WHAT HAPPENS TO THE GOAL? DISTINCT FAILURES OF POLICY ALIGNMENT IN GOAL MISGENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Goal misgeneralization is a common failure mode in which an agent behaves competently out of distribution but does not pursue the intended goal. However, we still have a weak understanding of what happens to the intended goal in misgeneralizing agents. Across three reinforcement-learning environments, we find three distinct forms of policy–goal misalignment: (1) functional target substitution, where behavior follows an alternative goal, (2) a policy shortcut, where behavior follows a learned action strategy, and (3) cue competition between intended and proxy goal cues. We distinguish these regimes by tracing where information about intended and proxy goal remains available internally, how strongly it influences the policy, and what role it plays in action selection. These distinctions are reflected in different recovery outcomes. In the cue-competition setting, steering along the direction in which the two cues compete raises intended-goal reach from 16.0% to 95.3%. Shortcut behavior instead recovers after learning new downstream wiring, raising success from 31.6% to 84.3%, even though steering the existing wiring does not recover goal-directed behavior. Under target substitution, neither intervention restores goal-directed behavior. Our results show that goal misgeneralization is not a single failure of goal representation, but distinct failures of policy alignment. Characterizing these differences provides a basis for understanding and mitigating misgeneralized behavior.

1 INTRODUCTION

Reinforcement learning specifies the intended goal only indirectly, through rewards that mark it. Whenever a proxy co-occurs with the goal during training, the rewards cannot tell the two apart, and the agent can learn the proxy in its place. This is why goal misgeneralization is common in reinforcement learning, from procedurally generated games (Langosco et al., 2022) to embodied agents (Jucys et al., 2024; Zhou et al., 2025) and large language models (Shah et al., 2022; Maniyar et al., 2026). It is a dangerous failure, as the agent stays competent and the misalignment only becomes visible once the co-occurrence is broken at test time.

So far, goal misgeneralization has mainly been characterized behaviorally (Langosco et al., 2022; Shah et al., 2022; Ramanauskas & Şimşek, 2023), and Langosco et al. (2022), for example, classify it simply by what the proxy is: a location, a direction, or a visual feature. Attempts to mitigate it intervene in training, by diversifying the training levels (Langosco et al., 2022), by level design that minimizes regret (Abdel Sadek et al., 2025), by continuing training on a new reward (Nait El Barj & Sautory, 2024; Armstrong et al., 2023), or by letting the agent ask for help (Trinh et al., 2024). Moreover, interpretability work has studied goal-related features inside individual agents (Mini et al., 2023; Trinh et al., 2024), but mostly for single environments and without connecting them to goal misgeneralization itself. The remaining gap is a quite direct question: *what actually happens to the true goal in misgeneralized agents?*

We therefore trace the intended goal through the agent with increasingly deeper questions, from the outside and the inside, and ask whether the goal is still represented, whether the policy reads it out, what organizes behavior instead, and whether goal-directed behavior can be recovered. We ask these questions in three classic environments of goal misgeneralization (Langosco et al., 2022;

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

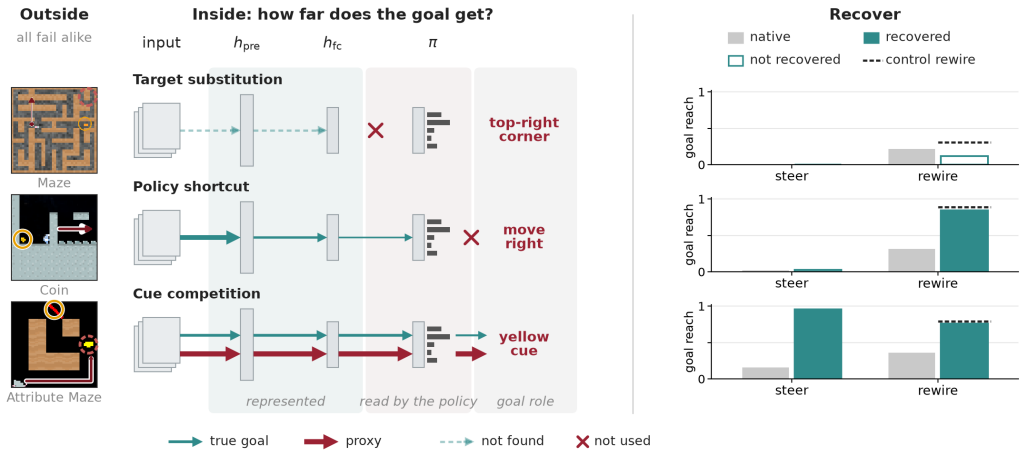

Figure 1: **All three agents fail alike from the outside, but inside the goal gets lost at different depths, and only a recovery that acts where the goal still survives brings it back.** Left: out-of-distribution frames in which each misgeneralized agent pursues the proxy (dashed ring) instead of the goal (solid ring). Middle: how far goal information travels through the agent, from the representation to the policy and to the role it plays in behavior. Right: goal reach before and after steering the existing wiring and after learning a new readout from h_{pre}, with the control-agent rewire as a dashed line. Steering is evaluated on conflict states and rewiring on held-out levels, so their native baselines differ.

Ramanauskas & Şimşek, 2023; Mini et al., 2023; Trinh et al., 2024; Nait El Barj & Sautory, 2024; Armstrong et al., 2023), MAZE, COIN and Maze II, which we call ATTRIBUTE MAZE (Figure 1).

What looks like one phenomenon from the outside is, in reality, three distinct failures of policy alignment, which show that the goal gets lost at different depths of the agent. In one environment, the goal seems not to be properly represented and is inert to the policy, so it is functionally substituted by a different target. In another, the goal is represented and weakly wired into the policy but does not function as a goal, and an action shortcut bypasses it. In the third, we find the opposite: the goal is represented, read out, and even reaches the decision, but it is overpowered by the proxy. Where the goal gets lost also determines which repair can recover it without further reinforcement learning. Steering the existing policy works where the goal reaches the decision, but fails in the other two environments. Relearning a new readout recovers the goal wherever it is still represented, including the environment where it is represented but not used as a goal. And where the goal is absent, neither repair works.

Our main contributions are:

1. We trace the intended goal through misgeneralized agents in three environments and locate three distinct failures of policy alignment at different depths, including, to our knowledge, the first internal characterization of cue competition in misgeneralized agents (Section 3).
2. We deepen the behavioral understanding of goal misgeneralization and add an internal view, proposing to view the policy through its Fisher information with respect to its internal representations. This isolates the directions that matter to the policy and reveals the geometry of the competition between goal and proxy. It also sharpens a proxy characterization of Langosco et al. (2022), as the MAZE agent follows the top-right extreme of the current maze rather than a fixed corner (Sections 3.2–3.4).
3. To our knowledge, we are the first to relate which repair succeeds to where the goal fails, and to recover goal-directed behavior in misgeneralized agents without further reinforcement learning and without retraining their representation, in two of three environments, and we explain why recovery fails in the third (Section 4).

We therefore propose that misgeneralized agents cannot be understood from their behavior alone, as only knowing where the goal is lost tells us which repair can bring it back.

2 SETUP

Problem formulation. We model each environment as an episodic Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \rho_0)$. The agent observes a rendered image of the state, which we write as s, samples actions $a_t \sim \pi(\cdot \mid s_t)$, and the reward r marks an intended goal $g^\star$. Goal misgeneralization arises when, on a training distribution Λ_{train}, reaching the goal coincides with reaching a proxy, so that the agent learns to follow the proxy instead of the goal on a shifted evaluation distribution Λ_{OOD}, where the two no longer coincide (Langosco et al., 2022; Shah et al., 2022). We write $\Gamma^\star(\tau) = 1$ when a trajectory τ reaches the goal and $\Gamma^{\text{prox}}(\tau) = 1$ when it reaches the proxy. Our main behavioral metric is the goal reach $\hat{J}^\star = \frac{1}{N}\sum_{i=1}^{N} \Gamma^\star(\tau_i)$, the fraction of N rollouts that reach the goal. Formal setup and definitions are in Appendices A and F.1.

Environments. We study three instances of goal misgeneralization, all procedurally generated video games (Cobbe et al., 2020) with a sparse reward for collecting a goal object, which defines $g^\star$ (Appendix D). In MAZE, the cheese sits in the top-right corner during training, and at test time the agent walks to the corner instead of the cheese placed elsewhere (Langosco et al., 2022). In COIN, a side-scrolling game, the coin sits at the right end of the level during training, and at test time the agent runs to the right end and ignores a coin placed elsewhere (Cobbe et al., 2019; Langosco et al., 2022). In ATTRIBUTE MAZE, a 5×5 variant of Maze II (Langosco et al., 2022; Ramanauskas & Şimşek, 2023), the rewarded object is a yellow line during training, so colour and shape are two cues for the same object, and at test time, when the cues are split across objects, the agent follows the yellow object instead of the line. In each environment, we compare three misgeneralized (GM) agents with three control (Non-GM) agents trained on diversified distributions (Appendix F).

Architecture. All agents are non-recurrent actor–critic networks with the same IMPALA architecture (Espeholt et al., 2018), trained with PPO (Schulman et al., 2017) (training and implementation details in Appendix H). Three convolutional blocks produce feature maps $H^{(k)}$, whose average-pooled vectors $h^{(k)}$ we use for probing, and a fully connected layer maps the flattened last feature map h_{pre} to h_{fc}. Most of our analyses read h_{pre}, the flattened last feature map and the last representation that keeps the spatial layout of the scene, and h_{fc}, which the policy and value heads read directly (Appendix A.1). We write $\tilde{\pi}(\cdot \mid h)$ for the policy that the downstream layers compute from a representation h, so that $\tilde{\pi}(\cdot \mid h(s)) = \pi(\cdot \mid s)$.

3 WHAT HAPPENS TO THE INTENDED GOAL?

3.1 IS THE GOAL STILL THERE?

If the true goal gets replaced by a different goal, the simplest question we can ask is whether the true goal is still in the misgeneralized agent. We test this using linear decodability (Alain & Bengio, 2017), by fitting linear probes at several points in the agent (Appendix A.1) on configuration sets that separate constructed goal concepts, scored by AUC for binary and R^2 for continuous concepts. For example, a bearing probe tests whether the agent encodes where the goal lies relative to itself. Probe construction, controls (Hewitt & Liang, 2019), and further probes are detailed in Appendix B.1.

For MAZE, the cheese object is trivially decodable from the input already, but GM agents do not decode it above randomly initialized networks, while control agents do (Figure 2). We therefore test cheese bearing and find that it is decodable in misgeneralized agents, but never better than from the agent's position alone, so the bearing may simply be read off where the agent is. With our probe sets, we could therefore not find non-trivial information about the cheese in misgeneralized agents. For COIN, the coin object is also largely decodable from the input, so we test coin bearing and find that this relative location is decodable similarly in both cohorts, well above randomly initialized networks and the agent's position alone, and most strongly in the middle layers. For ATTRIBUTE MAZE, whether the reward-carrying object is the line object is decodable in both cohorts, most clearly in early layers, and approaches the random-initialization baseline in later layers. So, goal information seems present in COIN and ATTRIBUTE MAZE agents. For MAZE, our probes did not find non-trivial decodable information, but we cannot rule out that it is present, as much of it is trivially available in the input. The bigger question is whether the policy reacts to goal changes, as information that is present need not be used (Elazar et al., 2021).

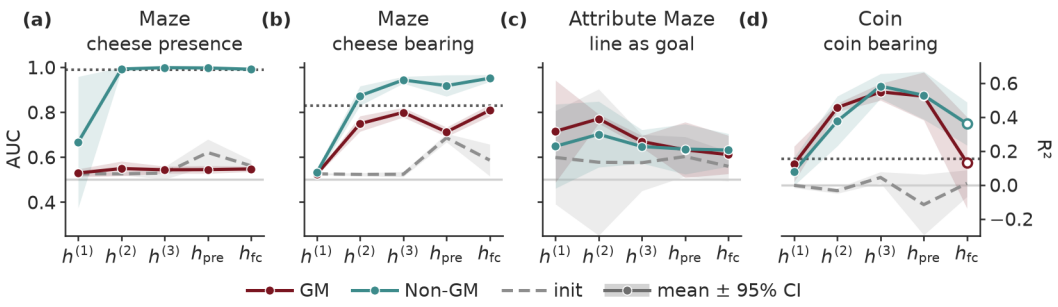

Figure 2: **Is the goal still there?** Held-out probe AUC (Coin: R^2) across network depth, with 95% t-intervals. Dashed lines are means of randomly initialized networks, dotted lines input-level controls (cheese-cell pixels for presence, agent position for bearing). Hollow Coin h_{fc} points depend on the evaluated frames and are not interpreted. Further probes and controls are in Appendix B.1 and Table 2.

3.2 Does the policy read out the intended goal?

We answer this question by intervening on the goal in the environment, from two perspectives: from the outside, we measure the change in behavior, and from the inside, whether the policy distribution moves along the corresponding representation displacement. A world edit e changes the goal in the level while keeping everything else fixed, and turns a state s into a matched state s_e, with $\Delta_e f(s) = f(s_e) - f(s)$ for any quantity f, and $e = \star$ or $e = p$ for an edit of the goal or of the proxy (Figure 11). In Maze and Coin, we relocate the goal object, the cheese or the coin, as in earlier interventions on the Maze agent (Mini et al., 2023). In Attribute Maze, in contrast, goal and proxy cue sit on the same object, so instead of moving the object, we keep all locations fixed and swap one characteristic between the two candidates: the shape swap changes the goal cue ($e = \star$) and the colour swap the proxy cue ($e = p$). All edits are then evaluated on banks of conflict states, in which the agent has to choose between the proxy and the intended goal (Appendices B.1.1 and E).

The immediate behavioral response is the change in the policy, which we measure with the KL divergence, $D_{\text{KL}}\big(\pi(\cdot \mid s) \,\|\, \pi(\cdot \mid s_e)\big)$. Inside the agent, the same edit moves the representation read by the policy by $\Delta_e(s) = \Delta_e h(s) = h(s_e) - h(s)$. We measure how strongly the policy distribution moves along this displacement with the Fisher information of the downstream policy at the base representation $h = h(s)$,

$$F_h = \mathbb{E}_{a\sim\tilde{\pi}(\cdot|h)}\big[\nabla_h \log \tilde{\pi}(a \mid h)\, \nabla_h \log \tilde{\pi}(a \mid h)^\top\big], \qquad Q_e(s) = \Delta_e(s)^\top F_h\, \Delta_e(s). \tag{1}$$

We use the Fisher information because it is the metric that the policy itself induces on its representation. It measures a displacement by how much it changes the action distribution rather than by its Euclidean length, and we use this Fisher lens throughout the paper. Up to second order, this Fisher sensitivity is exactly the KL divergence, measured from the inside, $D_{\text{KL}}\big(\tilde{\pi}(\cdot \mid h) \,\|\, \tilde{\pi}(\cdot \mid h + t\Delta_e(s))\big) = \frac{t^2}{2} Q_e(s) + o(t^2)$ (Amari, 1998; Martens, 2020) (we check this for finite edits in Appendix B.1.2). Moreover, to see what drives $Q_e(s)$, we decompose it into two factors,

$$Q_e(s) = G \cdot D_F, \qquad G = \frac{Q_e(s)/\|\Delta_e(s)\|_2^2}{\operatorname{tr}(F_h)/\dim h}, \qquad D_F = \|\Delta_e(s)\|_2^2\, \frac{\operatorname{tr}(F_h)}{\dim h}. \tag{2}$$

G asks whether the edit moves the representation in a direction the policy reads more than an average direction ($G \approx 1$ for randomly initialized networks), and D_F asks how far it moves, in units of the policy's average sensitivity, and we evaluate all inside quantities at h_{fc}.

We find that the outside and inside measures agree (Table 1). In Maze, the misgeneralized policy barely reacts to the cheese edit ($Q = 0.001$ versus 1.689 for control agents). In Coin, the policy seems to read some goal information ($Q = 0.021$ versus 0.339), so the goal appears weakly wired to the policy rather than inert as in Maze. In Attribute Maze, both cues affect the misgeneralized policy, with the colour swap larger on average ($Q = 2.126$ against 1.097 for the shape swap), while control agents barely react to colour at all ($Q = 0.029$). The decomposition locates these differences

Environment	Edit	Cohort	Outside KL	Inside (h_{fc}) Q	G	D_F
MAZE	Cheese	GM	0.015 ± 0.01	0.001 ± 0.00	0.727 ± 0.25	0.002 ± 0.00
		Non-GM	1.687 ± 0.06	1.689 ± 0.05	1.141 ± 0.39	1.667 ± 0.44
COIN	Coin	GM	0.044 ± 0.01	0.021 ± 0.01	1.350 ± 0.61	0.019 ± 0.01
		Non-GM	0.389 ± 0.11	0.339 ± 0.12	1.784 ± 0.05	0.207 ± 0.07
ATTRIBUTE MAZE	Shape (goal)	GM	0.563 ± 0.25	1.097 ± 0.45	17.165 ± 6.34	0.066 ± 0.02
		Non-GM	2.814 ± 0.24	3.569 ± 0.27	10.268 ± 4.46	0.395 ± 0.15
	Colour (proxy)	GM	1.258 ± 0.31	2.126 ± 0.83	13.759 ± 3.99	0.142 ± 0.03
		Non-GM	0.025 ± 0.02	0.029 ± 0.03	2.121 ± 0.89	0.013 ± 0.01

Table 1: **The policy's response to the same world edit, seen from outside and inside.** KL: KL divergence measuring behavioral change. Q: local policy sensitivity, decomposed into the relative gain G and the displacement D_F. Mean $\pm$ SD over seeds of state means (KL) or medians.

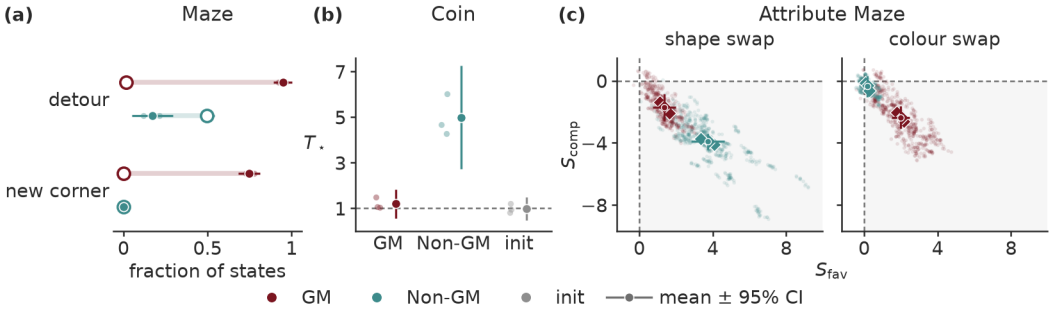

Figure 3: **What organizes behavior instead?** (a) MAZE: fraction of states in which agents move toward the top-right corner or the new corner (filled) versus the cheese or the old corner (open), on detour states and after a new top-right extreme is opened. (b) COIN: $T_\star$, the share of the edit-induced change on coin-directed actions relative to their probability. The dashed line $T_\star = 1$ marks no preferred direction. (c) ATTRIBUTE MAZE: change in log support of the favoured (S_{fav}) and competing (S_{comp}) candidate relative to neutral actions under each swap. Faint dots are states, diamonds seed means, and shading marks the push–pull quadrant. Mean $\pm$ 95% CI over seeds.

further (Table 1). In MAZE and COIN, the gap comes from the displacement rather than from the gain. The goal edit moves the representation by $D_F = 0.002$ versus 1.667 in MAZE and 0.019 versus 0.207 in COIN, while the gain of misgeneralized agents is not higher than that of control agents. In ATTRIBUTE MAZE, the picture is reversed. Both swaps move the representation only a little ($D_F = 0.066$ and 0.142), but along directions the policy reads strongly, with $G = 17.2$ for the shape swap and $G = 13.8$ for the colour swap, compared to $G \approx 1$ for an average direction. So the three failures differ in how much goal information reaches the policy: it is negligible in MAZE, weak in COIN, and strong in ATTRIBUTE MAZE, where the proxy reaches the policy even more strongly. Sensitivity, however, only captures the immediate action distribution, so we ask next what actually organizes the agent's behavior.

3.3 WHAT ORGANIZES BEHAVIOR INSTEAD?

Where the goal is fully inert, as in MAZE, the question is what actually replaces it: a directional shortcut, a memorized location, or a spatial concept. We first select detour states, on which every action that makes progress toward the top-right corner points neither up nor right. GM agents still choose these detour actions on 0.948 of the states, which rules out a directional shortcut (Figure 3a). We then open a path to a new top-right extreme, inspired by the corner edit of Mini et al. (2023), and ask which corner the agent follows. It reaches the new corner first in 0.748 of the rollouts and the old one never, and a new extreme placed away from the screen corner is still reached first in 0.792 of rollouts, against 0.114 for the old one, which rules out a memorized location (Figure 3a). Combined with the inert goal, this shows a functional target substitution, and we refer to Appendix B.2.1 for the full analysis. For MAZE, the substitute target is thus not a fixed place but a spatial relation, the

top-right extreme of the maze. This refines Langosco et al. (2022), who classify this failure as a location proxy but do not test what underlies it.

A weakly wired goal, as in COIN, raises a different question: is this a weaker substitution, or is behavior organized by something other than a target? Two observations point to a fixed action rule. First, when the end of the level is made traversable, GM agents run past the former endpoint in 0.94 of the episodes that reach it and keep choosing rightward actions at the same rate after passing it, which reconfirms the observation of Langosco et al. (2022). Second, the coin's weak influence carries no direction. When the coin moves, the change in the policy is spread over the actions in proportion to their probability. We measure this with $T_\star$, the share of the edit-induced change that falls on coin-directed actions, divided by the probability these actions hold, so that $T_\star = 1$ means no preferred direction (Appendix B.2.2). Coin-directed actions hold about 7% of the probability and receive about 9% of the change ($T_\star = 1.21$), close to randomly initialized networks ($T_\star = 0.99$) and far from Non-GM agents ($T_\star = 5.00$) (Figure 3b). The coin is therefore perceived as a change in the scene rather than as a goal that orients behavior. What organizes behavior is neither the goal nor a substitute target, but the simple rightward action rule that solved training, so goal misgeneralization in COIN is best described as a policy shortcut, in the sense of shortcut learning (Geirhos et al., 2020). A deeper analysis, including Fisher saliency maps, is given in Appendix B.2.2 and Appendix B.1.3.

When goal and proxy are both wired into the policy, as in ATTRIBUTE MAZE, the question becomes how the two interact. For a swap that favours one candidate, we measure S_{fav} and S_{comp}, the change in log support of the favoured candidate and of its competitor, each relative to the neutral actions (Appendix B.2.3). In GM agents, both the shape swap and the colour swap push support toward their candidate and pull it from the competitor, while Non-GM agents show this push–pull only for shape (Figure 3c, Appendix B.2.3). The two cues thus do not act independently but compete for the same choice.

What replaces the goal thus follows from where it fails: a substitute target where we find no goal information that the policy uses, an action rule where the goal is represented but not read as a goal, and a competing cue where the goal is read but overpowered. How this competition is organized in ATTRIBUTE MAZE is what we examine next.

3.4 WHEN GOAL AND PROXY BOTH REACH THE POLICY, HOW DO THEY COMPETE?

ATTRIBUTE MAZE is an example of a regime in which goal and proxy both act on the policy and push and pull against each other in a competition. We now want to understand geometrically where this push and pull comes from, and why the proxy wins. For each conflict state, the shape swap and the colour swap move the representation by $\Delta_\star(s)$ and $\Delta_p(s)$, and we compare such directions with the ordinary cosine $\cos_E$ in raw activations and with its Fisher variant $\cos_F(a, b) = a^\top F_h b \,/\, \sqrt{a^\top F_h a \; b^\top F_h b}$ in the policy's metric, which with F_h from Equation equation 1 only counts directions the policy actually reads. We find that the two cues push the policy in opposite directions (Figure 4a). In GM agents, $\cos_F(\Delta_\star, \Delta_p) = -0.919$, so for the policy the two displacements are nearly antipodal, whereas control agents are opposed more weakly (-0.557) and randomly initialized networks not at all ($+0.254$).

This raises the question whether there is one global competition axis. We therefore take the competition direction $d_{\text{comp}}(s) = \Delta_p(s) - \Delta_\star(s)$ in each state and compare it between states with the Euclidean and Fisher cosines from above (Figure 4b, c). In the Fisher metric, states of the same conflict share almost the same competition direction (0.942, against 0.148 at initialization), and states of the opposite conflict point the opposite way, while in raw activations the structure is visible but more heterogeneous and weaker (0.589). Thus, the policy reads the competition on one signed axis per conflict, which we call u_{comp} (Appendix B.3.2).

Finally, we ask who is actually winning on this axis. We project each cue's displacement onto u_{comp} with $\alpha_F(\Delta) = u_{\text{comp}}^\top F_h \Delta / u_{\text{comp}}^\top F_h u_{\text{comp}}$, which measures how far the cue pulls the state along it in the policy's Fisher geometry. In GM agents, the colour cue pulls 1.564 times as hard as the shape cue, whereas in control agents the ratio is only 0.067 (Figure 4d). The goal is therefore not lost in this regime but still pulls on a shared axis, only more weakly than the proxy, which suggests steering along this axis, as we test next (Appendices B.3 and B.3.3).

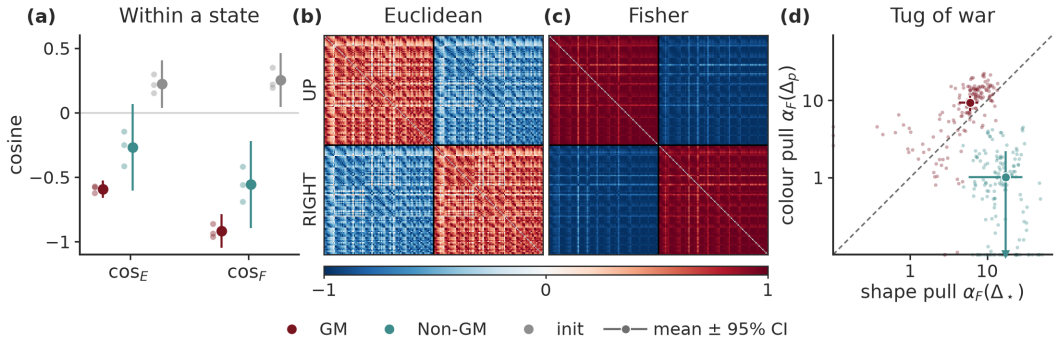

Figure 4: **How goal and proxy compete in ATTRIBUTE MAZE.** (a) Cosine between the two cue displacements within a state. (b, c) Cosine between the competition directions of pairs of states, grouped by whether the goal candidate lies UP or RIGHT. (d) Pull of each cue on the shared axis, per state. Above the dashed line, the proxy pulls harder. The colour/shape pull ratio is 1.56 for GM and 0.07 for Non-GM. All at h_{fc}, black background, mean with 95% t-interval over three seeds.

