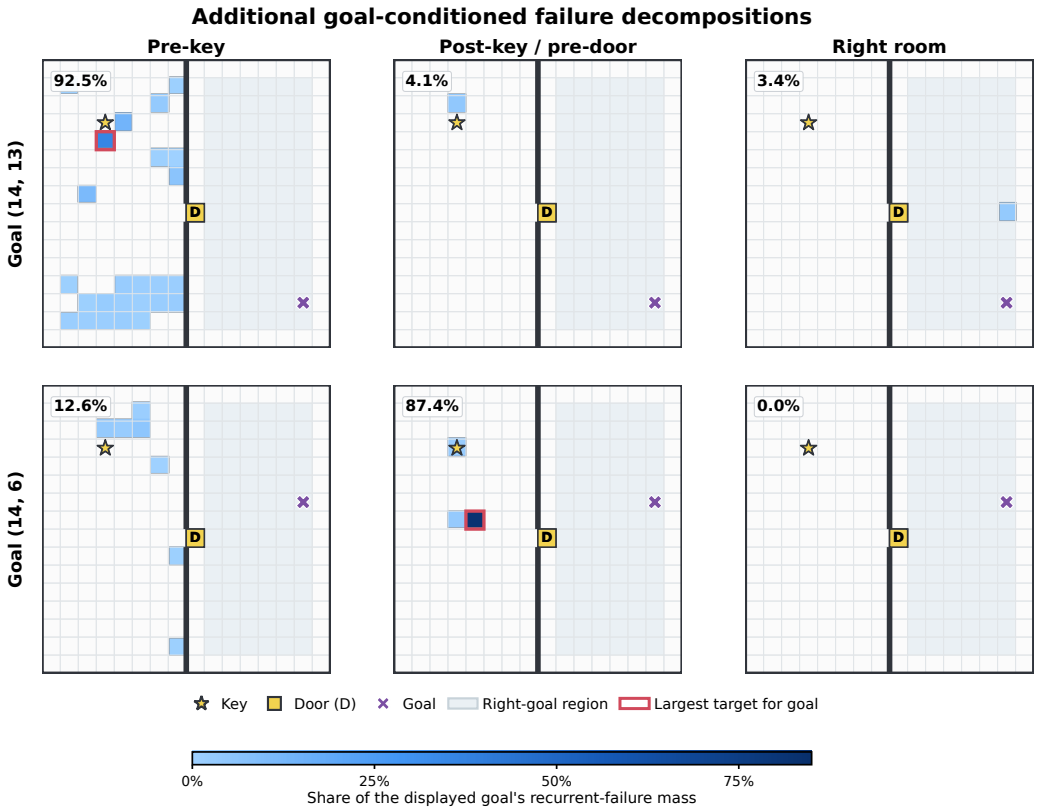

Figure 4: Additional exact MG16 goal-conditioned recurrent-failure decompositions. For goal $(14, 13)$, 92.5% of recurrent-failure mass occurs before key acquisition, with only 4.1% post-key/pre-door and 3.4% in the right room. For goal $(14, 6)$, the dominant bottleneck shifts to the post-key/pre-door stage, which contains 87.4% of recurrent-failure mass; 12.6% is pre-key and none occurs in the right room. Cell intensity denotes the share of the displayed goal's recurrent-failure mass. The red outline marks the largest recurrent target for each goal; K, D, and G denote the key, door, and goal.

Table 17: Sequential oracle repairability ceiling in FR12.

Metric	Value
Initial / final success	48.30% / 66.21%
Original failures recovered	2,777 (34.7%)
Total edits	198
Median / maximum edits per goal	2 / 5
Goals with any gain	100 / 125
Goals reaching 50% / 75% / 90%	104 / 50 / 11
Fully solved goals	2
Cycle / fixed-point edits	16 / 182

This is a ceiling rather than a deployable protocol: complete census information selects every edit.

Table 18: Iterative minimal-edge repair in MG16. Values are exact right-goal success; brackets show paired 95% intervals for the best-exit arm across 20 repetitions.

Round	No repair	Random location + best exit	Recurrent + random exit	Recurrent + best exit
0	1.42%	1.42%	1.42%	1.42%
1	1.42%	1.17%	3.73%	5.82% [5.54, 6.10]
2	1.42%	0.89%	5.97%	11.71% [11.43, 11.99]
3	1.42%	0.65%	8.89%	18.67% [18.44, 18.90]
4	1.42%	0.49%	11.34%	22.62% [22.23, 23.01]
5	1.42%	0.41%	13.40%	24.87% [24.55, 25.19]

Developmental MG16 v4 seed-0 source at approximately 1M steps; 32 IID starts per right-room goal.