4 CAN THE GOAL BE RECOVERED?

4.1 CAN THE GOAL BE RECOVERED THROUGH THE EXISTING WIRING?

What we found in Section 3 already tells us where steering can work. In MAZE, the goal is inert, and in COIN, it is only weakly read and does not direct navigation. In ATTRIBUTE MAZE, however, the goal is fully wired into the policy and loses its competition with the proxy on a shared axis, and this axis is something we can push on. To steer the three environments with one protocol, we contrast, at every state s, the representation of two counterfactual versions of it, $s^\star$ in which the evidence supports the goal and s^p in which it supports the proxy. Their difference $d(s) = h_{\text{fc}}(s^\star) - h_{\text{fc}}(s^p)$ is an additive steering vector (Turner et al., 2023b), along which we push the representation with dose β, $h'_{\text{fc}}(s_t) = h_{\text{fc}}(s_t) + \beta\, d(s_t)$. In MAZE, $s^\star$ and s^p place the cheese at its position in the level and in the top-right corner. In COIN, they place the coin at its position behind the agent and ahead of it on the shortcut path. In ATTRIBUTE MAZE, $s^\star$ and s^p are the colour and shape swaps, which put the yellow line on the goal and on the proxy candidate, so $d(s)$ is the competition direction $d_{\text{comp}}(s)$ of Section 3.4. We rebuild the direction as the agent moves, which makes the push state-dependent and allows for different geometries rather than one shared axis. As controls, we reverse the push, replace it by random directions of equal norm, or leave it out.

As we derived in Section 3.4, a simpler intervention is possible in ATTRIBUTE MAZE. As the policy reads the competition on one signed axis per conflict, we propose to replace the state-dependent direction by the identified shared axis u_{comp}, $h'_{\text{fc}}(s_t) = h_{\text{fc}}(s_t) + \beta\, \sigma(s_0)\, \|d(s_0)\|_2\, u_{\text{comp}}$, where $\sigma(s_0) \in \{-1, +1\}$ is the orientation of the conflict. This push is computed once at the start state s_0 and held fixed, and its norm matches the state's own direction, so that only the direction changes. We compare it with an unsigned push that drops σ and, as both are held fixed, with $d(s_0)$ held fixed as well (a Fisher-projected variant is in Appendix C.1.2).

In ATTRIBUTE MAZE, steering recovers the goal cleanly (Figure 5c). In GM agents, goal reach rises from 0.160 to 0.953 at $\beta = 1$, the reversed push drives it down to 0.007, and the random push leaves it unchanged at 0.171. The shared axis carries this effect as well (Figure 5d). At $\beta = 0.25$, the sign-gated push raises goal reach to 0.587, as much as $d(s_0)$ held fixed (0.560), and reversing it drives it close to zero. Without the sign $\sigma(s)$, the same axis helps one conflict orientation and hurts the other, so the unsigned push reaches only 0.300, and reversing it gives almost the same 0.307.

In MAZE and COIN, the same construction does not move the misgeneralized policy. GM agents in MAZE reach the cheese in at most 0.002 of states under any push, while in COIN GM goal reach stays at 0.018 under every push up to $\beta = 4$, and the corrective push is no more effective than a random one. We rule out that the construction is at fault, as control agents do respond to it. In MAZE, the reversed push removes their cheese pursuit almost entirely, from 0.567 to 0.002 at $\beta = 1$, while random pushes of equal norm only reduce it to 0.340. In COIN, control agents react to both the corrective push, from 0.121 to 0.373 at $\beta = 4$, and the reversed push, down to 0.012, and only

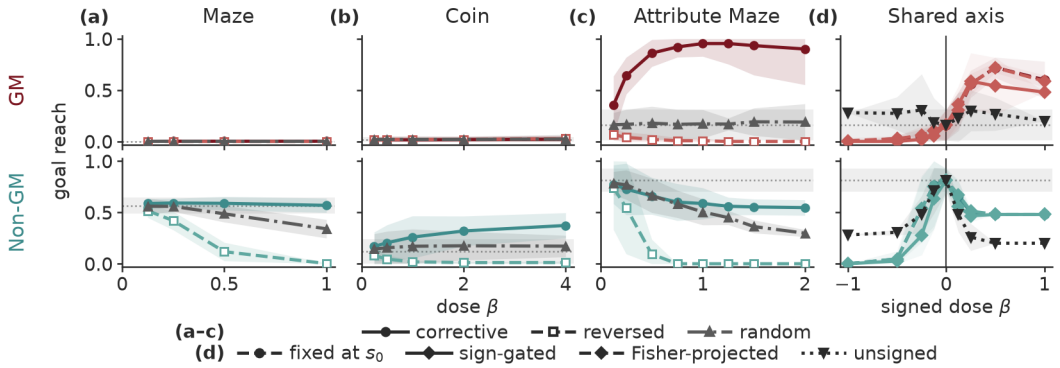

Figure 5: **Steering the existing wiring.** Goal reach $\hat{J}^\star$ under steering at h_{fc} as a function of the dose β, GM agents (top) and control agents (bottom). (a–c) State-dependent corrective, reversed and random pushes of equal norm. (d) ATTRIBUTE MAZE, shared-axis pushes (sign-gated, Fisher-projected, unsigned) against the state-dependent direction held fixed at s_0, negative β reversed. Grey bands and dotted lines show the unsteered baseline. Mean with 95% t-interval over three seeds.

slightly to random pushes (0.171). As the steering direction is built from edits that the GM policy barely responds to in COIN and not at all in MAZE, this failure is consistent with Section 3.

Steering the existing wiring thus recovers goal orientation when the goal information still reaches the decision. In ATTRIBUTE MAZE, the goal lost a competition on an axis the policy already reads, so pushing along that axis turns the policy towards it again. In MAZE and COIN, there is no such axis to push on, as the goal information that reaches the policy is too weak in COIN and not detectable in MAZE. Whether the goal information in the representation is still enough to learn an orientation towards the true goal is what we test next, by training new wiring. Details are given in Appendix C.1.1 and C.1.2.

4.2 CAN THE GOAL BE RECOVERED THROUGH NEW WIRING?

To understand at what depth the goal is truly lost, we learn a new wiring at two points in the architecture (Appendix A.1). We relearn either only the policy head on top of h_{fc} (4k trained parameters) or the head together with the fully connected layer that maps h_{pre} to h_{fc} (528k). In both cases, we train a freshly initialized copy with behavioral cloning (Pomerleau, 1991) on goal-directed actions, so that every agent receives the same target behavior and a failed recovery cannot be caused by exploration or credit assignment. In MAZE and ATTRIBUTE MAZE, the targets are oracle actions from a breadth-first search towards the true goal, while in COIN, where no oracle exists, we use the actions of the control agents (control-agent targets for all environments in Appendix C.2). As baselines, we rewire the control agents in the same way, compare every rewire to the native reach of both cohorts and to an untrained head, and remove the confound of head size with a fixed random remix, which replaces the learned map from h_{pre} to h_{fc} by a random one and trains only a head of the h_{fc} size.

Where the goal information is seemingly inert for the policy, as in MAZE, a new wiring does not recover goal orientation (Figure 6a). GM agents reach the cheese on 0.116 of episodes from h_{pre} and 0.119 from h_{fc}, below their native 0.219. As behavioral cloning alone leaves even the control rewire below its native reach (0.308 against 0.347), we also relabel on the states the rewired policy visits (Ross et al., 2011), which restores the control agents (0.351) but leaves GM agents below 0.13 (Appendix C.2). We conclude that the goal information is not recovered by a new wiring, as expected from the inert goal in the probes, the Fisher sensitivity and the steering results.

In COIN, where the goal is only weakly wired and steering was unsuccessful, the question is whether a new wiring can use the goal information that still reaches the late layers. From h_{pre}, GM agents recover the goal (Figure 6b), reaching 0.843, which is close to the control rewire (0.888) and well above the untrained head (0.626) and their native reach (0.316). From h_{fc}, they reach 0.666, which is where the untrained head lies (0.630). This matches the probes, where coin bearing is decodable in both cohorts, most strongly in the middle layers and through h_{pre}. Moreover, the remix shows

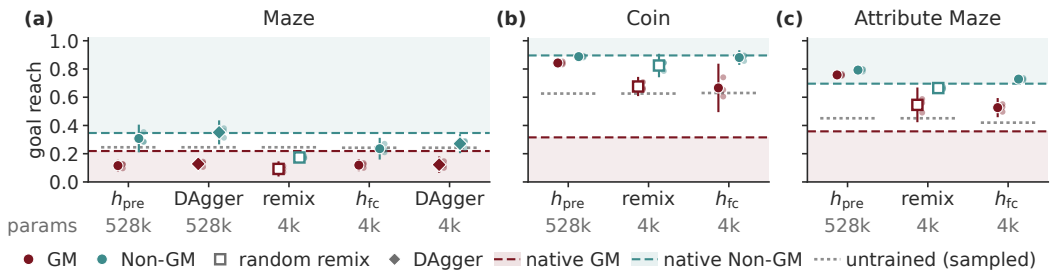

Figure 6: **Recovering the goal through new wiring.** Goal reach $\hat{J}^\star$ after relearning the network downstream of h_{pre} or h_{fc} by behavioral cloning, with oracle targets in MAZE and ATTRIBUTE MAZE and control-agent targets in COIN. Remix is a fixed random map of h_{pre} with a trained head of the h_{fc} size. DAgger adds three rounds of relabeling on the states the rewired policy visits. Grey numbers give the trained parameters. Points are means with 95% confidence intervals across agents, small points are individual agents.

where the information is lost, as a head on a random remix of GM h_{pre} reaches 0.676, as much as a head on the learned GM h_{fc}, while the same remix of control h_{pre} keeps 0.825. So the wiring the GM agents learned between h_{pre} and the policy head seems to keep no more goal information than a random mapping. Together with the steering null, this shows that in COIN the goal is still represented but lost in the readout, and a new readout can recover it. Retraining only the last layer, which suffices for spurious correlations in Kirichenko et al. (2023), does not suffice here.

Finally, in ATTRIBUTE MAZE, where goal and proxy information both reach the decision, rewiring from h_{pre} allows recovery and brings GM agents close to the control rewire (0.758 against 0.792, native 0.358, Figure 6c). From h_{fc}, they reach 0.526, above their native policy but not higher than the remix (0.546), while the control rewire reaches 0.728. A freshly trained head is a weaker test of what reaches h_{fc} than the native head, which steering shows to read the goal there. A new readout thus recovers the goal wherever goal information survives in the representation it reads, from h_{pre} in COIN and ATTRIBUTE MAZE, but nowhere in MAZE.

5 DISCUSSION AND CONCLUSION

Conclusion. While from the outside all three agents seem to fail in the same way, as they pursue a proxy instead of the intended goal, this is not one failure but three. Tracing the goal through the agents' behavior and representations reveals three distinct failures of policy alignment, which differ in how far goal information propagates into the agent and in what organizes their behavior instead. And the depth at which the goal gets lost explains which recovery works, as the goal can only be recovered where it is still present and usable. Where the goal is still wired into the policy but overpowered by a competing proxy, we can steer the existing policy towards it. Where it is lost in the readout but still present in the representation, the existing wiring is not good enough for steering, but a new readout can reestablish the goal relationship, turning information that is merely represented into goal-directed behavior. And where the goal is inert to the policy and functionally substituted by another target, neither attempt works. As behavior alone cannot tell these failures apart, we propose to first diagnose where the goal got lost, and only then choose how to recover it.

Discussion and outlook. Beyond these results, we want to pose a hypothesis that we do not test here, namely that the structure of the training ambiguity shapes how the goal fails. In ATTRIBUTE MAZE, goal and proxy are two attributes of the same object, so the agent may learn only its features, and once they are split across objects, both push towards a goal, with the proxy pushing harder. In COIN, moving right is a much simpler mapping from input to action that solves training in exactly the same way, and networks are known to latch onto such simple solutions (Geirhos et al., 2020; Hermann & Lampinen, 2020). In MAZE, a spatial relation to the top-right extreme is enough to navigate, which may be why we find no cheese information that the policy uses. Moreover, how deep the goal reaches and what replaces it are in principle separate dimensions, which happen to line up in our agents. Testing this causally, and extending it to more environments and to how these fingerprints emerge during training, is an exciting next step, for which we laid the first groundwork.

AI USE STATEMENT

Generative artificial intelligence aided in the development of this paper. Specifically, it helped with drafting and polishing prose from the authors' dictation, polishing mathematics, literature search, writing code for analyses and figures, and auditing reported numbers against the underlying results. We have reviewed and verified all LLM-generated text, code and analyses. All ideas behind the method and the scientific claims are products of the authors' own minds.

REFERENCES

Karim Abdel Sadek, Matthew Farrugia-Roberts, Usman Anwar, Hannah Erlebach, Christian Schroeder de Witt, David Krueger, and Michael Dennis. Mitigating goal misgeneralization via minimax regret. In *Reinforcement Learning Conference (RLC)*, 2025. URL https://arxiv.org/abs/2507.03068.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations (ICLR), Workshop Track*, 2017.

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998. doi: 10.1162/089976698300017746.

Stuart Armstrong, Alexandre Maranhão, Oliver Daniels-Koch, Patrick Leask, and Rebecca Gorman. CoinRun: Solving goal misgeneralisation, 2023. URL https://arxiv.org/abs/2309.16166.

Akanksha Atrey, Kaleigh Clary, and David Jensen. Exploratory not explanatory: Counterfactual analysis of saliency maps for deep reinforcement learning. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.05743.

Thomas Bush, Stephen Chung, Usman Anwar, Adrià Garriga-Alonso, and David Krueger. Interpreting emergent planning in model-free reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DzGe40glxs.

Karl Cobbe, Oleg Klimov, Christopher Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1282–1289. PMLR, 2019.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2048–2056. PMLR, 2020.

Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Autotelic agents with intrinsically motivated goal-conditioned reinforcement learning: A short survey. *Journal of Artificial Intelligence Research*, 74:1159–1199, 2022. doi: 10.1613/jair.1.13554.

Quentin Delfosse, Sebastian Sztwiertnia, Mark Rothermel, Wolfgang Stammer, and Kristian Kersting. Interpretable concept bottlenecks to align reinforcement learning agents. In *Advances in Neural Information Processing Systems*, volume 37, pp. 66826–66855, 2024. doi: 10.52202/079017-2135. URL https://arxiv.org/abs/2401.05821.

Diego Dorn, Neel Alex, and David Krueger. Goal misgeneralization as implicit goal conditioning. In *NeurIPS 2023 Workshop on Goal-Conditioned Reinforcement Learning (GCRL)*, 2023. URL https://openreview.net/forum?id=QT4tXTqTTr.

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

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1407–1416. PMLR, 2018.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1515–1528. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/florensa18a.html.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 9574–9586, 2021.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. doi: 10.1038/s42256-020-00257-z.

Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding Atari agents. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1792–1801. PMLR, 2018. URL https://proceedings.mlr.press/v80/greydanus18a.html.

Katherine L. Hermann and Andrew K. Lampinen. What shapes feature representations? Exploring datasets, architectures, and training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9995–10006, 2020.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1275.

Jacob Hilton, Nick Cammarata, Shan Carter, Gabriel Goh, and Chris Olah. Understanding RL vision. *Distill*, 5(11), 2020. doi: 10.23915/distill.00029.

Karolis Jucys, George Adamopoulos, Mehrab Hamidi, Stephanie Milani, Mohammad Reza Samsami, Artem Zholus, Sonia Joseph, Blake Richards, Irina Rish, and Özgür Şimşek. Interpretability in action: Exploratory analysis of VPT, a Minecraft agent. In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024. URL https://arxiv.org/abs/2407.12161.

Sham M. Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=Zb6c8A-Fghk.

Cassidy Laidlaw, Shivam Singhal, and Anca Dragan. Correlated proxies: A new definition and improved mitigation for reward hacking. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=msEr27EejF.

Lauro Langosco di Langosco, Jack Koch, Lee D. Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12004–12019. PMLR, 2022. URL https://proceedings.mlr.press/v162/langosco22a.html.

Hojoon Lee. Training procgen environment with PyTorch. https://github.com/joonleesky/train-procgen-pytorch, 2020.

Yoonho Lee, Annie S. Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=APuPRxjHvZ.

Suyash Maniyar, Armaan Sandhu, and Abhishek Mishra. Measuring reward hacking and reasoning-answer decoupling under position-confounded optimization. In *COLM 2026 Workshop on AI Measurement Science*, 2026. URL https://arxiv.org/abs/2608.15445.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

Ulisse Mini, Peli Grietzer, Mrinank Sharma, Austin Meek, Monte MacDiarmid, and Alexander Matt Turner. Understanding and controlling a maze-solving policy network, 2023. URL https://arxiv.org/abs/2310.08043.

Houda Nait El Barj and Theophile Sautory. Reinforcement learning from LLM feedback to counteract goal misgeneralization, 2024. URL https://arxiv.org/abs/2401.07181.

Robert G. Newcombe. Interval estimation for the difference between independent proportions: Comparison of eleven methods. *Statistics in Medicine*, 17(8):873–890, 1998. doi: 10.1002/(SICI)1097-0258(19980430)17:8⟨873::AID-SIM779⟩3.0.CO;2-I.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=JYtwGwIL7ye.

Dean A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991. doi: 10.1162/neco.1991.3.1.88.

Karolis Ramanauskas. Colour versus shape goal misgeneralization in reinforcement learning: A case study (models and test data), 2023.

Karolis Ramanauskas and Özgür Şimşek. Colour versus shape goal misgeneralization in reinforcement learning: A case study. In *NeurIPS 2023 Workshop on Attributing Model Behavior at Scale (ATTRIB)*, 2023. URL https://arxiv.org/abs/2312.03762.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1312–1320, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schaul15.html.

John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897. PMLR, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Rohin Shah, Vikrant Varma, Ramana Kumar, Mary Phuong, Victoria Krakovna, Jonathan Uesato, and Zac Kenton. Goal misgeneralization: Why correct specifications aren't enough for correct goals, 2022. URL https://arxiv.org/abs/2210.01790.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations, Workshop Track*, 2014. URL https://arxiv.org/abs/1312.6034.

Tu Trinh, Mohamad H. Danesh, Nguyen X. Khanh, and Benjamin Plaut. Getting by goal misgeneralization with a little help from a mentor. In *NeurIPS 2024 Workshop on Towards Safe and Trustworthy Agents (SATA)*, 2024. URL https://arxiv.org/abs/2410.21052.

Alexander Matt Turner, Peli Grietzer, and Lisa Thiergart. Maze-solving agents: Add a top-right vector, make the agent go to the top-right. LessWrong, cross-posted to the AI Alignment Forum, 31 March 2023, 2023a. URL https://www.lesswrong.com/posts/gRp6FAWcQiCWkouN5/maze-solving-agents-add-a-top-right-vector-make-the-agent-go.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2023b.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.

Chenxiao Zhao, P. Thomas Fletcher, Mixue Yu, Yaxin Peng, Guixu Zhang, and Chaomin Shen. The adversarial attack and detection under the Fisher information metric. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5869–5876, 2019. doi: 10.1609/aaai.v33i01.33015869.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization, 2025. URL https://arxiv.org/abs/2510.03827.

A Formal Setup and Notation

Overview of the appendix. Appendix A gives the formal setup, the network architecture and the target geometry used throughout.[1] Appendices B and C contain the extended experiments behind Sections 3 and 4. The supporting material follows: the environments (Appendix D), the evaluation banks (Appendix E), the behavioral qualification of the studied checkpoints (Appendix F), related work in detail, and implementation and training details (Appendix H).

Environment and policy. We consider an episodic Markov decision process

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \rho_0),$$

with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel $P(s' \mid s, a)$, reward function $r(s, a, s')$, and initial-state distribution ρ_0. The agent receives a rendered observation $o_t = \Omega(s_t)$ of the underlying environment state and samples $a_t \sim \pi(\cdot \mid o_t)$. Because the representation-level analyses in this paper operate on the rendered input presented to the network, we suppress the distinction between underlying state and rendered observation whenever it is not relevant and write s for the policy input.

Definition 1 (Intended-goal achievement)**.** *Let $g^\star$ denote the intended task goal and let $\mathcal{S}^{g^\star} \subseteq \mathcal{S}$ denote the set of environment states in which that goal is satisfied. For a trajectory $\tau = (s_0, a_0, \ldots, s_T)$, we define*

$$\Gamma^\star(\tau) = \mathbf{1}\left\{\exists\, t \le T : s_t \in \mathcal{S}^{g^\star}\right\}.$$

Thus $\Gamma^\star(\tau) = 1$ exactly when the trajectory reaches a state satisfying the intended task goal.

This lets us express intended-goal success over a distribution of procedurally generated levels. Let $\xi \in \Xi$ index a level and let Λ denote a distribution over levels. We define

$$J^\star(\pi; \Lambda) = \mathbb{E}_{\xi \sim \Lambda,\, \tau \sim P_\pi(\tau \mid \xi)}\left[\Gamma^\star(\tau)\right],$$

where $P_\pi(\tau \mid \xi)$ is the trajectory distribution induced by π on level ξ. Because $\Gamma^\star$ is binary, $J^\star$ is simply the probability that the policy achieves the intended goal under Λ.

In the sparse-reward environments studied here, reward marks successful completion of the intended task. We nevertheless keep r and $g^\star$ conceptually distinct. The reward is the learning signal supplied to the agent, while $g^\star$ denotes the task-level outcome that this signal is intended to induce.

Proxy regularities and distribution shift. We write Λ_{train} for the distribution of levels encountered during training and Λ_{OOD} for the shifted evaluation distribution.

Goal misgeneralization becomes possible when successful behavior under Λ_{train} is systematically associated with another regularity (Langosco et al., 2022; Shah et al., 2022). We denote such a training-correlated trajectory property by $\Gamma^{\text{prox}}(\tau)$, so that $\Pr_{\Lambda_{\text{train}}}\left(\Gamma^{\text{prox}}(\tau) = 1 \mid \Gamma^\star(\tau) = 1\right) \approx 1$. The parallel notation with $\Gamma^\star$ is deliberate: both quantities describe properties of the same trajectory, but only $\Gamma^\star$ is required to correspond to a designated goal set $\mathcal{S}^{g^\star}$.

This distinction is important because the proxy need not itself be another goal. In Maze, the proxy is associated with the top-right corner of the maze. In Attribute Maze, it is an object attribute correlated with the rewarded attribute during training. In Coin, it is a directional regularity associated with continuing rightward. We therefore use the term *proxy regularity* rather than assuming that every environment contains a separate proxy goal.

When a binary proxy criterion is available, strong training-time coupling can be expressed as

$$\Pr_{\substack{\xi \sim \Lambda_{\text{train}} \\ \tau \sim P_\pi(\tau \mid \xi)}}\left(\Gamma^{\text{prox}}(\tau) = 1 \mid \Gamma^\star(\tau) = 1\right) \approx 1.$$

For non-binary proxy statistics, the analogous requirement is that successful training trajectories make Γ^{prox} strongly predictable. We therefore use Γ^{prox} as generic notation for the training-correlated regularity rather than imposing one common functional form across environments.

[1] Code: `https://anonymous.4open.science/r/gms-missing-goal-6F22/`

The role of Λ_{OOD} is to break this training-time association. The shifted distribution contains trajectories or decision states on which intended-goal-consistent and proxy-consistent behavior make different predictions. In the binary case,

$$\Pr_{\substack{\xi\sim\Lambda_{\mathrm{OOD}}\\ \tau\sim P_\pi(\tau|\xi)}} [\Gamma^\star(\tau) \neq \Gamma^{\mathrm{prox}}(\tau)] > 0.$$

More generally, Λ_{OOD} creates situations in which following the intended goal and following the training-correlated regularity can be behaviorally distinguished. The environment-specific criteria used to identify capable but proxy-following policies are defined in Appendix F.

Network representations. All agents use a convolutional actor–critic architecture with three spatial IMPALA blocks (Espeholt et al., 2018) followed by a shared fully connected representation. Agents are trained with PPO (Schulman et al., 2017) using the code base of Langosco et al. (2022), itself based on Lee (2020), and ATTRIBUTE MAZE agents with the training code released by Ramanauskas & Şimşek (2023). We write the shared forward computation as

$$s \xrightarrow{\Phi^{(1)}} H^{(1)}(s) \xrightarrow{\Phi^{(2)}} H^{(2)}(s) \xrightarrow{\Phi^{(3)}} H^{(3)}(s) \xrightarrow{\mathrm{vec}} h_{\mathrm{pre}}(s) \xrightarrow{\Psi} h_{\mathrm{fc}}(s),$$

after which separate policy and value heads read out from the shared representation.

For state s,

$$H^{(k)}(s) = \Phi^{(k)}\Big(H^{(k-1)}(s)\Big), \qquad k \in \{1,2,3\},$$

with $H^{(0)}(s) = s$. Here

$$H^{(k)}(s) \in \mathbb{R}^{C_k \times X_k \times Y_k}$$

denotes the spatial output of block k, the residual sum before the rectification that the next stage applies. Its entry

$$H^{(k)}_{cxy}(s)$$

is the activation of channel c at spatial position (x, y), and

$$H^{(k)}_{:xy}(s) \in \mathbb{R}^{C_k}$$

denotes the channel vector at that position.

The final spatial representation is flattened as

$$h_{\mathrm{pre}}(s) = \mathrm{vec}\,\mathrm{ReLU}\Big(H^{(3)}(s)\Big) \in \mathbb{R}^{d_{\mathrm{pre}}}.$$

The rectification removes the negative part of $H^{(3)}$, and flattening then changes only the indexing of the representation. The shared fully connected layer then gives

$$h_{\mathrm{fc}}(s) = \Psi(h_{\mathrm{pre}}(s)) = \mathrm{ReLU}(W_{\mathrm{fc}} h_{\mathrm{pre}}(s) + b_{\mathrm{fc}}) \in \mathbb{R}^{d_{\mathrm{fc}}}.$$

When the specific representation site is not important, we write

$$h^{(m)}(s) \in \mathbb{R}^{d_m}, \qquad m \in \{1, 2, 3, \mathrm{pre}, \mathrm{fc}\}.$$

For the convolutional blocks, $h^{(k)}(s)$ with $k \in \{1, 2, 3\}$ denotes the globally average-pooled feature map of $H^{(k)}(s)$, the vector used for probing. The final spatial representation h_{pre} is instead kept in full, so $h^{(3)}$ and h_{pre} are different objects. Spatial analyses retain the tensor notation $H^{(k)}(s)$ and the corresponding cell-wise vectors $H^{(k)}_{:xy}(s)$.

Definition 2 (World-edit representation displacement)**.** *For a base state s and controlled environment edit e, let*

$$s_e$$

denote the matched state obtained after applying edit e. At representation site m, the world edit induces the displacement

$$\Delta^{(m)}_e(s) = h^{(m)}(s_e) - h^{(m)}(s).$$

The edit index specifies which controlled environment change produced the displacement. When the edit concerns the intended goal, we write

$$\Delta^{(m)}_\star(s),$$

and when a corresponding proxy edit is available, we write

$$\Delta^{(m)}_p(s).$$

The environment-specific counterfactual families are defined in Section B.1.1.

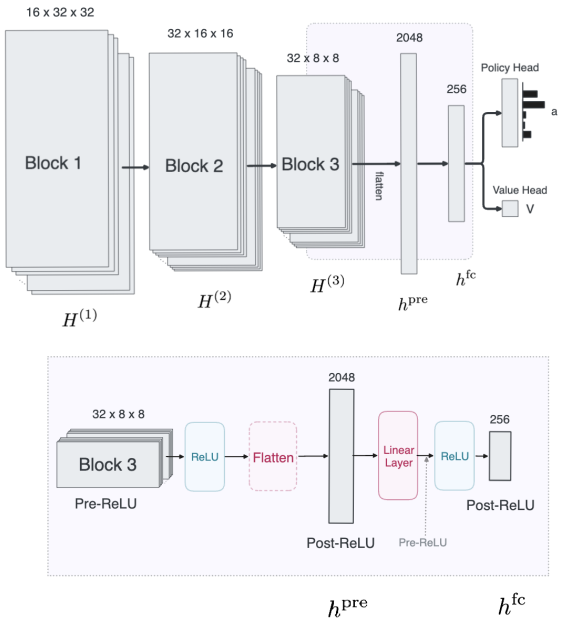

Figure 7: Shared actor-critic network. Three residual blocks produce $H^{(1)}$, $H^{(2)}$ and $H^{(3)}$, which is flattened into h_{pre} and mapped to h_{fc}, read by the policy and value heads. Lower panel: block 3, ReLU, flatten, linear layer and ReLU, with pre-ReLU and post-ReLU quantities marked.

A.1 Network architecture

All agents share the network in Figure 7, and most of our analyses read it at two sites, h_{pre} and h_{fc}. The input is a 64×64 RGB frame. Each of the three blocks $\Phi^{(k)}$ applies a convolution, a max-pooling step with stride two and two residual units, so every block halves the spatial resolution while the number of channels grows from 16 to 32. This gives

$$H^{(1)}(s) \in \mathbb{R}^{16\times32\times32}, \qquad H^{(2)}(s) \in \mathbb{R}^{32\times16\times16}, \qquad H^{(3)}(s) \in \mathbb{R}^{32\times8\times8}.$$

Because convolutions and pooling act locally, each channel vector $H^{(k)}_{:xy}(s)$ still belongs to a region of the frame, so the trunk keeps the spatial layout of the scene up to its last block.

The lower panel of Figure 7 enlarges the transition from this spatial trunk to the vector representation. The output of block 3 is rectified and flattened into $h_{\text{pre}}(s) \in \mathbb{R}^{2048}$, with $2048 = 32 \cdot 8 \cdot 8$. A single linear layer followed by a ReLU then gives $h_{\text{fc}}(s) \in \mathbb{R}^{256}$, so $d_{\text{pre}} = 2048$ and $d_{\text{fc}} = 256$. Both sites are therefore taken after a rectification, and the only pre-activation quantity between them is $W_{\text{fc}}h_{\text{pre}}(s) + b_{\text{fc}}$. The two heads read h_{fc} linearly. The policy head produces logits $\eta(s) = W_\pi h_{\text{fc}}(s) + b_\pi$ over the $|\mathcal{A}| = 15$ actions, with $\pi(\cdot \mid s) = \operatorname{softmax}\eta(s)$, and the value head is a single linear unit $V(s) = w_V^\top h_{\text{fc}}(s) + b_V$.

The two sites sit at opposite ends of this transition. h_{pre} is the last representation with spatial structure. Any information the policy uses about where the cheese, the coin or the candidate objects are has to pass through it, and it can still be read cell by cell through $H^{(3)}_{:xy}(s)$. h_{fc} has no spatial layout, but it is exactly what the heads read. As the policy head is linear, a displacement of h_{fc} reaches the action distribution only through W_π, and the activation-space Fisher at this site takes the closed form given after Proposition 2. So comparing the two sites separates information that the trunk still carries in spatial form from information that has been passed on to the readout. Training and implementation details are given in Appendix H.

A.2 Theory: Local Policy Geometry

Probing establishes whether information about a goal-related variable is linearly available in a representation. It does not establish whether the trained policy is sensitive to changes in that represen-

tation (Elazar et al., 2021). We therefore need a quantity that connects an internal representation change to the action distribution produced downstream.

The construction follows the standard information-geometric interpretation of Fisher information (Amari, 1998; Martens, 2020). Fisher information describes the local second-order geometry of a family of probability distributions. Natural-gradient methods usually apply this geometry with respect to model parameters (Kakade, 2001). Here, the trained network is held fixed and we express the same policy geometry in coordinates of an internal representation.

Definition 3 (Downstream policy)**.** *For representation site m, let*

$$\pi^{(m)}(\cdot \mid h)$$

denote the action distribution obtained by applying the remainder of the trained network downstream of site m to representation h. For every policy input s,

$$\pi(\cdot \mid s) = \pi^{(m)}\Big(\cdot \mid h^{(m)}(s)\Big).$$

The downstream policy does not define a new policy or a new state space. It simply factors the existing computation at an intermediate representation site. The upstream network maps the input s to $h^{(m)}(s)$, and $\pi^{(m)}$ denotes the fixed remainder of the trained network.

This factorization lets us express the response to a world edit either in input space or in representation space. For a base state s and environment edit e, let s_e denote the matched edited state and

$$\Delta_e^{(m)}(s) = h^{(m)}(s_e) - h^{(m)}(s)$$

the corresponding representation displacement from Definition 2. Writing

$$h = h^{(m)}(s),$$

we have

$$\pi(\cdot \mid s) = \pi^{(m)}(\cdot \mid h), \qquad \pi(\cdot \mid s_e) = \pi^{(m)}\left(\cdot \mid h + \Delta_e^{(m)}(s)\right).$$

Therefore,

$$D_{\mathrm{KL}}\left(\pi(\cdot \mid s) \,\|\, \pi(\cdot \mid s_e)\right) = D_{\mathrm{KL}}\left(\pi^{(m)}(\cdot \mid h) \,\middle\|\, \pi^{(m)}\left(\cdot \mid h + \Delta_e^{(m)}(s)\right)\right).$$

The two expressions describe the same finite policy change. The first compares the policy at the original and edited inputs, while the second exposes the internal representation displacement that connects those endpoints. Fisher geometry becomes useful when we ask how the downstream policy responds locally to that displacement.

Definition 4 (Activation-space Fisher information)**.** *For a fixed representation*

$$h = h^{(m)}(s) \in \mathbb{R}^{d_m},$$

the activation-space Fisher information of the downstream policy is

$$F^{(m)}(s) = \mathbb{E}_{a \sim \pi^{(m)}(\cdot \mid h)}\left[\nabla_h \log \pi^{(m)}(a \mid h)\, \nabla_h \log \pi^{(m)}(a \mid h)^\top\right] \in \mathbb{R}^{d_m \times d_m}.$$

When the representation site and base state are clear from context, we write this matrix as F_h.

The matrix F_h is positive semidefinite. Importantly, it is not a property of the representation alone. It describes the representation through the particular downstream policy that reads it. Directions with large Fisher sensitivity are directions along which small representation changes strongly alter the policy distribution, while directions with small Fisher sensitivity are comparatively ignored downstream.

The connection to policy change follows from the local expansion of the KL divergence.

Proposition 1 (Local policy geometry)**.** *For a sufficiently small displacement*

$$\delta \in \mathbb{R}^{d_m},$$

$$D_{\mathrm{KL}}\left(\pi^{(m)}(\cdot \mid h) \,\middle\|\, \pi^{(m)}(\cdot \mid h + \delta)\right) = \frac{1}{2}\delta^\top F_h \delta + O(\|\delta\|_2^3).$$

Proof. Abbreviate

$$\pi_h = \pi^{(m)}(\cdot \mid h).$$

Then

$$D_{\mathrm{KL}}\left(\pi_h \parallel \pi_{h+\delta}\right) = \mathbb{E}_{a\sim\pi_h}\left[\log \pi_h(a) - \log \pi_{h+\delta}(a)\right].$$

Expanding $\log \pi_{h+\delta}(a)$ to second order around h gives

$$\log \pi_{h+\delta}(a) = \log \pi_h(a) + \delta^\top \nabla_h \log \pi_h(a) + \frac{1}{2}\delta^\top \nabla_h^2 \log \pi_h(a)\delta + O(\|\delta\|_2^3).$$

Substituting into the KL divergence yields

$$D_{\mathrm{KL}} = -\delta^\top \mathbb{E}_{a\sim\pi_h}\left[\nabla_h \log \pi_h(a)\right] - \frac{1}{2}\delta^\top \mathbb{E}_{a\sim\pi_h}\left[\nabla_h^2 \log \pi_h(a)\right]\delta + O(\|\delta\|_2^3).$$

The first-order term vanishes because the score has zero expectation,

$$\mathbb{E}_{a\sim\pi_h}\left[\nabla_h \log \pi_h(a)\right] = 0.$$

Under the standard regularity conditions, the information identity gives

$$-\mathbb{E}_{a\sim\pi_h}\left[\nabla_h^2 \log \pi_h(a)\right] = \mathbb{E}_{a\sim\pi_h}\left[\nabla_h \log \pi_h(a) \nabla_h \log \pi_h(a)^\top\right] = F_h.$$

Therefore,

$$D_{\mathrm{KL}}\left(\pi_h \parallel \pi_{h+\delta}\right) = \frac{1}{2}\delta^\top F_h \delta + O(\|\delta\|_2^3).$$

□

The absence of a first-order term is important. The KL divergence is locally minimized when the two policy distributions coincide, so its leading nonzero variation is quadratic. The Fisher quadratic

$$\delta^\top F_h \delta$$

therefore measures the local policy sensitivity of a representation displacement.

Applied to the world-edit displacement

$$\delta = \Delta_e^{(m)}(s),$$

this gives

$$D_{\mathrm{KL}}\left(\pi^{(m)}(\cdot \mid h^{(m)}(s)) \,\middle\|\, \pi^{(m)}\left(\cdot \mid h^{(m)}(s) + \Delta_e^{(m)}(s)\right)\right) \approx \frac{1}{2}\Delta_e^{(m)}(s)^\top F^{(m)}(s)\Delta_e^{(m)}(s).$$

This is the quantity used in the world-edit Fisher analysis. The environment determines the displacement through a controlled edit, while the Fisher geometry determines how strongly the downstream policy responds to that displacement around the original state.

Definition 5 (World-edit Fisher sensitivity)**.** *For a base state s, controlled environment edit e, and representation site m, let*

$$\Delta_e^{(m)}(s) = h^{(m)}(s_e) - h^{(m)}(s)$$

be the world-edit representation displacement from Definition 2. We define the world-edit Fisher sensitivity

$$Q_e^{(m)}(s) = \Delta_e^{(m)}(s)^\top F^{(m)}(s)\Delta_e^{(m)}(s).$$

When the edit changes the intended-goal variable, we write

$$e = \star, \qquad s_\star, \qquad \Delta_\star^{(m)}(s).$$

When a corresponding proxy edit is well defined, we analogously write

$$e = p, \qquad s_p, \qquad \Delta_p^{(m)}(s).$$

Not every environment admits both kinds of edit.

By Proposition 1,

$$\frac{1}{2}Q_e^{(m)}(s)$$

is the second-order local approximation to the change in the downstream policy induced by moving from the base representation toward its world-edited counterpart:

$$D_{\mathrm{KL}}\left(\pi^{(m)}(\cdot \mid h^{(m)}(s)) \,\middle\|\, \pi^{(m)}\left(\cdot \mid h^{(m)}(s) + \Delta_e^{(m)}(s)\right)\right) \approx \frac{1}{2}Q_e^{(m)}(s).$$

Because

$$h^{(m)}(s) + \Delta_e^{(m)}(s) = h^{(m)}(s_e),$$

the finite endpoints are the same as those compared by the behavioral world-edit KL,

$$D_{\mathrm{KL}}\left(\pi(\cdot \mid s) \parallel \pi(\cdot \mid s_e)\right).$$

The two quantities differ in what they expose. The behavioral KL measures the complete policy change between the original and edited inputs. $Q_e^{(m)}(s)$ instead expresses the corresponding change locally through representation site m, and therefore measures how strongly the downstream policy is coupled to the representation displacement produced by edit e. A large or linearly informative displacement can consequently have small Fisher sensitivity if the trained downstream computation is comparatively insensitive to its direction.

Proposition 2 (Dependence on the trained downstream computation)**.** *Let*

$$\eta^{(m)}(h) \in \mathbb{R}^{|\mathcal{A}|}$$

denote the logits produced by the downstream network from representation h, and let

$$J_\eta^{(m)}(h) = \frac{\partial \eta^{(m)}(h)}{\partial h}$$

denote its Jacobian. Then the activation-space Fisher can be written as

$$F_h = J_\eta^{(m)}(h)^\top \left[\mathrm{diag}(\pi) - \pi\pi^\top\right] J_\eta^{(m)}(h),$$

where $\pi = \pi^{(m)}(\cdot \mid h)$.

Proof. For a softmax distribution, the Fisher information with respect to the logits is

$$F_\eta = \mathrm{diag}(\pi) - \pi\pi^\top.$$

Applying the chain rule to $\nabla_h \log \pi^{(m)}(a \mid h)$ pulls this metric back through the downstream logit map, giving

$$F_h = J_\eta^\top F_\eta J_\eta.$$

□

At the final fully connected representation, the downstream logit map is linear,

$$\eta^{(\mathrm{fc})}(h) = W_\pi h + b_\pi,$$

so

$$F^{(\mathrm{fc})}(s) = W_\pi^\top \left[\mathrm{diag}(\pi) - \pi\pi^\top\right] W_\pi.$$

At earlier representation sites, such as h_{pre}, the Jacobian additionally contains the downstream transformations between that site and the policy head.

Proposition 2 makes the interpretation of Q precise. Fisher sensitivity is a property of the representation *as read by the trained downstream policy*. Two agents can make the same information available in their internal representations while inducing very different Fisher geometries because their downstream computations use that information differently. A small $Q_e^{(m)}$ therefore does not imply that information associated with edit e is absent from the representation.

Separating displacement from directional sensitivity. The quadratic form Q combines two effects. A world edit can produce a large policy response because it moves the representation far, because the downstream policy is unusually sensitive to the direction of that movement, or both. We separate these effects relative to the policy's average local Fisher sensitivity.

Definition 6 (Fisher sensitivity decomposition)**.** *For a nonzero displacement $\Delta \in \mathbb{R}^d$, define its directional Fisher sensitivity*

$$S(\Delta) = \frac{\Delta^\top F_h \Delta}{\|\Delta\|_2^2} = \widehat{\Delta}^\top F_h \widehat{\Delta}, \qquad \widehat{\Delta} = \frac{\Delta}{\|\Delta\|_2}.$$

For an isotropically sampled unit vector $u \sim \mathrm{Unif}(\mathbb{S}^{d-1})$,

$$\mathbb{E}_u \left[u^\top F_h u\right] = \frac{\mathrm{tr}(F_h)}{d}.$$

We therefore define the relative directional gain

$$G(\Delta) = \frac{S(\Delta)}{\mathrm{tr}(F_h)/d}$$

The relative gain G is defined whenever $\mathrm{tr}(F_h) > 0$. If the local Fisher vanishes identically, then $Q = 0$ for every direction.

We also define the displacement in Fisher units

$$D_F(\Delta) = \|\Delta\|_2^2 \frac{\mathrm{tr}(F_h)}{d}.$$

The two quantities factor the Fisher sensitivity exactly:

$$Q = \underbrace{G}_{\text{relative directional gain}} \underbrace{D_F}_{\text{displacement in Fisher units}}.$$

D_F measures how far the counterfactual moves the representation in units of the policy's average local sensitivity. G measures whether the direction of that movement is more or less policy-sensitive than an average direction in the same representation space. Thus

$$G > 1$$

means that the counterfactual moves along a direction that is privileged by the downstream policy relative to an isotropic direction. In the experiments, we additionally compare this directional gain with matched-norm random directions to account for the non-isotropic spectrum of F_h.

Proposition 3 (Rescaling invariance)**.** *Consider a scalar reparameterization*

$$h' = ch$$

together with a compensating downstream transformation that leaves the represented policy unchanged. Then

$$Q' = Q, \qquad G' = G, \qquad D'_F = D_F,$$

whereas $\|\Delta\|_2$ and S individually are not invariant.

Proof. Under the reparameterization,

$$\Delta' = c\Delta.$$

Because the downstream computation is compensated to preserve the policy, gradients with respect to the new representation scale as c^{-1}, giving

$$F_{h'} = c^{-2} F_h.$$

Therefore,

$$Q' = (c\Delta)^\top (c^{-2} F_h)(c\Delta) = Q.$$

Furthermore,

$$S' = c^{-2} S, \qquad \frac{\mathrm{tr}(F_{h'})}{d} = c^{-2} \frac{\mathrm{tr}(F_h)}{d},$$

so

$$G' = G.$$

Finally,

$$D'_F = \|c\Delta\|_2^2 \frac{\mathrm{tr}(c^{-2}F_h)}{d} = D_F.$$

□

This invariance matters when comparing different trained networks. The scale of an internal representation is not itself identifiable, so neither the raw displacement norm nor its unnormalized directional sensitivity should be interpreted across agents in isolation. Q, G, and D_F remain unchanged under this elementary rescaling.

Corollary 1 (Dependence on policy sharpness)**.** *As the downstream action distribution approaches a point mass,*

$$\pi \to e_a,$$

we have

$$\mathrm{diag}(\pi) - \pi\pi^\top \to 0.$$

Consequently,

$$F_h \to 0$$

and

$$Q(\Delta) \to 0$$

for every fixed displacement Δ*.*

Thus a more deterministic policy can exhibit smaller Fisher sensitivity along every representation direction, independently of the particular feature being studied. We therefore report the average Fisher sensitivity

$$\frac{\mathrm{tr}(F_h)}{d}$$

per cohort next to D_F (Table 5) when comparing policies with different levels of sharpness.

Definition 7 (Fisher cosine)**.** *For two directions* Δ_1 *and* Δ_2 *satisfying*

$$\Delta_i^\top F_h \Delta_i > 0, \qquad i \in \{1, 2\}.$$

evaluated at the same base representation h *under the same local Fisher metric* F_h*, we define*

$$\cos_F(\Delta_1, \Delta_2) = \frac{\Delta_1^\top F_h \Delta_2}{\sqrt{(\Delta_1^\top F_h \Delta_1)(\Delta_2^\top F_h \Delta_2)}}.$$

The Fisher cosine measures alignment specifically in directions to which the downstream policy is locally sensitive. Values near $+1$ indicate aligned policy-relevant effects, values near 0 indicate local Fisher orthogonality, and values near -1 indicate opposing policy-relevant effects. We use this quantity later to characterize the geometry of cue competition in ATTRIBUTE MAZE.

Low-rank Fisher audit. Because the activation-space Fisher matrix has rank at most $|\mathcal{A}|-1 = 14$ in this environment, we additionally test whether low Fisher rank alone can induce extreme pairwise cosine values. Matched random 256-dimensional vectors evaluated under rank-14 random Fisher metrics produce essentially no pairs with $|\cos_F| > 0.9$, whereas 82–86% of real state pairs exceed this threshold across the three GM policies on the black background. The near-± 1 Fisher structure therefore cannot be explained by Fisher rank alone.

Definition 8 (Signed target-selection derivative)**.** *Let*

$$A_\star(s), A_p(s) \subseteq \mathcal{A}$$

denote the intended-goal- and proxy-consistent action sets for state s*. In* MAZE *and* ATTRIBUTE MAZE*, both sets are taken from the base state:* $A_\star(s)$ *is the first step toward the base cheese (respectively the line candidate) and* $A_p(s)$ *the first step toward the top-right corner (respectively the yellow candidate). In* COIN*, the base state holds the coin at a cell right of the agent, on the shortcut side, where the two sets would coincide, and the edit moves it to the cell one step left of the*

agent. So $A_\star(s) = A_{\text{left}}$, the leftward actions toward the relocated coin, and $A_p(s) = A_{\text{right}}$, the rightward shortcut. At representation site m, define the target-selection contrast

$$C_s^{(m)}(h) = \sum_{a \in A_\star(s)} \pi^{(m)}(a \mid h) - \sum_{a \in A_p(s)} \pi^{(m)}(a \mid h).$$

For world edit e with displacement $\Delta_e^{(m)}(s)$, the signed target-selection derivative is

$$\partial_{\Delta_e} C_s^{(m)} = \nabla_h C_s^{(m)} \left(h^{(m)}(s)\right)^\top \Delta_e^{(m)}(s).$$

The action sets are determined from the environment state s and are held fixed when differentiating with respect to h. For a nonzero displacement, we additionally define the displacement-normalized derivative

$$\widetilde{\partial_{\Delta_e} C_s^{(m)}} = \frac{\partial_{\Delta_e} C_s^{(m)}}{\|\Delta_e^{(m)}(s)\|_2}.$$

The raw derivative measures the first-order target-selective effect of the actual representation change induced by edit e. Its sign indicates which side of the intended-versus-proxy contrast is favored, while its magnitude combines direction and edit size. The normalized derivative removes the Euclidean magnitude of the displacement and therefore measures target-selective sensitivity per unit representation change.

Unlike the Fisher quadratic Q, the signed derivative is first-order and directional. Q asks how strongly the downstream action distribution changes along a world-induced representation direction. $\partial_{\Delta_e} C_s^{(m)}$ asks whether that same direction moves the behaviorally relevant target-selection contrast toward the intended or proxy-consistent side. So, the two measurements are complementary: Q measures whether the policy is sensitive to the world-induced displacement at all, while the signed derivative asks whether that sensitivity is organized along the behaviorally relevant target-selection contrast.

A.3 Projections and Vector Analysis

The constructions below define projections relative to a chosen reference direction $u^{(m)}$. They do not assume that one fixed reference direction provides a global description of the representation geometry across states.

Euclidean projection. Let $\Delta^{(m)}$ denote a representation displacement at site m, and let $u^{(m)} \neq 0$ be a reference direction in the same representation space. We write

$$\hat{u}^{(m)} = \frac{u^{(m)}}{\|u^{(m)}\|_2}.$$

The Euclidean projection of $\Delta^{(m)}$ onto the span of $u^{(m)}$ is

$$\Delta_{u,E}^{(m)} = \left(\hat{u}^{(m)\top} \Delta^{(m)}\right) \hat{u}^{(m)},$$

with residual

$$\Delta_{\perp,E}^{(m)} = \Delta^{(m)} - \Delta_{u,E}^{(m)}.$$

By construction,

$$\Delta_{u,E}^{(m)\top} \Delta_{\perp,E}^{(m)} = 0.$$

This decomposition measures how the representation displacement is oriented in the ordinary Euclidean geometry of activation space.

Although the Euclidean components are orthogonal in activation space, they are not generally orthogonal under the Fisher metric. Consequently,

$$Q^{(m)}\left(\Delta^{(m)}\right) = Q^{(m)}\left(\Delta_{u,E}^{(m)}\right) + Q^{(m)}\left(\Delta_{\perp,E}^{(m)}\right) + 2\Delta_{u,E}^{(m)\top} F^{(m)}(s) \Delta_{\perp,E}^{(m)}.$$

The cross term need not vanish because Euclidean orthogonality does not imply Fisher orthogonality.

Fisher-orthogonal projection. To obtain an additive decomposition of the Fisher quadratic, we instead define the projection using the local Fisher geometry. The downstream policy induces this local geometry through the activation-space Fisher matrix $F^{(m)}(s)$. When

$$u^{(m)\top} F^{(m)}(s) u^{(m)} > 0,$$

we define the Fisher-orthogonal projection of $\Delta^{(m)}$ onto the span of $u^{(m)}$ as

$$\Delta^{(m)}_{u,F} = \frac{u^{(m)\top} F^{(m)}(s) \Delta^{(m)}}{u^{(m)\top} F^{(m)}(s) u^{(m)}} u^{(m)},$$

with residual

$$\Delta^{(m)}_{\perp,F} = \Delta^{(m)} - \Delta^{(m)}_{u,F}.$$

The residual is orthogonal to the reference direction under the local Fisher metric,

$$u^{(m)\top} F^{(m)}(s) \Delta^{(m)}_{\perp,F} = 0.$$

Consequently, the Fisher quadratic decomposes exactly,

$$Q^{(m)}\left(\Delta^{(m)}\right) = Q^{(m)}\left(\Delta^{(m)}_{u,F}\right) + Q^{(m)}\left(\Delta^{(m)}_{\perp,F}\right),$$

where, for any representation-space direction v,

$$Q^{(m)}(v) = v^\top F^{(m)}(s) v.$$

The Euclidean and Fisher projections answer different questions. The Euclidean projection asks how much of the displacement lies along the reference direction in activation space. The Fisher projection instead asks how the policy-sensitive component of that displacement is distributed relative to the same direction.

Signed-effect decomposition. At a fixed base state, the local signed target-selection derivative is linear in the representation displacement. For the target-selection contrast $C^{(m)}_s(h)$,

$$\partial_\Delta C^{(m)}_s = \nabla_h C^{(m)}_s \left(h^{(m)}(s)\right)^\top \Delta^{(m)}.$$

Therefore, for any additive decomposition

$$\Delta^{(m)} = \Delta^{(m)}_1 + \Delta^{(m)}_2,$$

we have

$$\partial_\Delta C^{(m)}_s = \partial_{\Delta_1} C^{(m)}_s + \partial_{\Delta_2} C^{(m)}_s.$$

This exact additivity applies to the local directional derivative. It does not imply that the finite change

$$C^{(m)}_s\left(h^{(m)}(s) + \Delta^{(m)}\right) - C^{(m)}_s\left(h^{(m)}(s)\right)$$

decomposes additively for arbitrary finite displacements. Unlike the quadratic Fisher decomposition, the derivative decomposition follows directly from linearity and therefore does not depend on how the additive split was constructed.

Local and shared-axis descriptions. The projection constructions above can be used to compare a fully state-specific representation direction with a lower-dimensional shared description.

Let $d(s)$ denote a state-dependent representation direction of interest. A fully local description retains this vector separately for every state. A shared-axis description instead asks whether the state dependence can be compressed to a fixed reference direction u together with state-dependent coefficients.

The simplest such approximation is

$$d(s) \approx \alpha(s) u,$$

where $\alpha(s)$ is a scalar that may vary across states. If the orientation of the shared direction is itself determined by a discrete state variable, we write

$$d(s) \approx \sigma(s)\beta(s)u, \qquad \sigma(s) \in \{-1, +1\}, \qquad \beta(s) \geq 0.$$

Here, $\sigma(s)$ determines which orientation of the shared axis is active, while $\beta(s)$ controls the local magnitude.

This form does not imply that the full representation geometry is one-dimensional. It only states that the component relevant to the particular displacement family under study may be well described by one shared axis after accounting for state-dependent sign and magnitude. The residual

$$r(s) = d(s) - \sigma(s)\beta(s)u$$

contains the remaining state-specific structure.

Shared-axis coefficients under Euclidean and Fisher geometry. For a fixed shared direction u, the state-dependent coefficient can be defined under different geometries.