Table 19: Frozen-source identity check at continuation step zero.

Metric	Seed 0	Seed 1	Seed 2	Max within-seed arm spread
Overall success	32.5	36.7	58.6	0.0
Left success	59.6	54.7	78.6	0.0
Right success	3.1	16.8	34.6	0.0
Right key	20.8	55.6	52.4	0.0
Right door	6.3	46.5	37.7	0.0
Right room entry	6.0	41.9	36.6	0.0
Right recurrent mass	96.9	83.2	65.4	0.0

Values are percentages. For each seed, the ordinary, random-trigger, and recurrence-guided archives contain exactly matching step-zero evaluation metrics; the final column reports the maximum minus minimum across the three arms in percentage points. Seed 0 is developmental; seeds 1–2 are confirmatory.

Table 20: Terminal performance after the fixed 250k-step continuation. Higher is better except for recurrent mass. Bold marks the best arm within each seed and metric.

Seed	Arm	Overall	Left	Right	Key	Door	Entry	Recur.
0	Ordinary	35.1	60.6	7.5	34.8	18.8	15.6	92.5
	Random trigger	39.8	68.5	8.8	24.4	13.8	11.9	91.3
	Recurrence guided	**45.8**	**70.2**	**19.4**	**44.6**	**22.9**	**22.9**	**80.6**
1	Ordinary	35.1	45.1	24.0	65.7	24.0	24.0	76.0
	Random trigger	40.7	**74.3**	3.6	63.4	18.9	15.4	96.4
	Recurrence guided	**75.2**	69.3	**81.7**	**83.4**	**81.7**	**81.7**	**18.3**
2	Ordinary	52.4	58.6	44.9	80.4	62.6	61.0	55.1
	Random trigger	52.3	42.9	**63.7**	65.6	63.7	63.7	**36.3**
	Recurrence guided	**59.6**	**68.9**	48.5	**86.6**	**69.4**	**69.4**	51.5

Values are percentages from the fixed terminal evaluation. "Key", "Door", and "Entry" are right-goal stage-completion rates; "Recur." is right-goal recurrent-failure mass. Seed 0 is the developmental run used while constructing the protocol; seeds 1 and 2 are confirmatory runs executed under the frozen protocol. Intermediate checkpoints were not used for model selection.

Table 21: Full continuation-exposure audit.

Seed	Arm	Steps	Episodes	Mean len.	Updates	Left eps.	Right eps.	Diagnostics	Selected	Encounters	Targeted trans. (%)	Replay samples	Successful right
0	Ordinary	250,130	1,631	153.36	13,048	409	1,222	0	0	0	0 (0.0)	0	512
	Random trigger	250,183	1,729	144.70	13,832	436	1,293	26,880	16,575	2,363	54,279 (21.7)	442,624	667
	Recurrence guided	250,135	1,582	158.11	12,656	397	1,185	26,880	17,031	1,910	41,519 (16.6)	403,968	463
1	Ordinary	250,011	2,992	83.56	23,928	767	2,225	0	0	0	0 (0.0)	0	1,871
	Random trigger	250,041	3,103	80.58	24,816	792	2,311	26,880	9,069	3,095	65,888 (26.4)	794,112	2,094
	Recurrence guided	250,121	3,270	76.49	26,152	841	2,429	26,880	7,843	1,942	39,589 (15.8)	836,864	2,179
2	Ordinary	250,202	2,856	87.61	22,840	693	2,163	0	0	0	0 (0.0)	0	1,785
	Random trigger	250,003	2,811	88.94	22,488	683	2,128	26,880	11,729	3,473	71,589 (28.6)	719,616	1,789
	Recurrence guided	250,006	2,942	84.98	23,528	715	2,227	26,880	12,197	2,365	48,185 (19.3)	752,896	1,891

"Diagnostics" counts rollouts: 84 right goals $\times$ 32 starts $\times$ 10 rediscovery rounds $= 26{,}880$ for both targeted arms. "Selected" is the actual number of selected trigger states; "Targeted trans." reports retained targeted-window transitions and their percentage of learning steps. Updates were reconstructed from the fixed rule of eight updates after each completed episode once replay reached batch size 128. Diagnostic transition counts, exact high-ϵ step counts, and trigger-stage distributions were not logged and cannot be recovered exactly. Seed 0 is developmental; seeds 1–2 are confirmatory.