In Euclidean activation space, the signed coordinate of $d(s)$ along the unit direction

$$\hat{u} = \frac{u}{\|u\|_2}$$

is

$$\alpha_E(s) = \hat{u}^\top d(s).$$

The corresponding Euclidean projection is

$$d_{u,E}(s) = \alpha_E(s)\hat{u}.$$

Under the local Fisher geometry, the corresponding coefficient is

$$\alpha_F(s) = \frac{u^\top F(s) d(s)}{u^\top F(s) u},$$

giving the Fisher-orthogonal projection

$$d_{u,F}(s) = \alpha_F(s)u.$$

The two coefficients answer different questions. $\alpha_E(s)$ measures how much of the displacement lies along u in ordinary activation space. $\alpha_F(s)$ measures the coefficient of the same shared direction after orthogonality is defined by local policy sensitivity.

The resulting residual

$$d_{\perp,F}(s) = d(s) - d_{u,F}(s)$$

satisfies

$$u^\top F(s) d_{\perp,F}(s) = 0,$$

and therefore

$$Q(d(s)) = Q(d_{u,F}(s)) + Q(d_{\perp,F}(s)).$$

Interventions derived from these decompositions. The same distinction gives several intervention families.

A state-local intervention retains the complete direction,

$$\delta_{\text{local}}(s; \beta) = \beta\, d(s).$$

A shared-axis intervention retains only a fixed direction and a state-dependent sign and magnitude,

$$\delta_{\text{shared}}(s; \beta) = \beta\, \sigma(s)\, m(s)\, \hat{u},$$

where $m(s) \geq 0$ specifies the chosen local magnitude.

Removing the sign dependence gives the corresponding unsigned intervention,

$$\delta_{\text{unsigned}}(s; \beta) = \beta\, m(s)\, \hat{u}.$$

Finally, using the Fisher projection gives

$$\delta_F(s;\beta) = \beta\,\alpha_F(s)\,u.$$

These interventions separate different compression hypotheses. The local intervention retains the full state-dependent representation geometry. The shared-axis intervention then asks whether its causal effect can be reproduced using one common direction together with state-dependent gating. The unsigned intervention tests whether that gating is necessary. Finally, the Fisher-projected intervention retains only the component assigned to the shared direction by the local policy-sensitive geometry.

A.4 Environment-specific target geometry

On the training distribution the intended goal and the proxy are confounded. At most states one and the same action moves the agent toward both, so the policy's choice cannot tell us which of the two it is following. Behavior separates goal from proxy only at states where the two require different first actions, and finding these states requires knowing exactly which actions make progress toward each target. Walls make the straight-line bearing of a target a poor guide to this, so we define progress on the level's grid graph. The resulting graph-exact action sets are used throughout the paper, both to choose evaluation states (Section E) and to score behavior on them.

This needs a local notion of progress toward a spatial target. The MDP state $s \in \mathcal{S}$ describes the complete environment configuration, while navigation depends on the agent's location within the spatial structure of the current level. We therefore introduce a spatial projection of the environment state.

Spatial state projection. For a fixed level ξ, let

$$\mathcal{G}_\xi = (V_\xi, E_\xi)$$

denote its traversable grid graph. Each vertex $v \in V_\xi$ corresponds to a traversable cell, and an edge $(u, v) \in E_\xi$ indicates that the agent can move between the corresponding cells in one environment step.

We write

$$x_\xi : \mathcal{S} \to V_\xi$$

for the spatial projection that maps a full environment state s to the cell occupied by the agent. Thus s retains the complete MDP state, while $x_\xi(s)$ extracts only the agent's spatial location on level ξ.

For cells $u, v \in V_\xi$, we define the graph distance

$$d_\xi(u,v) = \min\left\{k \in \mathbb{N}_0 : \exists\, v_0,\dots,v_k \in V_\xi,\ v_0 = u,\ v_k = v,\ (v_{i-1}, v_i) \in E_\xi\ \forall i = 1,\dots,k\right\}.$$

Hence $d_\xi(u,v)$ is the minimum number of valid grid transitions required to move from u to v. We compute this distance exactly using breadth-first search.

Spatial state-transition projection. The MDP transition kernel acts on complete environment states,

$$s' \sim P(\cdot \mid s, a).$$

For the navigation environments considered here, the one-step spatial transition is deterministic conditional on the full state s and action a. We therefore define

$$x'_\xi(s,a) = x_\xi(s'), \qquad \text{where } P(s' \mid s, a) = 1,$$

as the agent cell after applying action a in state s on level ξ. Thus $x_\xi(s)$ and $x'_\xi(s,a)$ are respectively the spatial projections before and after the MDP transition. This notation introduces no additional transition model beyond P.

Definition 9 (Graph-exact target-progress action set)**.** *For a state s, level ξ, and candidate target cell $v \in V_\xi$, define*

$$A_v(s) = \left\{a \in \mathcal{A} : d_\xi\big(x'_\xi(s,a), v\big) < d_\xi(x_\xi(s), v)\right\} \subseteq \mathcal{A}.$$

$A_v(s)$ is therefore the subset of actions whose induced spatial transition strictly decreases the remaining shortest-path distance to v.

Because the grid edges have unit cost, an action belongs to $A_v(s)$ exactly when its induced movement lies on a shortest path from the current agent cell toward v. An action from which v remains eventually reachable need not belong to $A_v(s)$ if it does not decrease the remaining shortest-path distance on that step.

When multiple raw policy actions induce the same effective movement, we aggregate their probability mass rather than treating action aliases as distinct navigation choices.

Two candidate cells $v_1, v_2 \in V_\xi$ are locally action-discriminating at state s when

$$A_{v_1}(s) \neq A_{v_2}(s).$$

At such states, the two spatial target hypotheses make different predictions about immediate control and can therefore be distinguished from the native policy's action distribution.

Environment-specific target action sets. In MAZE, let $v_\star$ denote the cheese cell and v_p the proxy-associated top-right destination. We write

$$A_\star(s) \equiv A_{v_\star}(s), \qquad A_p(s) \equiv A_{v_p}(s).$$

Thus $A_\star(s)$ contains graph-exact progress actions toward the intended goal location, while $A_p(s)$ contains progress actions toward the proxy-associated destination.

In ATTRIBUTE MAZE, let v_{line} and v_{yellow} denote the locations of the line-bearing and yellow-bearing objects. Depending on the world edit under consideration, these locations instantiate the intended and proxy candidates. We again use

$$A_\star(s) \qquad \text{and} \qquad A_p(s)$$

as shorthand for the corresponding graph-exact action sets, while retaining $A_{v_{\mathsf{line}}}(s)$ and $A_{v_{\mathsf{yellow}}}(s)$ when the physical cue identity matters.

COIN does not use the grid-graph construction for its proxy. The learned shortcut is represented by the right-moving action family, and in every state of the COIN conflict bank the coin sits one cell to the left of the agent in the state on which target selection is read (Appendix E). Progress toward the coin is therefore leftward motion, and we use the two horizontal movement families

$$A_{\text{left}},\ A_{\text{right}} \subseteq \mathcal{A}_{\text{Coin}},$$

each of which holds the three actions that move the agent in that direction. For notational consistency in comparisons across environments, we write

$$A_\star(s) \equiv A_{\text{left}}, \qquad A_p(s) \equiv A_{\text{right}}.$$

A state is locally action-discriminating whenever

$$A_\star(s) \neq A_p(s).$$

Action-discriminating evaluation states. For a level ξ, let $\mathcal{S}_\xi \subseteq \mathcal{S}$ denote the environment states belonging to that level. For two candidate action sets $A_1(s)$ and $A_2(s)$, we call a state action-discriminating when

$$A_1(s) \neq A_2(s).$$

For the intended and proxy-associated targets, we therefore define

$$\mathcal{S}_{\text{conf}}(\xi) = \{s \in \mathcal{S}_\xi : A_\star(s) \neq A_p(s)\}.$$

These states are decision points at which progress toward the intended and proxy-associated targets implies different immediate action sets. Restricting evaluation to $\mathcal{S}_{\text{conf}}(\xi)$ therefore lets the observed policy action distinguish which target better organizes local behavior. We call the members of $\mathcal{S}_{\text{conf}}(\xi)$ *conflict states*.

Strict conflict. A conflict state can still have progress sets that overlap, and an action in the overlap serves both targets and so reveals nothing about which one the policy follows. We therefore build every evaluation bank from states in *strict conflict*,

$$A_\star(s) \cap A_p(s) = \emptyset,$$

at which every progress action serves exactly one of the two targets. In MAZE shortest paths are unique, so both sets contain a single action and every conflict state is a strict conflict. In COIN the left-moving and right-moving families are disjoint by construction. In ATTRIBUTE MAZE we additionally require the two candidates to be equidistant from the agent, so that neither target is favoured by being closer.

B EXPERIMENTAL DESIGN AND EXTENDED RESULTS

B.1 IS INFORMATION ABOUT THE INTENDED GOAL STILL THERE?

Motivation. Goal misgeneralization tells us that an agent behaves according to something other than the intended goal, but this does not tell us what happened to information about the intended goal inside the agent. One simple possibility is that the relevant goal information is no longer available in the learned representation. We therefore begin with the most basic internal question: *is information about the intended goal still there?*

We test this first through linear decodability. A successful probe gives an existence result: the corresponding information is available in a form accessible to a linear readout. We then ask where this information appears across the network, whether its accessibility differs between GM and Non-GM agents, and how much of that accessibility can be attributed to training rather than to structure already exposed by the input or architecture.

Design. For each environment, we define goal-related and, where meaningful, proxy-related target variables and fit regularized linear probes (Alain & Bengio, 2017) to internal activations. The probe targets are environment-specific because the information relevant to the goal differs across tasks. For a binary target y, the probe at representation m has score

$$f_m(s) = w_m^\top h^{(m)}(s) + b_m,$$

and we evaluate f_m using held-out ROC–AUC. For continuous spatial targets, we fit the corresponding ridge regressor

$$\hat{y}_m(s) = w_m^\top h^{(m)}(s) + b_m$$

and evaluate it using held-out R^2. Table 19 lists every probe with its label, bank, split and sites. In MAZE and COIN no level contributes observations to both probe fitting and evaluation, and in ATTRIBUTE MAZE the training and held-out configurations use disjoint level ranges. All probe preprocessing is fitted on the training partition only.

We probe the same computational sites across environments. Following the network architecture in Section A.1, we define

$$h^{(1)} = H^{(1)}, \qquad h^{(2)} = H^{(2)}, \qquad h^{(3)} = H^{(3)},$$

where $H^{(1)}, H^{(2)}, H^{(3)}$ denote the corresponding residual-block representations. No additional ReLU is inserted after the first two blocks because no such operation occurs in the policy forward pass. The subsequent shared representations are

$$h_{\text{pre}} = \text{vec}\left(\text{ReLU}(h^{(3)})\right)$$

and

$$h_{\text{fc}} = \text{ReLU}\left(W_{\text{fc}} h_{\text{pre}} + b_{\text{fc}}\right).$$

Thus h_{pre} is the actual input to the fully connected layer and h_{fc} is the final shared representation consumed by the policy and value readouts. For $h^{(1)}, h^{(2)}, h^{(3)}$, we globally average pool the convolutional feature maps before fitting the probe. The later representations are used directly. We exclude policy logits and value predictions from this sequence because they are separate readout branches rather than later shared representations.

Our primary quantity is absolute held-out probe performance. This answers the direct representational question: how linearly accessible is the target at each site? The layerwise figures therefore report held-out AUC or R^2 directly for GM and Non-GM agents.

Absolute decodability alone, however, does not establish how much accessibility was produced by training. Simple visual or spatial structure can already be linearly exposed by an untrained convolutional architecture (Hermann & Lampinen, 2020). We therefore evaluate randomly initialized networks of the same architecture on the same probe tasks, in the spirit of control tasks (Hewitt & Liang, 2019). For reference, we define the training-induced change

$$\Delta\text{AUC}_m = \text{AUC}^{(m)}_{\text{trained}} - \text{AUC}^{(m)}_{\text{init}}$$

for binary probes and

$$\Delta R^2_m = R^{2,(m)}_{\text{trained}} - R^{2,(m)}_{\text{init}}$$

for continuous probes. These differences are secondary comparisons rather than the primary measure of decodability. In particular, a small Δ can coexist with high absolute decodability when the initialized architecture already exposes the target, while a large ΔR^2 can be misleading when the initialized regression probe has negative held-out R^2. We therefore report absolute performance together with the corresponding initialization controls rather than interpreting Δ in isolation.

Where necessary, we additionally compare against simple input-level or positional controls. These controls distinguish information represented by the learned network from structure that can already be recovered directly from the observation or from correlated state variables. The relevant controls are environment-specific and are described with each probe analysis.

Finally, we construct empirical permutation nulls using the same grouping and evaluation structure as the corresponding probes. We use these as supplementary checks on the probe procedure rather than as the primary criterion for representational availability. Our central comparisons are therefore absolute held-out accessibility across representation sites, its difference between GM and Non-GM agents, and its relation to matched initialization and task-specific controls.

MAZE probe construction. In MAZE, we first ask whether the cheese object itself is linearly detectable, using the probe bank of Appendix E.3 (Table 19). Standard MAZE episodes do not contain cheese-absent states, so the negative class consists of states from which the cheese has been removed, and the presence probe is a counterfactual probe rather than an ordinary in-distribution task.

Presence alone does not establish that the representation contains the spatial information required to act toward the goal. We therefore additionally probe cheese bearing relative to the agent. Because cheese bearing is correlated with the agent's own position under the MAZE state distribution, we additionally compare against probes using agent position alone and evaluate a control in which corner direction is held fixed.

For the proxy-associated destination, the corner-bearing label uses the reachable top-right extreme of the maze, computed from maze geometry rather than identified with a fixed image corner. Corner bearing is retained only as an auxiliary spatial control because agent position alone predicts it with high accuracy.

For each trained checkpoint, we compare probe performance against randomly initialized networks. Permutation labels are shuffled at the level level. Because simple input variables can themselves make these targets linearly predictable, we additionally use raw-pixel and agent-position baselines to distinguish information present in the observation from features constructed by the learned network.

MAZE results. Figure 8 shows a strong cohort difference in whether the learned representations make cheese information linearly accessible. MAZE shows the clearest difference for cheese presence. The counterfactual presence edit changes only 2–5 pixels within the cheese cell, and those raw cheese-cell pixels linearly identify presence with $\text{AUC} = 0.988$. Thus the relevant visual information is plainly present in the observation.

The learned representations differ sharply. Across GM checkpoints, cheese-presence AUC remains approximately 0.53–0.55 at every probed site, stays within about 0.02 of the randomly initialized networks, and falls below them at h_{pre}. At h_{pre}, mean GM AUC is 0.544, compared with 0.997 for Non-GM agents, and all three GM checkpoints score below all three random initializations. Non-GM agents instead represent cheese presence near ceiling from the second convolutional block onward. The GM result therefore does not reflect absence of the relevant pixels from the input. Rather, GM training does not construct a linearly accessible cheese-presence feature under this counterfactual assay, and accessibility is suppressed relative to initialization at the late shared representation.

Cheese bearing shows a weaker version of the same distinction. Raw cheese-bearing AUC is approximately 0.71–0.81 in GM and 0.87–0.95 in Non-GM agents, but bearing is itself strongly predictable from the agent's position: a probe on agent (x, y) alone reaches 0.830, above GM at every site. When corner direction is held fixed, GM cheese-bearing AUC remains below 0.67 at every site, comparable to the 0.70 obtained from agent position alone, whereas Non-GM agents reach ap-

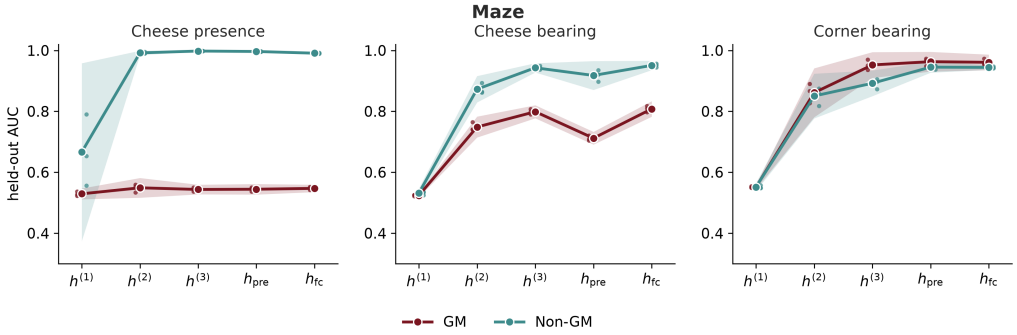

Figure 8: **Layerwise linear availability in MAZE.** Absolute held-out ROC–AUC for counterfactual cheese presence, cheese bearing relative to the agent, and bearing to the reachable top-right corner. Points show the three checkpoint seeds, lines show cohort means, and shaded regions show 95% t-intervals across the three checkpoints. Non-GM agents represent cheese presence near ceiling from $h^{(2)}$ onward, whereas GM agents remain near chance across the network. Cheese-bearing information is also substantially stronger in Non-GM agents. Corner bearing is retained only as an auxiliary spatial control because it is strongly predictable from agent position alone. Initialization, raw-pixel, and agent-position controls are reported separately in Table 2.

proximately 0.94 (Table 2). Thus the GM bearing probe does not provide evidence for substantial cheese-location information beyond the spatial structure already carried by the agent's own position, while Non-GM representations do.

The top-right-corner bearing probe is not interpreted as evidence for a learned proxy representation. Agent position alone predicts this label with $\mathrm{AUC} = 0.959$, showing that much of its apparent decodability follows directly from the geometry of the sampled states. We therefore retain corner bearing only as an auxiliary spatial control.

Together, these results sharpen the MAZE representational result. The cheese is trivially detectable from the observation, but GM training does not construct the strong cheese-presence representation seen in Non-GM agents. Cheese-bearing information in GM is also largely accounted for by agent position, whereas Non-GM agents contain substantially stronger goal-relative spatial information. The MAZE failure is therefore compatible with a learned representation that preserves generic spatial structure while failing to construct the goal-specific features that characterize the Non-GM policy.

COIN probe construction. In COIN, we separate object availability from spatial information about the goal, using the probe bank of Appendix E.3 (Table 19). Unlike cheese presence in MAZE, coin presence varies naturally during ordinary COIN trajectories as the coin moves into and out of the visible frame. Coin bearing is residualized on level progress because horizontal progress is strongly coupled to both observation structure and the training-time rightward trajectory.

For each of five split seeds, levels rather than individual frames are assigned to the probe partitions, giving 52 training levels and 29 held-out levels. This prevents a probe from identifying a level from its background and exploiting that level's typical coin bearing. The test-set coin-presence rate varies across splits from approximately 0.58 to 0.86. We therefore use ROC–AUC for the presence target.

The shared probe bank was generated by GM checkpoint 6033 at 20M and 200M training steps and is used to evaluate every checkpoint. The three GM checkpoints themselves were obtained at 20M, 120M, and 55M training steps. We therefore test the stability of cohort comparisons across the two frame sources and do not interpret a cohort ordering that changes with the evaluated frame population.

For each probe, trained checkpoints are compared with eight randomly initialized networks. For the continuous bearing target, we additionally report absolute R^2 where subtraction of the initialization score would be misleading. In particular, the high-dimensional h_{pre} representation produces negative held-out R^2 in some randomly initialized networks, so a large positive ΔR^2 at that site can exaggerate the meaningful training effect. For the continuous bearing target, the level-permutation

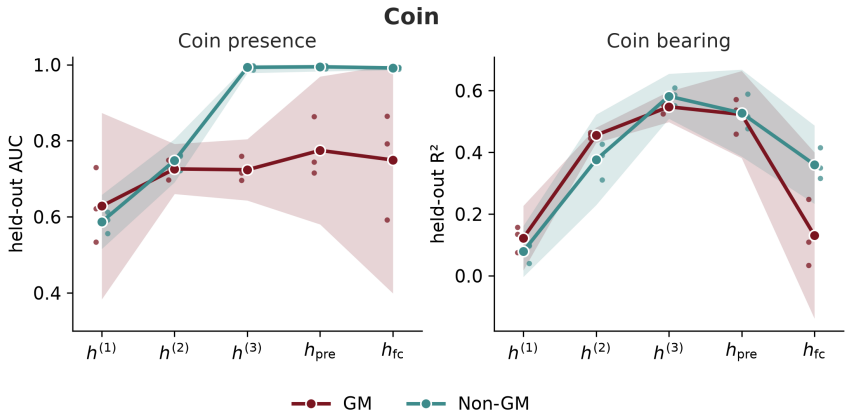

Figure 9: **Layerwise linear availability in COIN.** Absolute held-out probe performance for coin presence and residualized coin bearing under level-grouped train–test splits. Coin presence is evaluated using ROC–AUC and coin bearing using R^2. Points show the three checkpoint seeds, lines show cohort means, and shaded regions show 95% t-intervals across the three checkpoints. Coin presence separates the cohorts from $h^{(3)}$ onward: Non-GM agents approach ceiling, whereas GM agents remain substantially lower. Coin bearing is similarly accessible in both cohorts through h_{pre}. At h_{fc}, the GM–Non-GM bearing ordering depends on the evaluated frame population and is therefore not interpreted. Initialization and input-level controls are reported separately in Table 2.

null yields strongly negative held-out R^2 because permuted labels are constant within levels, so we use $R^2 = 0$ as the chance reference.

COIN results. Figure 9 shows a different pattern across the two goal variables. Coin presence provides the clearest representational cohort difference. At $h^{(2)}$, presence is decodable in both cohorts, with mean AUC 0.726 for GM and 0.748 for Non-GM agents. From the third convolutional representation onward, however, Non-GM agents approach ceiling: mean AUC is 0.993 at $h^{(3)}$, 0.995 at h_{pre}, and 0.991 at h_{fc}. GM agents remain substantially lower, with corresponding means of 0.724, 0.775, and 0.749.

The downstream GM scores also need to be interpreted against the input and initialization controls. At h_{pre} and h_{fc}, the three GM checkpoints do not consistently exceed the best randomly initialized network, and their performance is comparable to a raw-pixel probe with AUC ≈ 0.76. Non-GM agents, by contrast, retain near-ceiling coin-presence information at both sites. This presence separation is stable across the two frame sources and when checkpoint 6033, which generated the evaluation trajectories, is excluded. The robust COIN representational result is therefore not that the coin becomes completely undecodable in GM, but that Non-GM training constructs a much stronger downstream representation of coin presence than GM training does.

Coin bearing behaves differently. Both cohorts contain substantial bearing information through h_{pre}. At $h^{(2)}$, absolute held-out R^2 is 0.456 for GM and 0.376 for Non-GM agents. At $h^{(3)}$, the corresponding values are 0.548 and 0.582, and at h_{pre} they are 0.523 and 0.527. Agent position alone reaches only $R^2 = 0.152$ (Table 2), so coin bearing is not explained by agent position. We report the absolute h_{pre} scores rather than emphasizing ΔR^2, because the randomly initialized networks obtain negative mean held-out R^2 at this high-dimensional site and subtraction would exaggerate the apparent gain.

At h_{fc}, the full probe bank gives mean bearing $R^2 = 0.131$ for GM and 0.360 for Non-GM agents. This ordering is not robust to the frame population: on the 200M rollout subset the direction reverses. We therefore do not interpret the h_{fc} cohort contrast as a GM-specific bearing signature.

COIN consequently provides a heterogeneous result across goal variables. Goal bearing is strongly accessible in both cohorts through h_{pre}, with no stable cohort separation there. Coin presence, however, becomes nearly perfectly accessible in Non-GM agents while remaining substantially weaker in GM agents at the downstream shared representations. The robust representational distinction in

COIN is therefore the organization of goal-object presence rather than a stable difference in goal bearing. Trinh et al. (2024) suggest that a CoinRun agent trained with the coin at the right end does not represent the coin, based on an anomaly detector and a learned help-request policy applied to its penultimate-layer features. Their analysis differs from ours in method and site, since neither tool measures supervised linear decodability, and the penultimate layer corresponds to h_{fc} in our architecture, where GM coin presence is near the raw-pixel level and the GM–Non-GM bearing contrast is not stable across frame sets.

ATTRIBUTE MAZE probe construction. ATTRIBUTE MAZE requires a different decomposition because the intended and proxy cues are attributes of objects rather than separate spatial targets. Simply detecting a feature does not establish whether the representation associates that feature with the rewarded object. We therefore distinguish feature presence from goal role and probe four binary concepts,

$$L_{\mathrm{pres}}, \qquad Y_{\mathrm{pres}}, \qquad L_{\mathrm{role}}, \qquad Y_{\mathrm{role}},$$

where L_{pres} and Y_{pres} indicate whether the line and yellow attributes are present, and L_{role} and Y_{role} indicate whether the corresponding attribute belongs to the rewarded object. The first pair measures feature availability. The second pair measures whether feature identity is available in relation to the rewarded role.

To separate presence from role, we use the configuration grid of the probe bank (Appendix E.3, Table 19) rather than sampling only from the ordinary task distribution. The grid includes states in which an attribute is visible but belongs to the lure, allowing "feature present" and "feature belongs to the rewarded object" to disagree, and both candidate objects are independently placed on uniformly sampled free cells. For the line attribute, presence and rewarded role disagree in $4/12$ training configurations and $2/6$ held-out configurations. The yellow distinction has lower support, with Y_{pres} and Y_{role} differing in only one configuration in each split. We therefore interpret the yellow-role probe more cautiously and do not assume that the two role probes provide equally strong separation of feature identity from rewarded role.

To reduce sensitivity to rollout sampling, each ATTRIBUTE MAZE checkpoint is evaluated under three explicit rollout seeds. The reported cohorts therefore contain three checkpoint seeds with three rollout evaluations per checkpoint. We report these repeated evaluations for both the textured and black-background cohorts.

ATTRIBUTE MAZE results. Figure 10 shows the corresponding cue-presence and cue-role probes across the textured and black-background cohorts. ATTRIBUTE MAZE does not show a uniform suppression of either competing cue family. Across backgrounds, training increases linear accessibility of line and yellow information at the convolutional sites, while the role probes additionally show accessibility of information about which attribute belongs to the rewarded object. Accessibility at the two downstream shared representations is more variable. At h_{pre}, black-background GM line presence does not rise above initialization (0.792 versus 0.810, Table 2), and on the black background line presence is largely decodable from raw pixels (0.975).

On the primary black-background cohort, GM agents have numerically larger training-induced gains than Non-GM agents for several cue and role probes. At $h^{(3)}$, line-role $\Delta\mathrm{AUC}$ is 0.096 for GM and 0.070 for Non-GM, while yellow-role gains are 0.243 and 0.114. Yellow presence reaches 0.442 for GM and 0.300 for Non-GM at the same site. However, with uncertainty computed across the three independent checkpoints, these GM–Non-GM differences are not resolved at $n = 3$.

The numerical ordering also varies across backgrounds. For example, at $h^{(3)}$, yellow-presence gain is 0.442 for GM and 0.300 for Non-GM on the black background, but 0.370 and 0.471 in the textured cohort. We therefore do not interpret probe magnitude as establishing a systematic cohort difference in cue accessibility.

So to conclude, training increases linear accessibility of both competing cue families at the convolutional sites of ATTRIBUTE MAZE. Yellow presence is strongly training-enhanced under the convolutional readouts, while yellow-role information is resolved above initialization at $h^{(2)}$–$h^{(3)}$ in three of the four background-by-cohort groups. Line presence and line-role gains are weaker and more site-dependent. GM-versus-Non-GM differences in probe magnitude are generally not resolved with three checkpoints, so cue accessibility alone does not explain the behavioral difference.

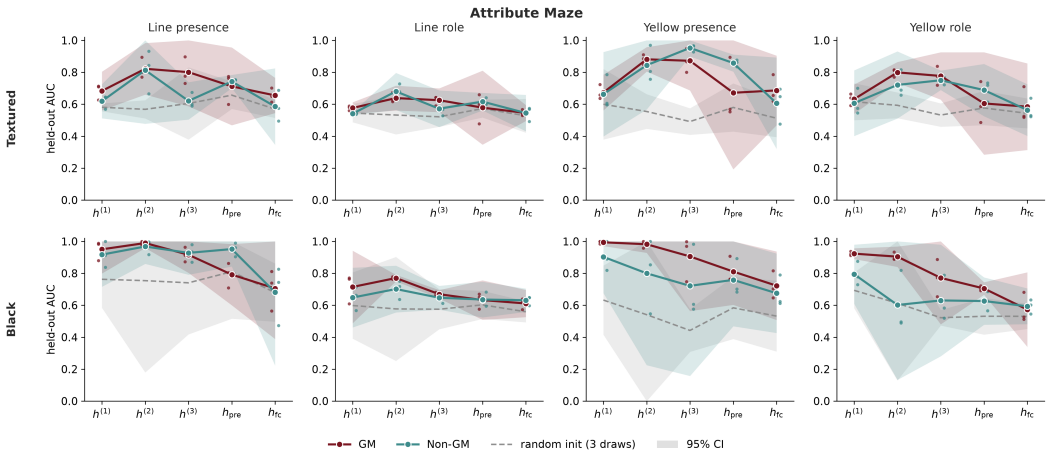

Figure 10: **Layerwise linear availability in ATTRIBUTE MAZE.** Absolute held-out ROC–AUC for line presence, line role, yellow presence, and yellow role. The top row shows the textured-background cohort and the bottom row the black-background cohort. Presence and role are probed separately because an attribute may be visible without belonging to the rewarded object. Points show the three checkpoint seeds, lines show cohort means, and shaded regions show 95% t-intervals across the three checkpoints. The grey dashed line shows the mean of three randomly initialized networks, each evaluated on the frames of all six checkpoints, with a 95% t-interval across the three networks. Both cue families remain linearly accessible across multiple representation sites, while GM–Non-GM differences vary across cue, site, and background. Initialization comparisons at h_{pre} for the black-background cohort are reported separately in Table 2.

Table 2: **Probe controls at h_{pre}.** Held-out ROC–AUC (Coin bearing: R^2). GM, Non-GM: trained agents, mean $\pm$ SD over three checkpoints. Init: probe on untrained networks, ruling out signal the architecture supplies. For ATTRIBUTE MAZE, Init is the mean of three untrained networks, each replayed on the frames of all six checkpoints. Input control: probe on pixels or agent position, ruling out targets read directly from the input.

Target	GM	Non-GM	Init	Input control
MAZE				
Cheese presence	0.544 ± 0.01	0.997 ± 0.00	0.620 ± 0.02	0.988 pixels[a]
Cheese bearing	0.712 ± 0.01	0.918 ± 0.02	0.686 ± 0.01	0.830 position
Cheese bearing, corner bearing held fixed[d]	0.598 ± 0.01	0.930 ± 0.02	0.729 ± 0.04	0.703 position
Corner bearing	0.963 ± 0.01	0.945 ± 0.01	0.727 ± 0.02	0.959 position
COIN				
Coin presence	0.775 ± 0.08	0.995 ± 0.00	0.713 ± 0.03	0.760 pixels[b]
Coin bearing (R^2)	0.523 ± 0.06	0.527 ± 0.06	-0.116 ± 0.21	0.152 position[c]
ATTRIBUTE MAZE, *black background (input control: pixels[b])*				
Line presence L_{pres}	0.792 ± 0.08	0.952 ± 0.04	0.810 ± 0.01	0.975 ± 0.01
Line role L_{role}	0.632 ± 0.05	0.636 ± 0.04	0.603 ± 0.02	0.629 ± 0.02
Yellow presence Y_{pres}	0.811 ± 0.08	0.759 ± 0.12	0.584 ± 0.02	0.662 ± 0.03
Yellow role Y_{role}	0.705 ± 0.02	0.626 ± 0.06	0.531 ± 0.02	0.556 ± 0.03

Init SD is across 3 untrained networks (MAZE), 8 (COIN), or the frames of all six checkpoints (ATTRIBUTE MAZE). Input controls are single probes, not per checkpoint, except for ATTRIBUTE MAZE, where the pixel probe is fitted on each checkpoint's frames and the SD is across the six frame sets. "Position" is a probe on agent position alone. [a]Linear probe on the 48 raw pixels of the cheese cell. [b]Linear probe on raw pixels ($16\times16\times3$ block means). [c]Split SD 0.06. The raw-pixel probe of [b] gives $R^2 = -0.552$. [d]Cheese bearing restricted to frames with corner bearing $= 1$ (1,498 frames), so any remaining signal is cheese-relative. The value comes from 10 level-grouped splits.

Conclusion. Linear probes reveal environment- and variable-specific differences rather than a uniform preservation or deletion of the intended goal. In MAZE, the cheese is almost perfectly identifiable from the relevant input pixels, but GM agents do not construct the strong cheese-presence repre-

sentation seen in Non-GM agents, and their cheese-bearing decodability is largely accounted for by agent position. In COIN, goal bearing is strongly accessible in both cohorts through h_{pre}, while coin presence becomes nearly perfectly accessible in Non-GM agents but remains substantially weaker in GM agents at the downstream shared representations. In ATTRIBUTE MAZE, training increases accessibility of both competing cue families at the convolutional sites, while the limited configuration grid prevents the role probes from establishing an equally strong distinction between feature presence and rewarded role.

These results rule out a single representational account of goal misgeneralization. The relevant goal information can be visually present without being strongly represented by the learned network, different goal variables can follow different trajectories within the same environment, and linear accessibility alone does not establish behavioral use. Figure 2 summarizes the key probes in the main text. We therefore next ask whether goal-related changes are coupled to the native policy and what functional role the resulting representations play in action selection.

B.1.1 WORLD-EDIT BEHAVIORAL DEPENDENCE

Motivation. Linear decodability establishes that goal-related information can remain available internally, but does not show whether the policy responds to that information. We therefore intervene directly on goal- and proxy-related variables in the environment and ask whether the resulting world change alters the policy, whether that change follows the edited feature, and whether it changes eventual goal achievement.

Design. We ask whether the trained policy actually responds when a goal-related variable is changed in the world. A *world edit* e changes a goal- or proxy-related variable in the level while keeping the rest of the state as fixed as the environment permits, and turns a base state s into a matched edited state s_e. Because nothing else changes, a change in the policy's output can be attributed to the edited variable. World edits are applied to the states of the conflict banks (Appendix E), and Figure 11 shows one base state and its edited state for each of the five edits.

The environments allow two kinds of edit. In MAZE and COIN the goal object can be moved without changing its identity or the level. For a state s with the goal object at location g, the *location edit*

$$s_e = \mathcal{E}^{\mathrm{loc}}_{g \to g'}(s) \tag{3}$$

moves that object from g to g' and holds the remaining level state fixed. In MAZE the cheese is moved while the layout, the agent state and the top-right destination stay unchanged. On the shared sample of the MAZE conflict bank the cheese is moved to the reachable bottom-left extreme, which gives every state the same kind of edit, and on the per-checkpoint sample it is moved to the opposite side of the agent. The structural MAZE edit instead opens access to the top-right region while otherwise preserving the maze state. In COIN, g is a cell to the right of the agent, on the shortcut side, and g' the cell one step to its left. The environment holds a single coin, so the edit moves it and does not add a second one. The end-of-level marker is left in place.

In ATTRIBUTE MAZE the goal and proxy are attributes of objects, and moving an object would change the navigation geometry of the conflict together with the cue. So we hold the two candidate locations fixed and swap their attributes. Let the candidates at locations u and v be

$$o_u = (\mathsf{line}, \mathsf{red}), \qquad o_v = (\mathsf{gem}, \mathsf{yellow}),$$

so the line cue sits at u and the yellow cue at v. The *shape swap* exchanges the shapes and keeps colours and locations fixed,

$$\mathcal{E}^{\mathrm{shape}}: \qquad (\mathsf{line}, \mathsf{red})_u,\ (\mathsf{gem}, \mathsf{yellow})_v \longmapsto (\mathsf{gem}, \mathsf{red})_u,\ (\mathsf{line}, \mathsf{yellow})_v. \tag{4}$$

The *colour swap* exchanges the colours and keeps shapes and locations fixed,

$$\mathcal{E}^{\mathrm{colour}}: \qquad (\mathsf{line}, \mathsf{red})_u,\ (\mathsf{gem}, \mathsf{yellow})_v \longmapsto (\mathsf{line}, \mathsf{yellow})_u,\ (\mathsf{gem}, \mathsf{red})_v. \tag{5}$$

These two swaps are the canonical counterfactuals of ATTRIBUTE MAZE. They resolve the same line and yellow conflict in opposite directions, since the training-consistent yellow line appears at v under the shape swap and at u under the colour swap. This return to a familiar conjunction is intended. The base state is the out-of-distribution conflict, and each swap asks how the policy

changes when the conflict is resolved toward one of the two fixed candidates, without introducing a further unfamiliar object combination. Agent position, walls, object count and candidate locations are unchanged, so the navigation geometry is identical across the base state and both swaps. The two swaps are not equally likely training perturbations, since the shape and colour vocabularies are not symmetric, and we use them only as paired resolutions of one spatial conflict.

We evaluate behavioral dependence at three increasingly downstream levels:

$$\text{policy reaction} \longrightarrow \text{edit-consistent following} \longrightarrow \text{task-level consequence}.$$

First, we measure whether the edit changes the immediate policy at all using

$$D_{\mathrm{KL},e}(s) = D_{\mathrm{KL}}\left(\pi(\cdot \mid s) \,\|\, \pi(\cdot \mid s_e)\right).$$

This quantity is direction-agnostic: a large value establishes behavioral dependence on the edited world variable, but not whether the policy changes in the behaviorally relevant direction.

Second, we also measure *signed follow*. Let $A_e^+(s)$ denote the environment-valid actions that make progress toward the candidate favoured by edit e in the edited state. We define

$$\mathrm{SF}_e(s) = \pi\big(A_e^+(s) \mid s_e\big) - \pi\big(A_e^+(s) \mid s\big),$$

where

$$\pi(A \mid s) = \sum_{a \in A} \pi(a \mid s).$$

Positive values therefore indicate that the edit increases probability mass on actions that follow the edited feature. Whenever obstacles make image-space bearing ambiguous, $A_e^+(s)$ is defined from the environment transition graph.

Finally, we measure task-level consequence using

$$\mathrm{Reach}_e(s) = \Pr_{\pi}\left(\Gamma^\star(\tau) = 1 \mid s_0 = s_e\right),$$

where $\Gamma^\star$ is the intended-goal indicator defined in Section A. Signed follow is edit-relative, whereas Reach is always goal-relative. A proxy-directed edit may therefore increase local following of the proxy while decreasing eventual intended-goal success. Table 3 reports KL and SF for every edit, while Reach is reported in the text where goal and proxy remain behaviorally separable.

For the behavioral world-edit quantities, we first average over evaluation states within each seed and then report the mean and sample standard deviation across seeds. This aggregation differs deliberately from the seed-level median used for the Fisher quantities below. In particular, Reach is a binary per-state outcome whose median would discard variation in success probability.

MAZE. In MAZE, for the cheese location edit, $D_{\mathrm{KL},e}$ and SF_e quantify the immediate response toward the relocated cheese, and Reach_e measures whether the perturbed rollout reaches it. Reach is scored with sampled actions and a 150-step cap from each state of the checkpoint's own sample of the conflict bank (140 to 142 states on 47 to 55 levels), averaged first within each level and then over levels.

For the structural edit that opens the top-right region, $D_{\mathrm{KL},e}$ and SF_e quantify the immediate response toward the newly accessible corner. Reach_e remains defined relative to the intended cheese goal and therefore measures whether this proxy-directed edit changes subsequent cheese acquisition.

COIN. In COIN, $D_{\mathrm{KL},e}$ measures the change in the immediate policy when the location edit moves the coin from the right placement to the cell on the left of the agent. With $A_e^+ = A_{\mathrm{left}}$, SF_e measures whether probability mass shifts toward the leftward actions implied by the relocated coin rather than preserving the learned rightward tendency. At trajectory level, Reach_e is the fraction of greedy rollouts from s_e, capped at 200 steps, that collect the relocated coin, with one rollout per state. The primary analysis evaluates all six checkpoints on the same 600 states of the COIN conflict bank.

ATTRIBUTE MAZE. In ATTRIBUTE MAZE, each conflict-bank state yields both swaps (Equations 4 and 5). For $e \in \{\mathrm{shape}, \mathrm{colour}\}$, $D_{\mathrm{KL},e}$ measures whether the policy reacts to the attribute

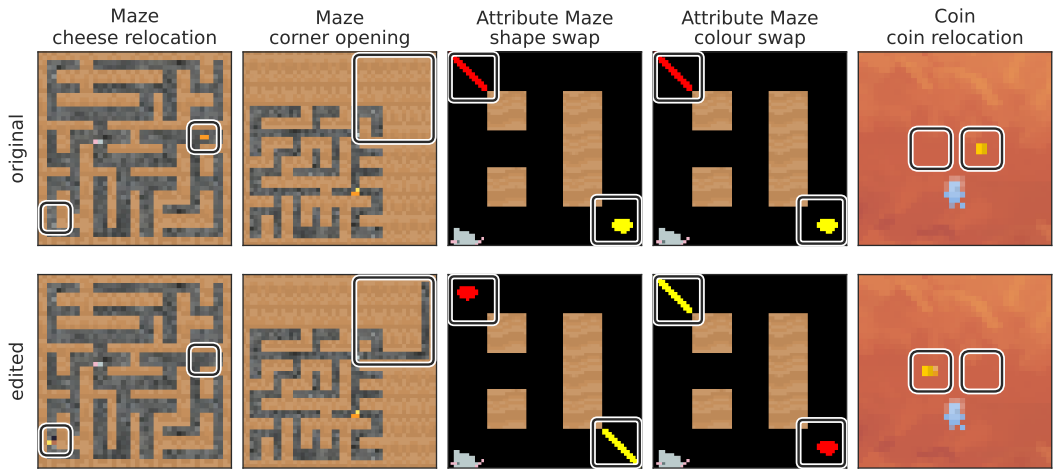

Figure 11: Observations before (top) and after (bottom) each world edit. MAZE: cheese relocation, and the structural edit that opens the top-right region. ATTRIBUTE MAZE (black background, conflict bank): shape swap and colour swap of the same base state. COIN: coin relocation from the placement right of the agent to the cell on its left (600-state conflict bank). Each column shows a single example state. Boxes enclose every changed pixel. All other pixels are identical.

Table 3: **Behavioral dependence under matched world edits.** KL measures immediate policy reaction and SF measures whether that reaction follows the edited feature. Reach is reported in the text where goal and proxy remain behaviorally separable (MAZE cheese and COIN edits). Values are mean $\pm$ sample standard deviation across three seeds, with each seed value obtained by averaging over its evaluation states (600 conflict-bank states in COIN, 196 in ATTRIBUTE MAZE). MAZE rows use the per-checkpoint sample of the conflict bank (140–142 states per checkpoint) with the cheese moved to the opposite side of the agent. On its shared 128-state sample, where the cheese is moved to the bottom-left extreme, the cheese KL is 0.015 ± 0.01 (GM) versus 1.687 ± 0.06 (Non-GM), the values used in Table 1.

Environment	Edited feature	Cohort	KL	SF
MAZE	Cheese	GM	0.012 ± 0.01	$+0.000 \pm 0.00$
		Non-GM	2.652 ± 0.16	$+0.337 \pm 0.06$
	Corner	GM	2.359 ± 0.36	$+0.109 \pm 0.04$
		Non-GM	0.126 ± 0.03	$+0.013 \pm 0.01$
ATTRIBUTE MAZE	Shape	Black GM	0.563 ± 0.25	$+0.335 \pm 0.07$
		Black Non-GM	2.814 ± 0.24	$+0.629 \pm 0.01$
		Textured GM	0.291 ± 0.32	$+0.218 \pm 0.13$
		Textured Non-GM	2.689 ± 0.16	$+0.600 \pm 0.04$
	Colour	Black GM	1.258 ± 0.31	$+0.463 \pm 0.07$
		Black Non-GM	0.025 ± 0.02	$+0.047 \pm 0.03$
		Textured GM	1.986 ± 0.65	$+0.508 \pm 0.11$
		Textured Non-GM	0.086 ± 0.08	$+0.090 \pm 0.05$
COIN	COIN	GM	0.044 ± 0.01	$+0.000 \pm 0.01$
		Non-GM	0.389 ± 0.11	$+0.179 \pm 0.05$

reassignment, while SF_e measures whether probability mass shifts toward graph-exact progress actions leading to the candidate favoured by the edited cue.

After either swap, however, the intended line shape and the training-correlated yellow colour occur on the same candidate. Target reach therefore cannot identify which of the two cues drove the resulting trajectory. We consequently use KL and signed follow as the diagnostic quantities for ATTRIBUTE MAZE, and do not interpret post-swap reach as cue-specific evidence.

Results. The world edits reveal qualitatively different patterns of behavioral dependence across the three environments (Table 3).

MAZE. MAZE shows the clearest reversal in local behavioral dependence between the intended goal and the proxy-associated destination (Table 3, MAZE rows). Relocating the cheese barely changes the GM policy, with KL 0.012 and essentially zero signed follow, while the same edit strongly reorganizes Non-GM behavior (KL 2.652, SF $+0.337$). On the shared sample of the conflict bank, with all checkpoints on the same 128 states, the gap is 0.015 versus 1.687. This difference persists over trajectories: GM agents reach the relocated cheese in only 0.080 ± 0.00 of rollouts, compared with 0.871 ± 0.03 for Non-GM agents (mean $\pm$ SD across seeds).

The pattern reverses for the proxy-associated corner at the level of the immediate policy (Table 3, MAZE Corner rows). Opening access to that region produces a large response in GM policies (KL 2.359, SF $+0.109$), while Non-GM policies respond only weakly. We do not use intended-goal Reach to quantify this corner effect because GM goal acquisition is already near floor before the edit, making further suppression uninformative.

Thus GM MAZE policies are nearly inert to relocation of the intended goal but remain strongly responsive to structural changes involving the proxy-associated destination, while Non-GM policies show the opposite local response pattern.

ATTRIBUTE MAZE. ATTRIBUTE MAZE differs from MAZE in that neither cue is behaviorally inert (Table 3, ATTRIBUTE MAZE rows). Instead, the relative influence of shape and colour changes across training regimes.

For Non-GM policies, behavior is strongly dominated by shape. On the black background, the shape swap substantially reorganizes the policy ($D_{\mathrm{KL}} = 2.814$) and shifts probability toward the shape-supported candidate (SF $= +0.629$), whereas the colour swap has little effect ($D_{\mathrm{KL}} = 0.025$, SF $= +0.047$). The same pattern appears for the textured Non-GM cohort (SF $= +0.600$ for shape versus $+0.090$ for colour).

GM policies show a different weighting. On black backgrounds, the colour swap produces the larger signed response (SF $= +0.463$), but the shape swap still produces substantial following ($+0.335$). The textured GM cohort shows the same ordering ($+0.508$ for colour versus $+0.218$ for shape), although with greater variation across seeds.

Thus goal misgeneralization in ATTRIBUTE MAZE is not a loss of sensitivity to the intended shape cue, nor a pure colour-following policy. Both cues remain behaviorally active, but their relative control over action selection differs: Non-GM policies are strongly shape-dominated, whereas GM policies place substantially greater weight on colour while retaining a measurable response to shape. This qualitative ordering is unchanged when restricting to $d > 1$ states, to the held-out evaluation split, or to either conflict sign separately.

COIN. COIN occupies an intermediate regime (Table 3, COIN rows). Relocating the coin produces a much smaller immediate policy response in GM than in Non-GM agents (KL 0.044 versus 0.389). The direction of the response also differs: GM policies do not systematically shift probability mass toward the relocated coin (SF $+0.000$), whereas Non-GM policies show a positive goal-following response ($+0.179$).

This attenuation persists at trajectory level. GM agents reach the relocated coin in only 0.019 ± 0.01 of rollouts, compared with 0.179 ± 0.09 for Non-GM agents (mean $\pm$ SD across seeds). A complementary paired-trajectory diagnostic shows the same broad attenuation of retargeting (Table 4): relative to the matched unedited rollout, coin relocation moves GM trajectories slightly farther from the new coin on average (-0.132), whereas Non-GM trajectories move closer ($+0.339$). This paired diagnostic is supplementary to the conflict-bank world-edit analysis above and is computed from a separate contiguous sweep.

Thus the GM COIN policy still reacts to coin relocation, but the response is much weaker and does not reliably retarget behavior toward the new goal.

Table 4: **Paired COIN trajectory retargeting.** In each matched pair, the edited rollout moves the coin to a free cell behind the agent, to its left, and positive values indicate that the edited trajectory approaches the relocated coin more closely than the corresponding unedited trajectory. Values are mean ± sample standard deviation across three seeds.

Cohort	Paired retargeting
GM	-0.132 ± 0.090
Non-GM	$+0.339 \pm 0.166$

B.1.2 NATIVE-POLICY SENSITIVITY: FISHER GEOMETRY

Motivation. The behavioral world edits in Appendix B.1.1 measure whether changing a goal-related property of the environment changes the policy. We next ask the corresponding question inside the policy: *how strongly is the representation change induced by the same world edit coupled to the policy's action distribution?*

Rather than selecting an internal direction independently of the environment, we let the controlled world edit determine the representation displacement that we evaluate. For a base state s, let s_e denote the matched state obtained after applying edit e. At representation site m, the edit induces

$$\Delta_e^{(m)}(s) = h^{(m)}(s_e) - h^{(m)}(s).$$

The environment-specific edits are defined in Section B.1.1. They change the goal- or proxy-related property under investigation while holding the remainder of the state fixed according to the matched counterfactual construction of each environment.

We evaluate the world-induced displacements at the two post-ReLU representation sites h_{pre} and h_{fc} defined in Section A.1. To measure how strongly these representation changes couple to the policy, we use the activation-space Fisher geometry developed in Appendix A.2, which weights directions by their local effect on the action distribution rather than by Euclidean size alone.

Fisher sensitivity. Let $F^{(m)}(s)$ denote the activation-space Fisher information at representation site m, evaluated at the unedited state s. For edit e, we define the *world-edit Fisher sensitivity*

$$Q_e^{(m)}(s) = \Delta_e^{(m)}(s)^\top F^{(m)}(s) \Delta_e^{(m)}(s).$$

As derived in Appendix A.2, for a small displacement along the edit-induced direction,

$$D_{\mathrm{KL}}\left(\pi^{(m)}(\cdot \mid h^{(m)}(s)) \,\middle\|\, \pi^{(m)}(\cdot \mid h^{(m)}(s) + t\Delta_e^{(m)}(s))\right) = \frac{t^2}{2} Q_e^{(m)}(s) + o(t^2).$$

Thus Q asks how strongly the local action distribution is coupled to the representation direction produced by the same world edit used in the behavioral analysis.

A small Q does not imply that information about the edited property is absent. Instead, it means that the particular representation displacement produced by the edit has little local effect on the native action distribution. The Fisher is always evaluated at the unedited state, while the controlled world edit determines the direction in which we probe that local geometry.

We additionally verify the local approximation directly by evaluating the exact policy KL along fractions of the finite edit displacement. The quadratic approximation is exact to the expected order near $t = 0$ and tracks the finite-edit KL closely for most conditions. For the largest Non-GM edits it instead underestimates the exact KL, including the ATTRIBUTE MAZE shape swap and MAZE goal relocation. Thus departures from the local approximation shrink rather than create the observed GM–Non-GM sensitivity gaps. We report the results below.

Sensitivity decomposition. A large Q can arise because the edit moves the representation far in policy-sensitive units, because the induced direction is unusually privileged by the downstream policy, or because both effects occur together. We therefore use the scale-invariant decomposition derived in Appendix A.2,

$$Q_e^{(m)}(s) = \underbrace{G_e^{(m)}(s)}_{\text{relative directional gain}} \quad \underbrace{D_{F,e}^{(m)}(s)}_{\text{Fisher-unit displacement}},$$

		$10^3\,\mathrm{tr}(F)/d$		$\|\Delta\|_2$	
Environment	Site	GM	Non-GM	GM	Non-GM
MAZE	h_{pre}	0.515 ± 0.13	1.192 ± 0.36	1.212 ± 0.07	18.052 ± 3.10
	h_{fc}	4.328 ± 0.41	5.393 ± 0.11	0.849 ± 0.04	18.591 ± 2.64
COIN	h_{pre}	1.336 ± 0.77	0.538 ± 0.12	3.692 ± 0.73	11.814 ± 1.49
	h_{fc}	7.219 ± 4.54	3.712 ± 0.25	1.714 ± 0.33	7.685 ± 1.38
ATTRIBUTE MAZE	h_{pre}	0.225 ± 0.08	0.096 ± 0.03	14.271 ± 1.45	43.387 ± 10.53
	h_{fc}	0.893 ± 0.16	0.774 ± 0.14	8.792 ± 1.00	23.476 ± 5.40

Table 5: **The two factors of D_F.** Mean Fisher sensitivity $\mathrm{tr}(F)/d$ at the unedited state and Euclidean norm of the edit-induced displacement, each taken as the median over the evaluation states of one checkpoint and then reported as mean $\pm$ SD across the three checkpoints of a cohort. States are those of Table 6: 128 in MAZE (cheese relocation), 600 in COIN (coin relocation) and 196 in ATTRIBUTE MAZE (black background, shape swap). Because medians do not multiply, the product of the two factors only approximates the D_F of Table 6.

with

$$G_e^{(m)}(s) = \frac{Q_e^{(m)}(s)/\|\Delta_e^{(m)}(s)\|_2^2}{\mathrm{tr}(F^{(m)}(s))/d_m}$$

and

$$D_{F,e}^{(m)}(s) = \|\Delta_e^{(m)}(s)\|_2^2 \frac{\mathrm{tr}(F^{(m)}(s))}{d_m}.$$

D_F measures how far the counterfactual moves the representation in units of the network's average local policy sensitivity. G measures whether the direction of that movement is more or less policy-sensitive than an isotropic direction in the same activation coordinate system. Together, they let us distinguish a world edit that produces little policy-relevant representational movement from one whose induced direction is particularly privileged by the downstream policy.

Displacement versus policy sharpness. D_F is the product of the squared displacement norm and the mean Fisher sensitivity $\mathrm{tr}(F)/d$, and Corollary 1 shows that the second factor vanishes as the policy becomes deterministic. A small GM D_F could therefore come from a sharper GM policy rather than from a smaller movement of the representation. Table 5 separates the two factors on the states of Table 6. The mean Fisher sensitivity of GM policies is at most 2.3 times smaller than that of Non-GM policies (MAZE, h_{pre}), and in COIN and ATTRIBUTE MAZE it is larger. The displacement norm instead differs by a factor of 15 to 22 in MAZE and 3 to 4.5 in COIN, which gives factors of about 220 to 480 and 10 to 20 in $\|\Delta\|_2^2$. So in these networks the GM–Non-GM gap in D_F comes from the size of the displacement and not from a collapse of the Fisher toward a deterministic policy. In ATTRIBUTE MAZE, where the action entropy was saved, GM policies are also less sharp than Non-GM policies on the conflict-bank states (2.331 ± 0.24 against 1.962 ± 0.14 nats, three checkpoints each, black background). This split has one limit. D_F is invariant to a rescaling of the representation (Proposition 3), but its two factors are not, because $\|\Delta\|_2$ grows and $\mathrm{tr}(F)/d$ shrinks with the scale of h. The comparison is therefore a statement about the trained coordinates of each network, and we hold it with moderate confidence.

Comparing representation sites. Q is approximately the same at h_{pre} and h_{fc}, and this follows from the construction rather than from the data. The policy depends on h_{pre} only through h_{fc}, so with $J = \partial h_{\mathrm{fc}}/\partial h_{\mathrm{pre}}$ the Fisher information pulls back exactly as $F_{\mathrm{pre}} = J^\top F_{\mathrm{fc}} J$. To first order the edit-induced displacements satisfy $\Delta_{\mathrm{fc}} \approx J\Delta_{\mathrm{pre}}$, and therefore

$$Q_{\mathrm{pre}} = \Delta_{\mathrm{pre}}^\top J^\top F_{\mathrm{fc}} J\, \Delta_{\mathrm{pre}} \approx \Delta_{\mathrm{fc}}^\top F_{\mathrm{fc}}\, \Delta_{\mathrm{fc}} = Q_{\mathrm{fc}}.$$

Remaining differences come from the nonlinearity of the fc layer over a finite edit. The two factors G and D_F are not invariant in this way, because they depend on $\|\Delta\|_2$ and on $\mathrm{tr}(F)/d$, which change with the coordinates and the dimension of each site. A change in the split between G and D_F from h_{pre} to h_{fc} therefore describes how the fc layer re-expresses the same policy sensitivity, not where goal information is lost. We compare G and D_F only between cohorts at the same site, and localize where goal information becomes inaccessible with the probes (Appendix B.1) and with rewiring from both sites (Section C.2).

Signed target-selection effect. Q tells us how strongly the policy responds locally, but it is unsigned. We therefore also ask what the same representation displacement does to target selection.

For each evaluated state s, let

$$C_s^{(m)}(h)$$

denote the environment-specific contrast between intended- and proxy-consistent action sets. We measure its directional derivative along the world-induced displacement,

$$\partial_{\Delta_e} C_s^{(m)} = \nabla_h C_s^{(m)} \left(h^{(m)}(s)\right)^\top \Delta_e^{(m)}(s).$$

This is the first-order change in target-selection contrast induced by moving the representation in the direction produced by edit e.

Because the raw derivative also scales with the magnitude of the representation displacement, we use the normalized derivative as a secondary diagnostic,

$$\widetilde{\partial_{\Delta_e} C_s^{(m)}} = \frac{\partial_{\Delta_e} C_s^{(m)}}{\|\Delta_e^{(m)}(s)\|_2}.$$

The raw derivative measures the target-selective effect of the actual world-induced displacement. The normalized version instead asks how much target-selective leverage its direction carries per unit Euclidean representation change. Throughout, the sign is interpreted relative to the environment-specific target contrast and the direction of the corresponding edit.

Evaluation states. The evaluation populations differ across environments because each assay requires states that make the corresponding goal–proxy distinction meaningful. Each environment uses its conflict bank (Table 18), with the 128 states of the shared MAZE sample, the 600 COIN states and the 196 ATTRIBUTE MAZE states. Within each environment, the same evaluation states are shared across the agents being compared. In COIN, F is evaluated at the base state with the coin at the placement right of the agent, Δ is the change in h when the coin moves to the cell on its left, and the signed contrast is $C_s = \pi(A_{\text{left}}) - \pi(A_{\text{right}})$, so following the edit reads positive.

Results: the behavioral difference is visible inside the policy. The world-edit experiments above gave us an external view of native goal use. The Fisher analysis asks whether the same distinction is visible internally, by following the representation displacement created by exactly those world changes. Figure 12 and Table 6 make the contrast particularly clear. The three environments do not simply differ by one common reduction in policy sensitivity. Instead, the intended-goal edit reaches the native policy in different ways.

ATTRIBUTE MAZE. ATTRIBUTE MAZE is the clearest case in which both sides of the conflict remain wired into the native policy. In Fig. 12, both shape- and colour-induced displacements sit far away from the randomly initialized reference at both representation sites, and both produce substantial total Fisher sensitivity.

What changes between GM and Non-GM is the organization of that sensitivity. For the shape swap, Non-GM agents move farther in Fisher-sensitive units ($D_F = 0.395$ versus 0.066 at h_{fc}), which gives them the larger total response ($Q = 3.569$ versus 1.097). For the colour swap the ordering reverses. GM agents show a strong response ($Q = 2.126$), while the Non-GM response is almost absent ($Q = 0.029$). This organization is already visible at h_{pre} and is reproduced under textured observations.

The geometry in Fig. 12 also makes a second point visible. The GM displacements occupy comparatively high-gain directions for both swaps, whereas much of the Non-GM shape response comes from a larger displacement in Fisher-sensitive units. So the two cohorts can reach substantial policy sensitivity through rather different combinations of displacement and directional privilege. The important point is not that one scalar is universally larger, but that both cue-induced changes remain strongly connected to the GM policy.

The signed target-selection effect then tells us what these two active directions do. Because the shape swap and the colour swap favour opposite candidates, the contrast C_s assigns them opposite

Environment	Site	Edit	Cohort	Q	G	D_F
MAZE	h_{pre}	Cheese	GM	0.001 ± 0.00	1.641 ± 0.34	0.001 ± 0.00
			Non-GM	1.650 ± 0.08	4.800 ± 0.46	0.348 ± 0.01
			init	0.000 ± 0.00	0.959 ± 0.03	0.000 ± 0.00
	h_{fc}	Cheese	GM	0.001 ± 0.00	0.727 ± 0.25	0.002 ± 0.00
			Non-GM	1.689 ± 0.05	1.141 ± 0.39	1.667 ± 0.44
			init	0.000 ± 0.00	0.962 ± 0.03	0.000 ± 0.00
COIN	h_{pre}	COIN	GM	0.021 ± 0.01	1.508 ± 0.39	0.016 ± 0.01
			Non-GM	0.324 ± 0.10	4.112 ± 1.03	0.070 ± 0.00
			init	0.000 ± 0.00	1.028 ± 0.11	0.000 ± 0.00
	h_{fc}	COIN	GM	0.021 ± 0.01	1.350 ± 0.61	0.019 ± 0.01
			Non-GM	0.339 ± 0.12	1.784 ± 0.05	0.207 ± 0.07
			init	0.000 ± 0.00	1.014 ± 0.09	0.000 ± 0.00
ATTRIBUTE MAZE	h_{pre}	Shape	GM	1.354 ± 0.63	32.304 ± 9.24	0.048 ± 0.02
			Non-GM	2.458 ± 0.25	16.454 ± 6.70	0.162 ± 0.04
			init	0.000 ± 0.00	0.963 ± 0.05	0.000 ± 0.00
	h_{pre}	Colour	GM	2.821 ± 0.77	27.647 ± 3.45	0.090 ± 0.02
			Non-GM	0.032 ± 0.03	2.633 ± 1.73	0.011 ± 0.01
			init	0.000 ± 0.00	1.032 ± 0.08	0.000 ± 0.00
	h_{fc}	Shape	GM	1.097 ± 0.45	17.165 ± 6.34	0.066 ± 0.02
			Non-GM	3.569 ± 0.27	10.268 ± 4.46	0.395 ± 0.15
			init	0.000 ± 0.00	0.914 ± 0.05	0.000 ± 0.00
	h_{fc}	Colour	GM	2.126 ± 0.83	13.759 ± 3.99	0.142 ± 0.03
			Non-GM	0.029 ± 0.03	2.121 ± 0.89	0.013 ± 0.01
			init	0.000 ± 0.00	1.015 ± 0.07	0.000 ± 0.00

Table 6: **World-edit Fisher sensitivity.** Values are mean ± SD across three checkpoints per cohort (three random initializations for init), each summarized by its median over the evaluation states: 128 in MAZE, 600 in COIN and 196 in ATTRIBUTE MAZE (black background). $Q = GD_F$ measures local sensitivity to the edit-induced representation displacement, with D_F describing its Fisher-unit size and G its relative directional gain.

signs whenever the policy follows both edits, so the sign pattern alone is not evidence of competition. What matters is that both effects are large and consistent (Figure 13). For GM agents, each displacement shifts preference toward the candidate its swap favours in almost every conflict state (-0.665 for shape and $+0.918$ for colour at h_{fc}), and the same holds at h_{pre}. Non-GM agents follow the shape edit just as strongly (-0.897), while their colour response is an order of magnitude smaller ($+0.063$). Thus both ATTRIBUTE MAZE cues are read by the native GM policy, and each moves target preference toward its own candidate. Whether the two cues actually pull against each other is a separate question, which we test next through the push–pull redistribution of target support and the within-state displacement geometry (Appendix B.3.1).

MAZE. MAZE looks almost like the opposite case. Behaviorally, relocating the cheese barely moved the GM policy. From the inside, the same edit produces almost no policy-sensitive representational movement. At both representations, GM sensitivity is approximately $Q = 0.001$, compared with roughly 1.65 in Non-GM agents.

Figure 12 shows where this separation comes from. The cheese displacement in GM agents lies orders of magnitude lower in D_F than in the Non-GM agents. At h_{pre}, D_F is approximately 0.001 in GM agents and 0.348 in Non-GM agents. At h_{fc}, the difference is 0.002 versus 1.667. The direction itself is not unusually privileged enough to compensate for this tiny displacement. So although cheese information can still be represented elsewhere in the network, moving the cheese produces almost no change along representation directions that matter to the native policy.

The signed view tells the same story. In Fig. 13, the GM distribution stays tightly around zero at both representations, whereas Non-GM agents show a substantial target-selective response. The internal assay therefore mirrors the behavioral relocation result: for these GM agents, changing the intended goal barely reaches native action selection.

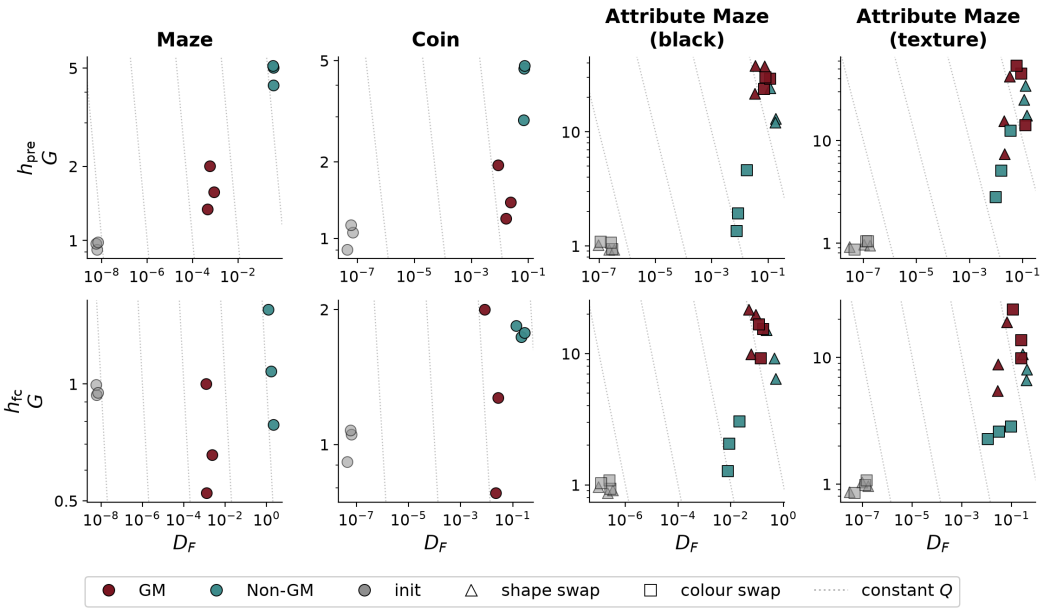

Figure 12: **World-edit sensitivity separates into displacement and directional gain.** Each trained point is one seed, with D_F and G summarized across evaluation states at h_{pre} (top) and h_{fc} (bottom). Randomly initialized networks provide the architectural reference. ATTRIBUTE MAZE is shown separately for black and textured observations, with shape swaps as triangles and colour swaps as squares. Dotted diagonals are constant-Q reference contours. Three checkpoints per cohort, and no interval is drawn. Because G and D_F are summarized separately, their plotted product need not equal the reported median Q exactly.

COIN. COIN lies between these two regimes. The intended-goal displacement remains measurably coupled to the GM policy, but substantially less strongly than in Non-GM agents. At h_{pre}, total sensitivity is $Q = 0.021$ for GM agents and 0.324 for Non-GM agents. At h_{fc}, the corresponding values are 0.021 and 0.339. Thus Non-GM sensitivity is roughly fifteen times larger at both representation sites, a much smaller separation than in MAZE but still a clear attenuation of native goal coupling.

The decomposition in Fig. 12 shows how that difference changes across the two representation sites. At h_{pre}, directional gain is 1.508 in GM agents and 4.112 in Non-GM agents, while D_F is 0.016 and 0.070. At h_{fc}, the gain values become more similar (1.350 versus 1.784), while the displacement difference grows (0.019 versus 0.207). Because Q itself is nearly unchanged between the two sites, this shift reflects how the fc layer re-expresses the coin displacement rather than a loss of sensitivity at that layer (Appendix B.1.2).

The signed effect sharpens this distinction. GM agents remain close to zero at both representation sites, while Non-GM agents show a positive target-selection effect. Thus the COIN goal displacement is not disconnected from the native GM policy, but its influence is substantially attenuated and does not produce a reliable goal-directed shift in action preference.

Finite-edit calibration. Because Q is a local quadratic approximation, we also compare $\frac{1}{2}t^2Q$ with the exact policy KL along fractions of the full edit-induced displacement. Near $t = 0$, the approximation converges to the exact KL as expected. Across the complete ATTRIBUTE MAZE evaluation bank, it remains close for most conditions at h_{fc}. The largest deviations occur for strong Non-GM edits, where the quadratic approximation underestimates the finite KL by approximately 1.4–$2.1\times$. Thus the finite-edit correction strengthens rather than creates the observed GM–Non-GM sensitivity differences.

Together. The Fisher view therefore recovers the same progression that was visible from behavior, but now from inside the policy. In MAZE, the intended-goal edit produces almost no policy-relevant

Environment	Site	Direction	GM	Non-GM	init
MAZE	h_{pre}	$\Delta_\star$	-0.001 ± 0.00	-0.271 ± 0.07	0.000 ± 0.00
	h_{fc}	$\Delta_\star$	-0.001 ± 0.00	-0.269 ± 0.02	0.000 ± 0.00
COIN	h_{pre}	$\Delta_\star$	-0.009 ± 0.02	$+0.072 \pm 0.03$	0.000 ± 0.00
	h_{fc}	$\Delta_\star$	-0.010 ± 0.02	$+0.139 \pm 0.05$	0.000 ± 0.00
ATTRIBUTE MAZE	h_{pre}	$\Delta_\star$	-0.718 ± 0.24	-0.704 ± 0.06	0.000 ± 0.00
	h_{pre}	Δ_p	$+1.033 \pm 0.28$	$+0.068 \pm 0.05$	0.000 ± 0.00
	h_{fc}	$\Delta_\star$	-0.665 ± 0.21	-0.897 ± 0.02	0.000 ± 0.00
	h_{fc}	Δ_p	$+0.918 \pm 0.31$	$+0.063 \pm 0.04$	0.000 ± 0.00

Table 7: **Signed target-selection effects.** Values are mean $\pm$ SD across three checkpoints per cohort, each summarized over the evaluation states of Table 6. The derivative measures the first-order change in the environment-specific target-selection contrast along the world-induced displacement (Definition 8). Sign key: a policy that follows the edit reads negative for the MAZE cheese relocation and the ATTRIBUTE MAZE shape swap, and positive for the ATTRIBUTE MAZE colour swap and the COIN relocation.

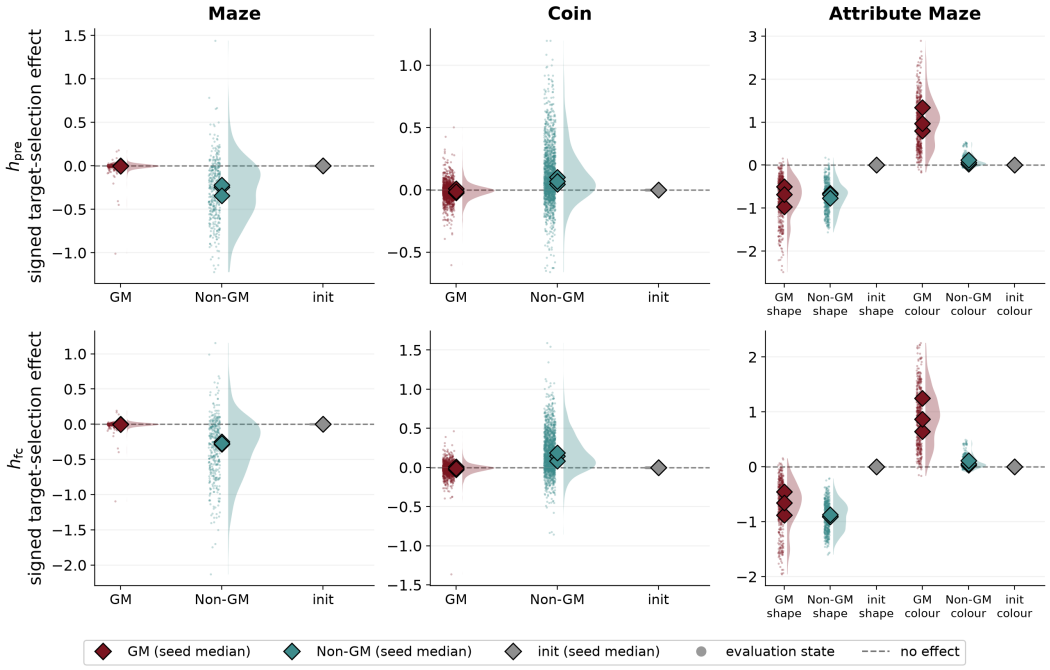

Figure 13: **Signed effects of world-induced representation changes on target selection.** Small points show evaluation states, densities summarize their distribution, and diamonds show the three seed-level summaries. The top row reports h_{pre} and the bottom row h_{fc}. Randomly initialized networks are shown at zero for reference. No interval across checkpoints is drawn. ATTRIBUTE MAZE panels show black-background policies, separately for the shape and the colour swap. Signs follow the key in Table 7.

internal movement and almost no signed target-selection effect. COIN retains more coupling, but the goal signal is attenuated relative to the Non-GM policy. ATTRIBUTE MAZE is qualitatively different: both the intended and proxy edits remain strongly connected to the native policy, and each moves target preference toward its own candidate.

So the common behavioral label "goal misgeneralization" hides different relationships between the intended goal and the native policy. The goal can be nearly ignored, weakly coupled, or actively read alongside a competing proxy. The next sections take each environment in turn and investigate these GM fingerprints in more detail.

B.1.3 POLICY-FISHER INPUT SALIENCY

Definition. To see which parts of the observation the action distribution depends on, we move the activation-space Fisher of Definition 4 to the input. For an observation o with colour channels c and pixels x we define

$$S(x) = \sum_c \mathbb{E}_{a\sim\pi(\cdot|o)} \left[\left(\frac{\partial \log \pi(a \mid o)}{\partial o_{c,x}} \right)^2 \right],$$

the diagonal of the input-space Fisher information (Zhao et al., 2019), summed over the colour channels of pixel x. Writing z for the logits, $\partial \log \pi(a \mid o) = \partial z_a - \sum_b \pi(b \mid o)\, \partial z_b$, so

$$S(x) = \sum_c \mathrm{Var}_{a\sim\pi(\cdot|o)} \left[\frac{\partial z_a}{\partial o_{c,x}} \right],$$

the variance of the logit gradients across actions, weighted by how likely the policy is to take each action. For a small change ε of one channel of pixel x, the policy moves by

$$D_{\mathrm{KL}} \left(\pi(\cdot \mid o) \,\|\, \pi(\cdot \mid o + \varepsilon e_{c,x}) \right) = \frac{\varepsilon^2}{2}\, \mathbb{E}_{a\sim\pi} \left[\left(\frac{\partial \log \pi(a \mid o)}{\partial o_{c,x}} \right)^2 \right] + O(\varepsilon^3),$$

so $\frac{1}{2}\varepsilon^2 S(x)$ is the change of the whole action distribution, summed over channels, when pixel x changes slightly. $S(x)$ therefore measures what the policy's decision depends on, not what the policy pursues. A pixel can matter because changing it would tip the choice between two candidates, whichever of the two the policy prefers. We compute S exactly, with one backward pass per action (15 per state), and a finite-difference check of one gradient entry agrees to within 2%.

Relation to saliency maps. Gradient saliency (Simonyan et al., 2014) differentiates a single output, usually the logit of one action or the value, so it also responds to changes that raise all logits together and leave the policy unchanged. Perturbation saliency (Greydanus et al., 2018) blurs a patch and measures the squared change of the full logit vector, which covers all actions but weights them equally and uses a finite, spatially extended perturbation. $S(x)$ is a Fisher-weighted variant of gradient saliency rather than a new kind of attribution. It concerns the whole action distribution, ignores common shifts of all logits by construction and weights each action by its probability.

How we read it. Saliency maps suggest hypotheses rather than explain behavior (Atrey et al., 2020), so we use $S(x)$ only in four restricted ways. First, GM and Non-GM policies are scored on identical states, so a difference at a fixed region is a difference between networks and not between scenes. Second, every region is read against a randomly initialized network on the same states, because an untrained network already gives high-contrast sprites 1.9 to 2.8 times the image-mean saliency. Third, we normalize per image. Enrichment is the saliency per pixel in a region divided by the image-mean saliency per pixel of the same state, reported as a geometric mean over states because the per-state ratio is heavy-tailed. As a π-weighted variance, S shrinks everywhere when the policy becomes nearly deterministic, and this normalization removes that global scale, so entropy differences between cohorts do not enter the comparison. Fourth, saliency decays with distance from the agent, with the natural log of the enrichment falling by about 0.2 per cell in MAZE. The cheese is usually nearer the agent than the top-right extreme (median 3 against 8 cells), so a contrast between two regions within one network is confounded by distance, and we compare cohorts at a fixed region instead. Causal claims rest on the world edits (Appendix B.1.1) and on the corner edits of Appendix B.2.1, not on the maps.

States and regions. In MAZE we use the shared 128-state sample of the conflict bank. The goal region is the cheese cell and the proxy region the 3×3 cells around the reachable top-right extreme. In COIN we use the 589 of the 600 conflict-bank states whose replay reproduces the bank's agent pose and both coin cells, with the coin one cell left of the agent. The goal region is the coin cell and the proxy region the end-wall block, which is on screen in 103 of these states. Regions are exclusive, with the agent cell taking priority over goal and proxy.

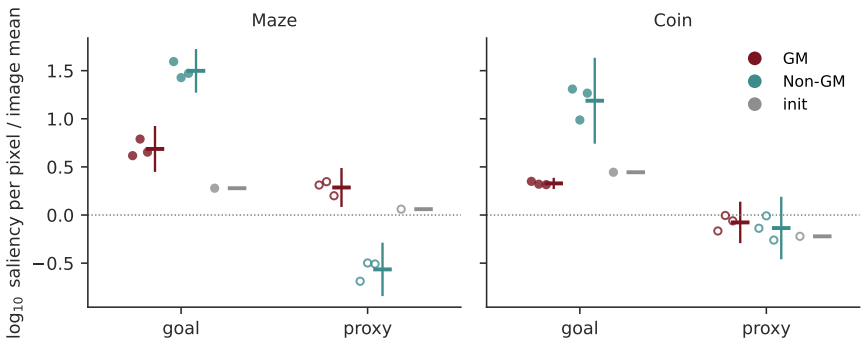

Figure 14: **Fisher-weighted input saliency in MAZE and COIN.** Saliency per pixel relative to the image mean, $\log_{10}$ of the geometric mean over states. Goal is the cheese or coin cell (filled markers). Proxy is the 3×3 cells around the reachable top-right extreme in MAZE and the end-wall block in COIN, on the 103 states where it is visible (open markers). Small dots are checkpoints, ticks the mean over three with its 95% t-interval. The randomly initialized baseline is one network. Saliency decays with distance from the agent, and the cheese is usually nearer the agent than the top-right region.

B.2 WHAT ROLE DOES THE INTENDED-GOAL INFORMATION PLAY?

Fisher sensitivity establishes whether a goal-related representation displacement is coupled to the native policy, but not what role that information plays in action selection. A feature may influence the action distribution without serving as the target around which behavior is organized. We therefore ask a stronger question: when intended-goal information remains connected to the policy, does it actually organize target selection?

What the policy depends on. Before asking what the chosen action serves, we ask which parts of the observation the action distribution depends on, using the policy-Fisher input saliency of Appendix B.1.3 (Fig. 14). In MAZE, the Non-GM decision depends sharply on the cheese, which carries $31.5\times$ the image-mean saliency, and it depends less than average on the region around the reachable top-right extreme. The GM decision still depends on the cheese more than a randomly initialized network does, but far less than the Non-GM decision, and the top-right region is where GM saliency exceeds both the Non-GM policies and init (Fig. 15). In COIN, the Non-GM decision depends on the coin ($15.4\times$ the image mean), whereas GM coin saliency ($2.13\times$) stays at the level of a random network, which reaches it from sprite contrast alone. No cohort's decision depends on the end wall more than on an average pixel, so the GM COIN policy shows no localized substitute target on screen (Fig. 16). The GM cheese saliency above init does not contradict Section 3.1, where probes find no linear cheese information, or the inert cheese edit. $S(x)$ is the local sensitivity of the action distribution to the cheese pixels at the current frame. It is neither decodable information about where the goal is nor evidence that moving the goal redirects the agent. The saliency agrees in direction with the world edits of Table 3, where the goal object moves Non-GM policies far more than GM policies and the top-right region matters only for GM MAZE policies.

B.2.1 MAZE: FUNCTIONAL TARGET SUBSTITUTION

Motivation. The preceding analyses show a strong asymmetry in MAZE. Our probes find no non-trivial cheese information in GM agents (Appendix B.1), and relocating the cheese produces almost no change in the GM policy. By contrast, world edits of the proxy-associated top-right region produce a large behavioral response. This suggests that behavior is organized around the proxy rather than the intended goal, but does not yet determine what kind of policy structure the proxy induces.

A strong preference for the top-right region could arise because the policy treats it as an alternative destination, or because it follows a simpler directional strategy such as repeatedly moving up and right. We distinguish these possibilities in two steps. First, we compare local progress toward the cheese with progress toward the proxy-associated destination at states where the two require

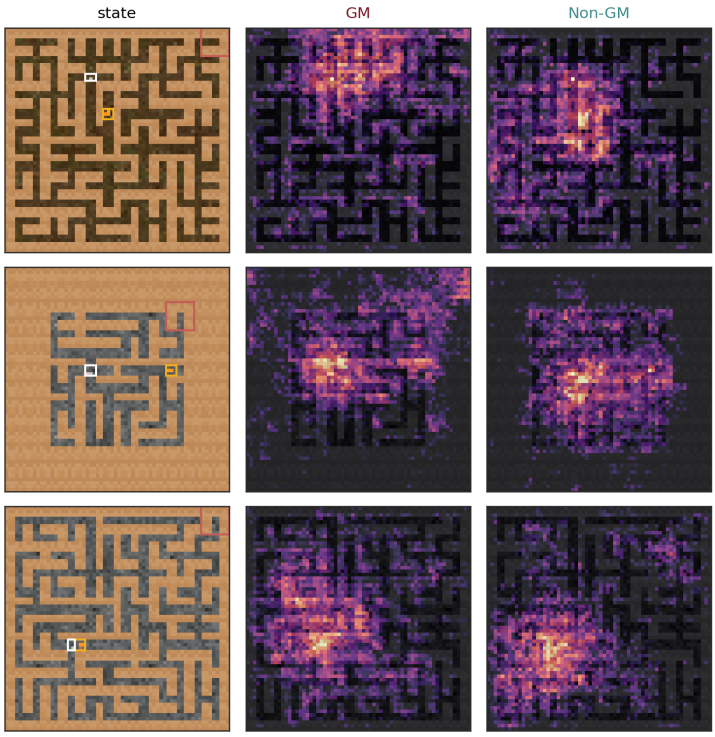

Figure 15: **Example saliency maps in MAZE.** Rows are three states from the shared 128-state sample of the conflict bank, the first three with the cheese at least 6 cells from the top-right extreme, with no selection on saliency. Left: observation with the agent (white), the cheese (orange) and the 3×3 region around the reachable top-right extreme (red). Middle and right: $\log_{10}$ saliency relative to the image mean (colour range -1 to 2) over the greyscale frame, for one GM and one Non-GM checkpoint.

different actions. Second, the detour test asks whether the policy continues to make progress toward the proxy when doing so requires moving outside the naive up-right direction.

Action-set discrimination. We evaluate the policy on the subset of states for which progress toward the intended and proxy-associated targets requires different local actions. We define the conflict-state set

$$\mathcal{S}_{\mathrm{C}} = \{s \in \mathcal{S} : A_\star(s) \neq A_p(s)\},$$

where $A_\star(s)$ and $A_p(s)$ are the graph-exact progress-action sets toward the cheese and proxy-associated destination from Definition 9. At states in $\mathcal{S}_{\mathrm{C}}$, the two candidate targets imply different local control choices, so the policy's action distribution can distinguish which target better organizes behavior. The evaluation set is the detour bank of Table 18, 1,500 states on 618 levels whose cells are chosen from the maze graph alone (Appendix E). Every one of these states satisfies $A_\star(s) \cap A_p(s) = \varnothing$, so $|\mathcal{S}_{\mathrm{C}}| = 1{,}500$ is the denominator of Choice and Mass below, and no state on which one action serves both targets enters it. Because several action indices produce the same grid move, $A_\star(s)$ and $A_p(s)$ contain every index of the corresponding move.

For $j \in \{\star, p\}$, we define

$$\mathrm{Choice}_j = \frac{1}{|\mathcal{S}_{\mathrm{C}}|} \sum_{s \in \mathcal{S}_{\mathrm{C}}} \mathbf{1}\left[\arg\max_{a \in \mathcal{A}} \pi(a \mid s) \in A_j(s)\right],$$

the fraction of conflict states on which the policy's highest-probability action makes progress toward target j.

We additionally define

$$\mathrm{Mass}_j = \frac{1}{|\mathcal{S}_{\mathrm{C}}|} \sum_{s \in \mathcal{S}_{\mathrm{C}}} \sum_{a \in A_j(s)} \pi(a \mid s),$$

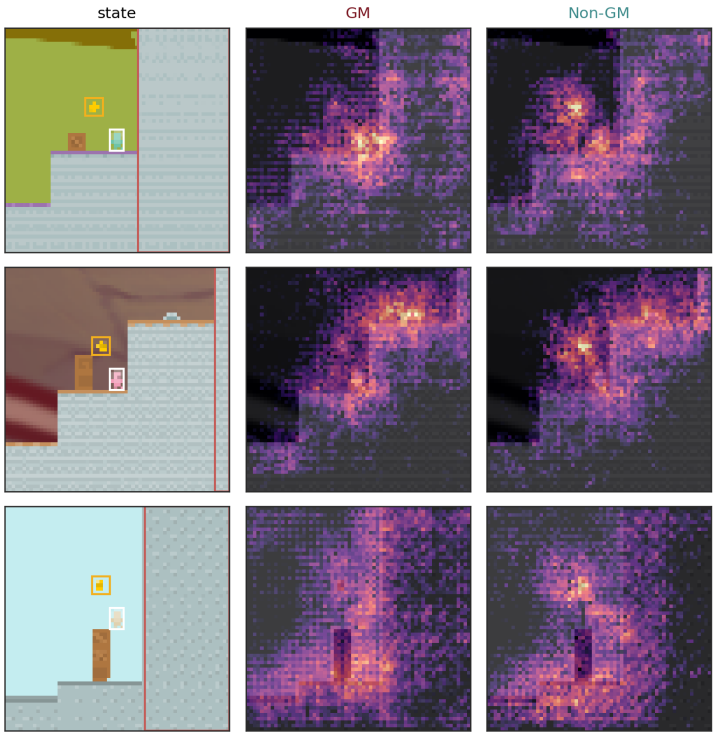

Figure 16: **Example saliency maps in COIN.** Rows are the first three conflict-bank states whose replay reproduces the bank pose and that show the end wall, with no selection on saliency. Left: observation with the agent (white), the coin one cell to its left (yellow) and the end-wall block (red). Middle and right: $\log_{10}$ saliency relative to the image mean (colour range -1 to 2) over the greyscale frame, for one GM and one Non-GM checkpoint.

the mean policy probability assigned to actions that make progress toward target j. Choice summarizes which target is reflected in the selected action, while Mass measures whether the same preference is present in the full action distribution.

Detour test. A preference for corner-progress actions could still arise from a simple directional shortcut because the proxy-associated destination lies in the top-right region. We therefore define the detour subset of the detour bank

$$\mathcal{S}_{\mathrm{D}} = \left\{ s \in \mathcal{S}_{\mathrm{C}} : A_p(s) \cap A_{\mathrm{up/right}} = \varnothing \right\}.$$

where $A_{\mathrm{up/right}}$ denotes immediate upward or rightward movement. On these states, every action that makes graph-exact progress toward the proxy-associated destination lies outside the naive up-right action family. A fixed up-right strategy therefore cannot explain corner progress on this subset, whereas progress toward the proxy-associated destination can.

Let

$$a_\pi(s) = \arg\max_{a \in \mathcal{A}} \pi(a \mid s)$$

denote the greedy policy action. We classify $a_\pi(s)$ according to whether it makes progress toward the proxy-associated destination, makes progress toward the cheese, follows the naive up-right strategy without making progress toward either target, or belongs to none of these groups.

Applying the detour criterion yields 168 states. For all of them,

$$A_p(s) \cap A_\star(s) = \varnothing,$$

Table 8: **Functional-role discrimination in MAZE and COIN.** Choice reports the fraction of conflict states on which the highest-probability action belongs to the intended-goal ($\star$) or proxy-associated (p) action set. Mass reports the mean policy probability assigned to the corresponding action set. Values are mean $\pm$ sample standard deviation across three checkpoints per cohort, on the 1,500 states (618 levels) of the MAZE detour bank and the 600 states of the COIN conflict bank.

Environment	Cohort	$\text{Choice}_\star$	Choice_p	$\text{Mass}_\star$	Mass_p
MAZE	GM	0.013 ± 0.02	0.962 ± 0.05	0.010 ± 0.01	0.916 ± 0.04
	Non-GM	0.585 ± 0.00	0.076 ± 0.02	0.440 ± 0.01	0.087 ± 0.01
COIN	GM	0.047 ± 0.04	0.637 ± 0.04	0.073 ± 0.02	0.423 ± 0.02
	Non-GM	0.318 ± 0.08	0.490 ± 0.10	0.244 ± 0.05	0.308 ± 0.06

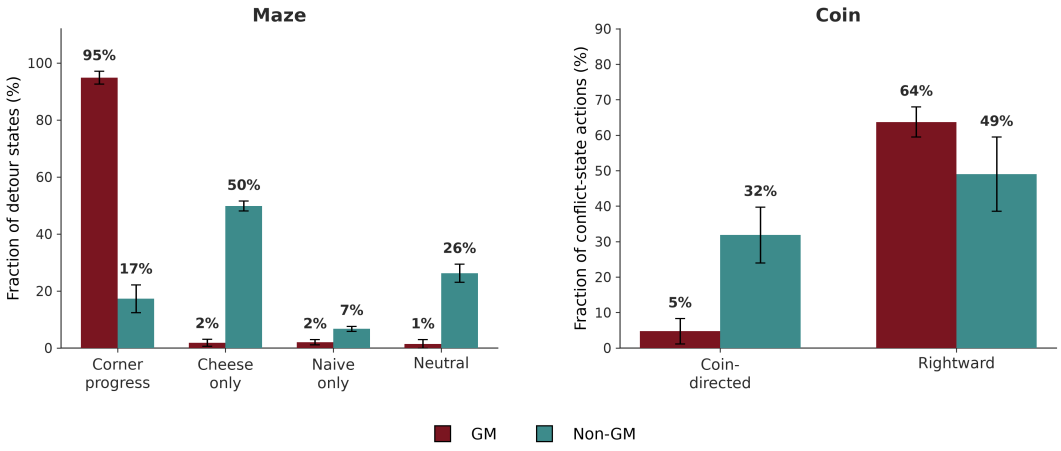

Figure 17: **Different functional organization of MAZE and COIN policies.** Left: action categories on MAZE detour states, where graph-exact progress toward the proxy-associated destination requires moving outside the naive up-right action family. GM policies overwhelmingly continue to make corner progress, while Non-GM policies show substantially more cheese-progress and neutral action selection. Right: on COIN conflict states, GM policies rarely choose coin-directed actions and instead retain a strong rightward tendency. MAZE categories: corner progress, cheese progress ("Cheese only", since no action makes progress toward both), up-right actions that make progress toward neither target ("Naive only"), and all other actions (neutral). Bars show cohort means and error bars show the sample standard deviation across three seeds, on the 168 MAZE detour states and the 600 COIN conflict-bank states.

so no action simultaneously makes progress toward both candidate targets. We therefore use the mutually exclusive categories

$$\begin{aligned} \text{Corner progress:} &\quad a_\pi(s) \in A_p(s), \\ \text{Cheese progress:} &\quad a_\pi(s) \in A_\star(s), \\ \text{Naive only:} &\quad a_\pi(s) \in A_{\text{up/right}} \setminus \left(A_p(s) \cup A_\star(s)\right), \\ \text{Neutral:} &\quad a_\pi(s) \notin A_p(s) \cup A_\star(s) \cup A_{\text{up/right}}. \end{aligned}$$

Cheese-progress actions can themselves point upward or rightward. We assign these overlaps to the cheese-progress category and reserve *naive only* for directional actions that are not explained by progress toward either candidate target. The naive-only category therefore measures residual directional-shortcut behavior rather than total up-right consistency.

Results. We find that GM behavior in MAZE is organized much more strongly around the proxy-associated destination than around the cheese (Table 8). On conflict states, $\text{Choice}_\star = 0.013$, whereas $\text{Choice}_p = 0.962$. The full action distribution shows the same asymmetry ($\text{Mass}_\star = 0.010$, $\text{Mass}_p = 0.916$), so the result is not confined to the greedy action. Non-GM policies show the opposite organization, with $\text{Choice}_\star = 0.585$ and $\text{Choice}_p = 0.076$ and more probability mass on cheese-progress than on corner-progress actions.

The detour test shows that the GM corner preference is not a fixed up-right shortcut (Fig. 17, left). Across the 168 detour states, GM policies choose corner-progress actions on 0.948 of states, and cheese-progress, naive-only and neutral responses are each rare. Every corner-progress action on these states lies outside the up-right family, so the agent makes progress toward the proxy-associated destination rather than simply continuing upward or rightward. Non-GM policies are far less concentrated. Corner progress falls to 0.173, cheese-progress actions account for 0.498 of states and neutral responses for a further 0.262. Because cheese-progress and up-right actions can overlap, the small naive-only category does not measure total up-right consistency.

Splitting the response to cheese relocation on the shared 128-state sample of the conflict bank into progress toward the relocated cheese, corner progress and other actions gives a total variation of 0.018 for GM against 0.621 for Non-GM, with -0.001 against $+0.357$ of probability moved onto cheese progress, so in GM there is almost nothing to distribute. The ratio of total-variation share to prior mass used for COIN (Appendix B.2.2) is not informative here. GM places 0.94 of its probability on corner progress, so a change with no preferred direction already gives about 5.5 on the small cheese group (observed 7.8), whereas Non-GM reaches 3.55 against its own no-direction value of 1.33.

Together with the earlier cheese-relocation and corner-structure world edits, these results identify a destination-directed organization of GM behavior. The policy is largely insensitive to relocation of the intended goal, responds strongly to changes in the proxy-associated region, and keeps making progress toward that region even when this requires an immediate action outside the naive up-right strategy. The proxy-associated destination therefore functions as an alternative behavioral target rather than merely inducing a fixed directional shortcut. We refer to this regime as *functional target substitution*.

Which corner? The structural corner edit (Appendix B.1.1) carves a path to the top-right interior cell of the full grid, which lies outside the original maze in 80% of states, so it usually creates a new top-right extreme rather than opening the old one. On the states where the first steps toward the old and the new corner differ, GM agents step toward the new corner in 0.909 ± 0.01 of cases and toward the old one in 0.013 ± 0.02 (mean $\pm$ SD across three seeds). Their probability moves off the first step toward the old corner and onto the first step toward the new one, while the Non-GM first step barely changes (Fig. 18d). In greedy rollouts from these states they reach the new corner first in 0.748 ± 0.03 of cases and the old one first in none, although on the states where the edit moves the extreme the same agents reach the old corner in 0.857 ± 0.02 when the maze is left unedited (Fig. 18a–c).

This new extreme is also the top-right cell of the screen. To separate the two, we carve a new extreme only two cells beyond the old one, joined to the maze by a three-cell corridor that does not pass through the old corner, with every carved cell at least three cells from the screen corner. GM agents reach this new extreme first in 0.792 ± 0.05 of rollouts and the old one first in 0.114 ± 0.09, although they reach the old corner in 0.932 ± 0.04 of the same states without the edit. A corridor of the same length carved beyond the bottom-left extreme is never entered (0.000), so newly opened space alone does not attract the agent. The test needs room for the corridor in the margin, so it covers mazes of size at most 17, which is 64% of the states. Mini et al. (2023) open a path to a new top-right extreme to build a steering vector and report qualitatively that it increases the agent's attraction to that corner. The two edits here quantify that observation. The proxy is therefore the top-right extreme of the current maze layout, not a remembered location and not the screen's corner cell. A target defined as the reachable cell nearest the screen corner coincides with this extreme in both edits and is not ruled out. Because the new extreme lies further up and right than the old one in both edits, they do not separate a top-right extreme from continued up-right movement. The detour test above answers that alternative.

B.2.2 COIN: POLICY SHORTCUT

Motivation. The preceding analyses show that the GM COIN policy remains weakly sensitive to coin relocation, but that this influence is much smaller and less consistently goal-directed than in Non-GM policies. The intended goal therefore still affects the policy, but does not appear to organize behavior in the same way as in a goal-seeking agent. Because successful training behavior is strongly associated with continued rightward movement, we ask whether local action selection

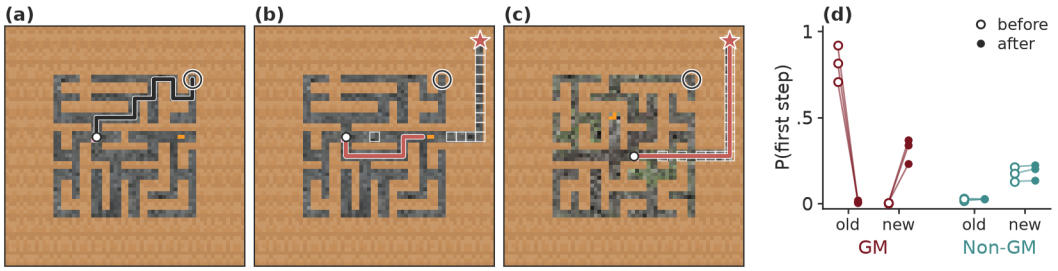

Figure 18: **GM agents go to a new top-right extreme.** (a) Unedited maze for GM checkpoint 1080 at the first state, in level order, where the first steps toward the old and the new corner differ. The greedy rollout from the agent (white dot) reaches the old top-right corner (ring) after 20 steps. (b) The same state after the global corner edit, with carved cells outlined. The agent turns away from the old corner onto the route to the new corner (star) and reaches the cheese, which lies on that route, after 13 steps, which ends the episode. (c) The next such state in level order with the same corridor texture, with no selection on outcome. The agent reaches the new corner after 22 steps, and on the unedited maze it reaches the old corner after 22 steps. (d) Probability of the first step toward the old corner (old) and toward the new corner (new) before (open) and after (filled) the edit, on the same kind of states. Dots are checkpoints and lines join the same checkpoint.

is better aligned with progress toward the current coin or with the training-associated rightward strategy.

Action-set discrimination. Let $A_\star(s) = A_{\text{left}} \subseteq \mathcal{A}_{\text{Coin}}$ denote the left-moving actions, which make immediate progress toward the coin in every evaluated state (Section A.4), and let

$$A_p(s) = A_{\text{right}} \subseteq \mathcal{A}_{\text{Coin}}$$

denote the rightward action family.

As in MAZE, we evaluate the policy on the finite conflict-state set

$$\mathcal{S}_{\text{C}} = \{s : A_\star(s) \neq A_p(s)\},$$

so that progress toward the current coin and continuation of the rightward strategy imply different local action choices. In COIN, $\mathcal{S}_{\text{C}}$ is the 600-state COIN conflict bank, also used for the Fisher analyses. Choice and Mass are read at the edited state s_e of each bank state, in which the coin sits one cell to the left of the agent and the rightward continuation lies to its right. All policies are evaluated on this identical set of states.

Using the definitions from Appendix B.2.1, we compare $\text{Choice}_\star$ and Choice_p, and use $\text{Mass}_\star$ and Mass_p to test whether the same organization is reflected in the full action distribution.

Results. We find that GM behavior in COIN is much more strongly aligned with the rightward strategy than with the current coin location (Table 8 and Fig. 17, right). On conflict states, $\text{Choice}_\star = 0.047$, whereas $\text{Choice}_p = 0.637$. The full policy distribution shows the same asymmetry, with $\text{Mass}_\star = 0.073$ and $\text{Mass}_p = 0.423$. As in MAZE, the agreement between Choice and Mass shows that the result is not confined to the greedy action. The full action distribution is also substantially more aligned with the rightward strategy than with progress toward the coin.

Non-GM policies retain a rightward tendency, but show substantially more coin-directed action selection. Their actions are coin-consistent on 0.318 of conflict states and rightward-consistent on 0.490, with corresponding probability masses of 0.244 and 0.308. The distinction between the two cohorts is therefore not the presence or absence of rightward behavior. Rather, GM policies show a much stronger reduction in coin-directed influence while retaining the training-associated directional tendency.

Together, these results distinguish COIN from the functional target-substitution regime observed in MAZE. The current coin retains a weak influence on action selection, while behavior remains much more strongly organized around the training-associated rightward regularity. Because rightward movement is tightly coupled to progress toward the end of the level in COIN, these experiments do

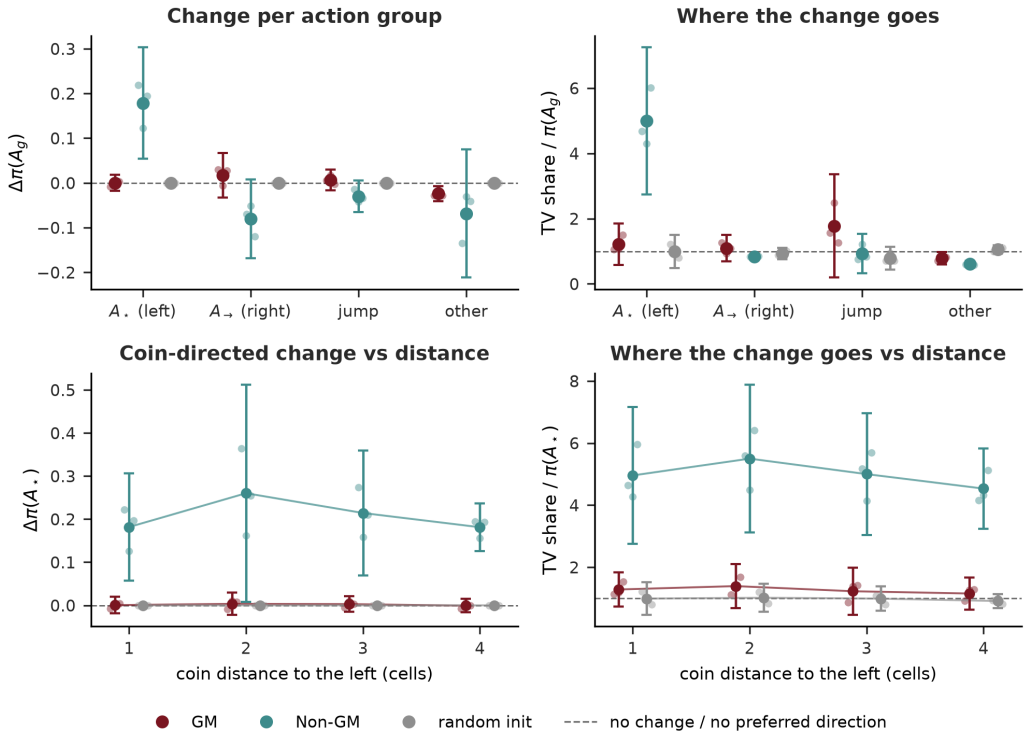

Figure 19: **Where the coin's influence goes in COIN.** Top: change in probability of each action group and its share of the total variation divided by its prior mass, when the coin moves from the cell right of the agent to the cell left of it, on 593 of the 600 conflict-bank states (the other 7 could not be reconstructed exactly). Bottom: the same two quantities for $A_\star$ with the coin 1 to 4 cells left of the agent, on the 572 states where the coin fits and is visible at every distance. Groups are $A_\star$ (LEFT, LEFT+DOWN, LEFT+UP), $A_\rightarrow$ (RIGHT, RIGHT+DOWN, RIGHT+UP), jump (UP) and other (DOWN, no-op and unbound keys). Small dots are checkpoints or random draws, large markers the mean over three, and whiskers 95% t-intervals.

not fully distinguish a literal directional shortcut from endpoint-directed behavior, and Langosco et al. (2022) report both kinds of proxy in CoinRun, a directional one in the actor and a location one in the critic. We therefore use *policy shortcut* (after Geirhos et al., 2020) to describe the observed persistence of the learned rightward strategy without claiming a unique underlying destination representation.

Where the coin's influence goes. Relocating the coin from the cell right of the agent to the cell left of it still changes the GM policy (total variation 0.080 against 0.280 for Non-GM), so we ask which actions absorb this change. For each action group A_g we divide its share of the total variation by the probability $\pi(A_g)$ it held before the edit, so a value of 1 means the change has no preferred direction and lands on each group in proportion to what it already holds. For $A_\star$, we write this ratio as $T_\star$. GM policies move $+0.000$ (95% CI -0.018 to 0.019) of probability onto the coin-directed actions $A_\star$, against $+0.179$ (0.054 to 0.303) for Non-GM, and their ratio on $A_\star$ is 1.21 (0.58 to 1.84), against 5.00 (2.74 to 7.25) for Non-GM and 0.99 (0.48 to 1.51) for three randomly initialized networks on the same states (Fig. 19, top). Drawing logit changes with no preferred direction on each policy's own states gives 1.01 for GM and 0.99 for Non-GM, so 1 is the right reference for both cohorts. When we place the coin 1 to 4 cells left of the agent, GM stays at $\Delta\pi(A_\star)$ between -0.001 and 0.003 with a ratio between 1.15 and 1.39 at every distance, while Non-GM moves 0.18 to 0.26 onto $A_\star$ with a ratio between 4.5 and 5.5 (Fig. 19, bottom). So the coin is registered as a change in the scene, but it redistributes probability across actions in proportion to what they already hold instead of toward the coin, and it is not connected to where the agent goes. In MAZE the GM cheese edit moves almost no probability, and the same ratio is not informative there (Appendix B.2.1).

Removing the end wall. We also test whether GM agents stop at the historical end of the level. On 60 held-out levels we remove the native end wall and extend the floor for 25 further columns, then roll out each checkpoint greedily and record whether an agent that reaches the old wall face continues more than 5 tiles past it and how much of its remaining time it spends at that face. GM agents continue past the former endpoint in 47 of the 50 episodes that reach it (fraction per checkpoint 0.938 ± 0.06) and spend 3% of their post-arrival steps at the former wall face, against 100% with the wall in place. This is consistent with the observation of Langosco et al. (2022) that misgeneralizing CoinRun agents move to the end of the level while ignoring the coin. It rules out stopping at the former endpoint but not running toward a visible end wall, as every continuing agent stops at the next wall, and Non-GM agents rarely reach the wall because they collect the coin first.

B.2.3 ATTRIBUTE MAZE: CUE COMPETITION

Motivation. ATTRIBUTE MAZE is different from MAZE and COIN in one important respect. The behavioral edits above show that GM policies respond not only to the proxy colour cue, but also to changes in the intended shape cue. The native policy therefore remains coupled to both sides of the shape–colour conflict.

This raises a more specific question: *when one cue changes which candidate it supports, how is policy support redistributed between the two competing targets?* A change in relative preference would be consistent with cue competition. A stronger pattern would be push–pull: support increases for the candidate favoured by the edit while simultaneously decreasing for its competitor.

Counter-target competition. We evaluate the same conflict bank of 196 unique strict-conflict states used throughout the ATTRIBUTE MAZE world-edit analyses, with black and textured observations providing matched renderings of the same underlying states. In every state, the line-associated and yellow-associated candidates occupy fixed locations and require disjoint graph-exact progress actions.

For each state we apply the matched shape or colour swap defined in Section B.1.1. The shape swap exchanges the candidates' shape assignments while leaving their colours fixed, while the colour swap exchanges their colours while leaving their shapes fixed. Agent position, maze geometry, candidate locations, and object count are unchanged.

For edit $e \in \{\text{shape}, \text{colour}\}$, let $A^e_{\text{fav}}(s)$ denote the graph-exact progress-action set toward the candidate favoured by the edit and $A^e_{\text{comp}}(s)$ the corresponding set toward the competing candidate. For any action set $A \subseteq \mathcal{A}$, we write

$$\pi(A \mid s) = \sum_{a \in A} \pi(a \mid s)$$

for the policy probability assigned to that action set.

Let s_e denote the matched state after applying edit e (Definition 2). Because policy probabilities are normalized, increasing support for one action group can mechanically reduce probability elsewhere. We therefore compare both candidate action groups with a neutral reference:

$$A^e_0(s) = \mathcal{A} \setminus \left(A^e_{\text{fav}}(s) \cup A^e_{\text{comp}}(s)\right).$$

For the candidate favoured by edit e, define

$$S^e_{\text{fav}}(s) = \log \frac{\pi(A^e_{\text{fav}}(s) \mid s_e)}{\pi(A^e_0(s) \mid s_e)} - \log \frac{\pi(A^e_{\text{fav}}(s) \mid s)}{\pi(A^e_0(s) \mid s)}.$$

Likewise,

$$S^e_{\text{comp}}(s) = \log \frac{\pi(A^e_{\text{comp}}(s) \mid s_e)}{\pi(A^e_0(s) \mid s_e)} - \log \frac{\pi(A^e_{\text{comp}}(s) \mid s)}{\pi(A^e_0(s) \mid s)}.$$

Thus, $S^e_{\text{fav}} > 0$ means that the edit increases support for its favoured candidate relative to neutral actions, while $S^e_{\text{comp}} < 0$ means that it suppresses the competing candidate relative to the same reference. The joint pattern

$$S^e_{\text{fav}} > 0, \qquad S^e_{\text{comp}} < 0$$

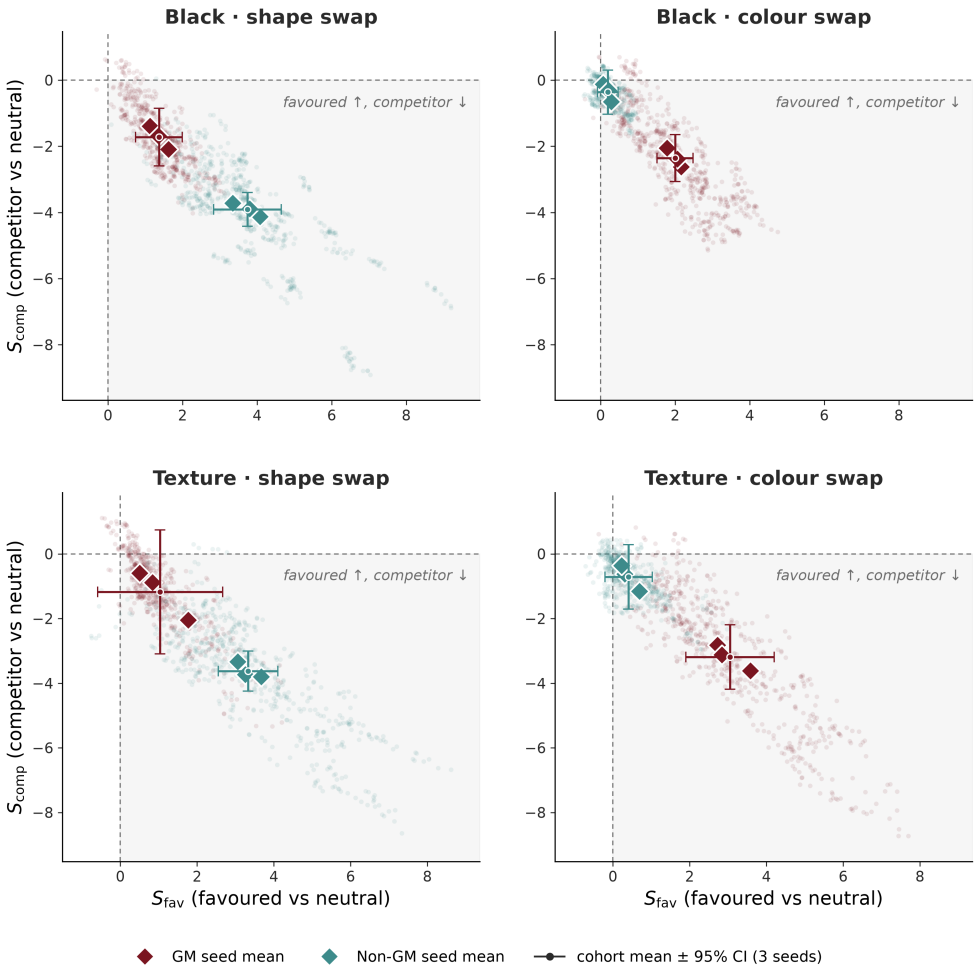

Figure 20: **Matched attribute swaps redistribute native policy support between competing targets.** Small points show individual conflict-bank states. Diamonds show individual seed means and the central marker shows the cohort mean with 95% t-intervals across three seeds. $S_{\text{fav}} > 0$ denotes increased support for the candidate favoured by the swap relative to neutral actions, while $S_{\text{comp}} < 0$ denotes suppression of its competitor. Black and textured cohorts are evaluated on the same 196 conflict-bank states. The qualitative conclusions are unchanged after excluding the 22 states in which both candidates are one step away ($d = 1$).

is the push–pull form of cue competition.

We additionally measure the change in relative preference between the two candidate targets,

$$\Delta\ell^e(s) = \log \frac{\pi(A^e_{\text{fav}}(s) \mid s_e)}{\pi(A^e_{\text{comp}}(s) \mid s_e)} - \log \frac{\pi(A^e_{\text{fav}}(s) \mid s)}{\pi(A^e_{\text{comp}}(s) \mid s)}.$$

By construction,

$$\Delta\ell^e(s) = S^e_{\text{fav}}(s) - S^e_{\text{comp}}(s).$$

A positive value therefore means that the edit shifts relative target preference toward the candidate supported by the edited cue. Unlike the stricter push–pull criterion, this does not require simultaneous enhancement of one candidate and suppression of the other.

Results. The matched swaps systematically redistribute policy support between the two candidate targets (Fig. 20 and Table 9). Every seed shifts relative preference toward the candidate favoured by the swap in every condition (24 of 24 seed means $\Delta\ell > 0$). Across seeds, the 95% interval excludes zero for both GM swaps on black, the GM colour swap on texture and the Non-GM shape swap on both backgrounds. It includes zero for the GM shape swap on texture (2.213, $[-1.33, 5.76]$),

Background	Cohort	Swap	S_{fav}	S_{comp}	$\Delta\ell$
Black	GM	Shape	1.368 ± 0.25	-1.721 ± 0.35	3.089 ± 0.60
	GM	Colour	2.001 ± 0.19	-2.354 ± 0.29	4.355 ± 0.48
	Non-GM	Shape	3.740 ± 0.37	-3.910 ± 0.21	7.650 ± 0.57
	Non-GM	Colour	0.192 ± 0.11	-0.362 ± 0.27	0.555 ± 0.38
Texture	GM	Shape	1.037 ± 0.66	-1.176 ± 0.77	2.213 ± 1.43
	GM	Colour	3.053 ± 0.46	-3.189 ± 0.40	6.242 ± 0.86
	Non-GM	Shape	3.330 ± 0.31	-3.623 ± 0.25	6.953 ± 0.54
	Non-GM	Colour	0.410 ± 0.25	-0.714 ± 0.40	1.123 ± 0.65

Table 9: **Cue competition in ATTRIBUTE MAZE.** Values are mean $\pm$ sample standard deviation across three seeds on the 196-state conflict bank. S_{fav} and S_{comp} measure changes relative to neutral actions, and $\Delta\ell = S_{\text{fav}} - S_{\text{comp}}$ measures the resulting shift in relative target preference.

where one seed dominates, and for the Non-GM colour swap, as expected for a shape-selective policy. On the diversified bank of 480 conflict states, with the agent placed at every walkable cell, push–pull holds for both swaps in every GM seed and for shape only in Non-GM agents, across all 15 action-conflict classes, with effects about half the canonical size in both cohorts (GM black $\Delta\ell = 1.823 \pm 0.24$ for shape and 2.293 ± 0.29 for colour).

The contrast between GM and Non-GM policies mirrors the behavioral dependence results above. Non-GM policies are strongly shape-selective: the shape swap produces a large push–pull response ($\Delta\ell = 7.650$ on black and 6.953 on texture), whereas the colour swap produces a much smaller redistribution (0.555 and 1.123, respectively). Thus changing colour alone has comparatively little control over the native target preference of Non-GM policies.

GM policies respond to both attribute swaps. On black backgrounds, shape and colour swaps produce $\Delta\ell = 3.089$ and 4.355, respectively. On textured backgrounds the corresponding shifts are 2.213 and 6.242. In each case the edit increases support for its favoured candidate and suppresses its competitor on average. The shape response is therefore not absent in GM policies: both the intended shape cue and the proxy colour cue remain coupled to native target selection.

We do not interpret the larger colour-swap response in GM policies as a direct estimate of a larger internal "colour weight." The size of a finite swap also depends on the policy's pre-edit preference and the available headroom for redistribution. The stronger conclusion supported here is that the two matched swaps can each reorganize target preference in GM policies, whereas Non-GM policies are much more selectively controlled by shape.

The same qualitative pattern appears on black and textured backgrounds and is unchanged when restricting to the held-out evaluation states, analysing the two conflict signs separately, or excluding $d = 1$ states.

Finally, these swaps should be interpreted as world edits on the *shape–colour conflict*, rather than as a complete identification of independent cue channels. Each swap resolves the conflict by restoring the familiar yellow-line conjunction at one of the two candidates. What the experiment establishes is that these matched resolutions systematically redistribute native policy support between the competing targets. Isolating independent shape and colour contributions would require additional factorial world edits. Ramanauskas & Şimşek (2023) attribute the colour preference in this environment to goal detection through a single colour channel, and because our swaps also change which channels carry line-shaped evidence, the shape and colour cues we manipulate need not correspond to separate internal features.

The next section asks how this target-level interaction is organized in representation space.

B.3 GEOMETRY OF CUE COMPETITION

We found that the intended shape cue and proxy colour cue both remain active in ATTRIBUTE MAZE, and that changing either cue shifts target preference toward the candidate supported by that cue. On the black-background states, this takes the stronger push–pull form in which the favoured candidate gains support while its competitor is simultaneously suppressed. So, we established cue

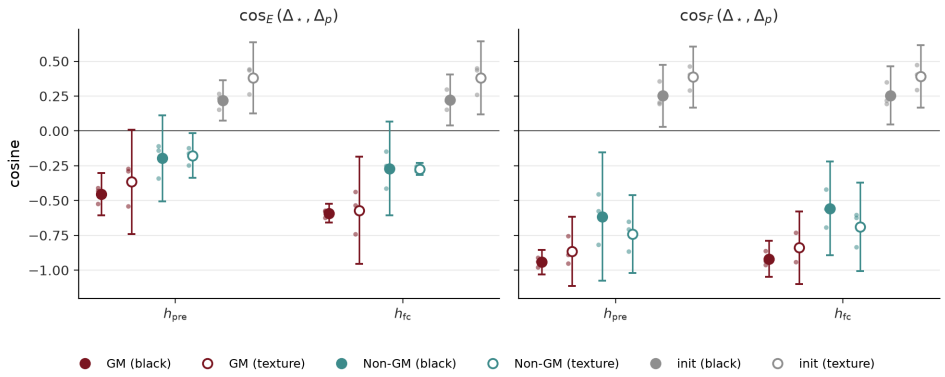

Figure 21: **Within-state cue geometry.** Euclidean (left) and Fisher (right) cosine between the intended-cue displacement $\Delta_\star(s)$ and the proxy-cue displacement $\Delta_p(s)$, averaged over the 196 conflict-bank states. Filled markers show black-background policies and open markers show texture-background policies, each evaluated on its own training background. Grey markers are freshly initialized networks rendered on each background. Small points are individual seeds. Large markers and error bars show the mean and 95% t-interval across three seeds, which can extend beyond $[-1, 1]$ at $n = 3$.

competition at the level of policy behavior. What remains open is what that competition looks like inside the representation.

We ask two questions. First, *when the intended and proxy cues are changed from the same base state, how are the resulting representation displacements related?* Second, *once we define their contrast, is the resulting competition direction reused across states?*

B.3.1 How are the two cue directions organised within a state?

We reuse the same conflict states, graph-exact candidate action sets, and matched cue swaps defined above. The conflict bank contains 196 strict conflict states. For each state $s \in S_{\text{conf}}$, the intended-cue swap induces the representation displacement $\Delta_\star(s)$ and the proxy-cue swap induces $\Delta_p(s)$. Both displacements are measured relative to the same unedited representation. The agent position, maze geometry, candidate locations, and object count remain fixed, while only the relevant cue assignment changes. So this lets us compare how the two cue edits move the representation from the same starting point.

To ask whether the two cue edits induce related representation changes, we first compare the orientation of their displacement vectors. In ordinary activation space, their alignment is measured by the Euclidean cosine

$$\cos_E(\Delta_\star(s), \Delta_p(s)) = \frac{\Delta_\star(s)^\top \Delta_p(s)}{\|\Delta_\star(s)\|_2 \|\Delta_p(s)\|_2}.$$

Positive values indicate aligned representation changes, values near zero indicate weak alignment, and negative values indicate opposing directions.

The Euclidean cosine treats every activation-space direction equally. We are additionally interested in whether the two edits oppose each other specifically along directions to which the downstream policy is sensitive. Because both displacements originate from the same base state, they share the same local Fisher metric. We therefore also compute the Fisher cosine from Definition 7,

$$\cos_F(\Delta_\star(s), \Delta_p(s)).$$

We evaluate both quantities at h_{pre} and h_{fc}. The first gives the final spatial representation before the fully connected transformation, while the second gives the representation supplied downstream after that transformation. Each policy is evaluated on the background it was trained on, and freshly initialized networks serve as the baseline.

Results. Figure 21 and Table 10 show that the two cue edits already push the representation in somewhat opposing directions in ordinary activation space. For GM policies on the black back-

Table 10: **Within-state cue geometry at h_{pre} and h_{fc}.** Entries are cohort mean $\pm$ SD across three policies, each summarized over the 196 conflict-bank states.

Cohort	Background	$\cos_E(h_{\mathrm{pre}})$	$\cos_E(h_{\mathrm{fc}})$	$\cos_F(h_{\mathrm{pre}})$	$\cos_F(h_{\mathrm{fc}})$
GM	black	-0.454 ± 0.06	-0.591 ± 0.03	-0.942 ± 0.04	-0.919 ± 0.05
GM	texture	-0.366 ± 0.15	-0.570 ± 0.15	-0.866 ± 0.10	-0.838 ± 0.11
Non-GM	black	-0.197 ± 0.12	-0.270 ± 0.14	-0.616 ± 0.19	-0.557 ± 0.14
Non-GM	texture	-0.177 ± 0.06	-0.274 ± 0.02	-0.740 ± 0.11	-0.687 ± 0.13
Init	black	0.219 ± 0.06	0.222 ± 0.07	0.252 ± 0.09	0.254 ± 0.08
Init	texture	0.381 ± 0.10	0.382 ± 0.11	0.386 ± 0.09	0.392 ± 0.09

ground, the Euclidean cosine is -0.454 at h_{pre} and -0.591 at h_{fc}. Non-GM policies show the same sign with smaller magnitude.

The Fisher geometry makes this opposition much sharper. For GM policies on the black background, the mean Fisher cosine is -0.942 at h_{pre} and -0.919 at h_{fc}. So, once we only look at the directions the policy reads, the two cue edits are close to antipodal. Non-GM policies are also clearly opposed, with Fisher cosines between -0.56 and -0.74.

Freshly initialized networks behave very differently. Both cosines are positive, at approximately 0.25 on the black background. This has a simple reason. As both swaps change the same two sprites at the same two cells, an untrained network sees a shared "both cells changed" component in the two displacements. Training reverses this relationship in exactly the part of the representation the policy reads, because the two swaps favour opposite candidates.

The opposition does not depend on the swap construction. Single-cue ablations remove one cue instead of exchanging it, and they edit different cells, so their displacements share no sprite change. Under ablations the Euclidean cosine is much weaker, between -0.19 and 0.05 in trained policies, while the Fisher cosine stays clearly negative (for example -0.68 at h_{pre} and -0.70 at h_{fc} for GM on black), and initialized networks sit at approximately zero on both measures. Under ablations, however, GM policies are no more opposed than Non-GM policies. We therefore treat the within-state opposition as a property of trained policies in general, not as a signature of goal misgeneralization.

This motivates a competition direction defined separately for every state,

$$d_{\mathrm{comp}}(s) = \Delta_p(s) - \Delta_\star(s) = h(s_p) - h(s_\star).$$

Because both displacements are measured from the same base representation, the base representation cancels in the difference. The resulting vector is therefore the direct representation-space contrast between the two competing cue assignments at state s. Its positive orientation points from the swap under which both cues support the yellow-gem candidate toward the swap under which both cues support the line candidate. So, pushing along $+d_{\mathrm{comp}}(s)$ moves the representation toward the intended goal.

B.3.2 Is the competition direction shared across states?

The within-state analysis gives us one competition direction $d_{\mathrm{comp}}(s)$ for every conflict state. We now change the unit of comparison and ask how these directions relate across states. In particular, we want to know whether each state uses its own unrelated direction, or whether the competition reuses a simpler structure across different spatial configurations.

Two possible organizations. The simplest possibility is that the same orientation is reused everywhere,

$$d_{\mathrm{comp}}(s) \approx \beta(s) u_{\mathrm{comp}}, \qquad \beta(s) > 0.$$

Here, states may differ in the strength of the competition signal, but not in its direction.

A slightly richer possibility is that the same underlying axis is reused while its orientation changes with the state,

$$d_{\mathrm{comp}}(s) \approx \sigma(s)\beta(s) u_{\mathrm{comp}}, \qquad \sigma(s) \in \{-1, +1\}.$$

Under this account, different states can express opposite orientations of the same competition axis.

Table 11: **Cross-state competition geometry at h_{pre} and h_{fc}.** The first two columns give mean similarity over pairs of states with the same sign σ. The last two give the alignment of held-out competition directions with the shared axis u_{comp} fit on the remaining states. Entries are cohort mean $\pm$ SD across three policies, each evaluated on its training background.

Cohort	Background	Position	κ_E same	κ_F same	$\cos_E(\sigma d_{\mathrm{comp}}, u_{\mathrm{comp}})$	$\cos_F(\sigma d_{\mathrm{comp}}, u_{\mathrm{comp}})$
GM	black	h_{pre}	0.354 ± 0.02	0.938 ± 0.04	0.521 ± 0.02	0.961 ± 0.03
GM	black	h_{fc}	0.589 ± 0.02	0.942 ± 0.05	0.709 ± 0.00	0.959 ± 0.03
GM	texture	h_{pre}	0.265 ± 0.03	0.913 ± 0.06	0.447 ± 0.03	0.943 ± 0.05
GM	texture	h_{fc}	0.588 ± 0.06	0.918 ± 0.07	0.709 ± 0.05	0.946 ± 0.05
Non-GM	black	h_{pre}	0.284 ± 0.02	0.844 ± 0.05	0.468 ± 0.02	0.914 ± 0.05
Non-GM	black	h_{fc}	0.532 ± 0.04	0.952 ± 0.02	0.679 ± 0.05	0.970 ± 0.02
Non-GM	texture	h_{pre}	0.243 ± 0.04	0.820 ± 0.04	0.426 ± 0.04	0.931 ± 0.05
Non-GM	texture	h_{fc}	0.458 ± 0.02	0.900 ± 0.04	0.626 ± 0.02	0.932 ± 0.04
Init	black	h_{pre}	0.173 ± 0.02	0.160 ± 0.03	0.357 ± 0.03	0.330 ± 0.06
Init	black	h_{fc}	0.177 ± 0.03	0.148 ± 0.03	0.364 ± 0.03	0.301 ± 0.06

Comparing competition directions across states. For two states $s_i, s_j \in S_{\mathrm{conf}}$, we first compare their competition directions in ordinary activation space,

$$\kappa_E(s_i, s_j) = \cos_E(d_{\mathrm{comp}}(s_i), d_{\mathrm{comp}}(s_j)) .$$

Positive values mean that the two states use similarly oriented competition directions, values near zero indicate weak alignment, and negative values indicate opposing orientations.

For the policy-sensitive comparison, the two states generally have different local Fisher metrics. We therefore use their pair-average metric

$$F_{ij} = \frac{1}{2}(F_i + F_j)$$

and define

$$\kappa_F(s_i, s_j) = \frac{d_i^\top F_{ij} d_j}{\sqrt{(d_i^\top F_{ij} d_i)(d_j^\top F_{ij} d_j)}},$$

where $d_i = d_{\mathrm{comp}}(s_i)$ and $d_j = d_{\mathrm{comp}}(s_j)$.

So, κ_E and κ_F ask the same cross-state question under two different geometries. The first asks whether the representation directions themselves align. The second asks whether they align once the representation is weighted by the directions that matter to the policy.

What the conflict bank can and cannot tell us. In every conflict-bank state the agent starts at the same cell, $(0, 0)$. From there, the two candidates can only be approached by moving UP or by moving RIGHT. Half of the states place the line candidate along UP and the yellow gem along RIGHT, and the other half reverse this assignment. So, every conflict in the conflict bank is the same UP-versus-RIGHT action contrast, and a signed axis of the second kind is available by design. The important question is therefore not whether such an axis can exist, but whether independently generated cue edits collapse onto it in the part of the representation read by the policy. We first characterize this fixed-conflict geometry on the conflict bank, then use the diversified bank to test whether the resulting axis extends beyond the UP-versus-RIGHT conflict class.

We order the 196 states first by this sign, with 98 states in each group, and then by maze layout and candidate distance. The same ordering is used across the three GM policies, so we can average the pairwise similarity matrices cell by cell. To test whether a single axis transfers to new states, we also fit u_{comp} as the sign-folded mean of d_{comp} over 146 fit states and evaluate its alignment with $\sigma(s)\, d_{\mathrm{comp}}(s)$ on the 50 held-out states.

Results. The independently generated cue edits do collapse onto the signed axis, and this compression is much stronger in the policy-sensitive geometry. The two rows of Figure 22 show this directly.

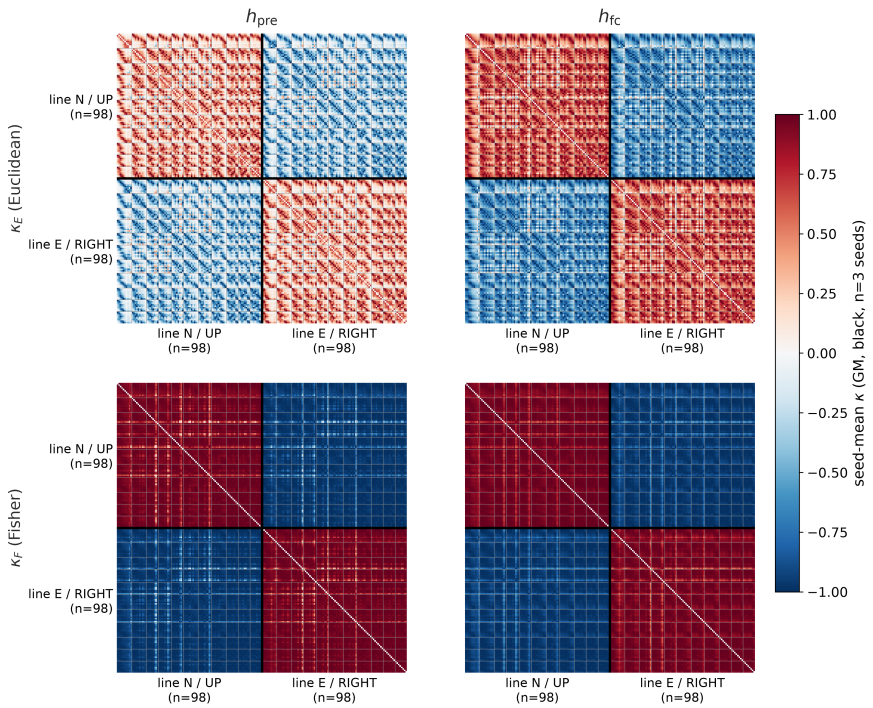

Figure 22: **The policy compresses a rich competition geometry onto one signed axis.** Each matrix shows pairwise similarity between state-local competition directions across the 196 conflict-bank states, averaged over the three GM policies on the black background. Columns show h_{pre} and h_{fc}. The upper row shows Euclidean similarity κ_E and the lower row shows Fisher-weighted similarity κ_F. No spread across the three policies is shown. States are grouped by whether the line candidate lies along UP or along RIGHT, and are ordered by layout and candidate distance within each group. Under the swap construction, mirrored states satisfy $d_{\mathrm{comp}}(\bar{s}) = -d_{\mathrm{comp}}(s)$, so the cross-sign blocks are antisymmetric by construction. The informative structure is therefore the within-sign alignment and its compression under the Fisher metric.

In ordinary activation space, the within-sign blocks are rich and heterogeneous. They carry a fine texture of layout and distance structure, and the mean same-sign similarity is only 0.354 at h_{pre} and 0.589 at h_{fc} for GM policies (Table 11). So, the raw competition vectors of two states with the same action contrast are related but far from identical.

The Fisher geometry removes most of that texture. Same-sign similarity rises to 0.938 at h_{pre} and 0.942 at h_{fc}, compared with 0.160 and 0.148 at initialization. The two diagonal blocks become nearly uniform. So, from the perspective of the policy, the competition direction of each conflict-bank state is almost the same signed vector.

A single axis also transfers to new states. Fit on 146 states, u_{comp} aligns with the held-out competition directions at $\cos_F = 0.96$ in GM policies on the black background, and the sign $\sigma(s)$ is recovered for every held-out state. In Euclidean terms the same axis reaches only 0.52 at h_{pre} and 0.71 at h_{fc}. Between 74% and 86% of the Fisher energy of d_{comp} lies on a single logit contrast, the difference between the UP action and the RIGHT action group, compared with approximately 8% at initialization. The conflict-bank geometry is therefore well described by the second organization, $d_{\mathrm{comp}}(s) \approx \sigma(s)\beta(s)u_{\mathrm{comp}}$. The state mainly chooses the sign and the magnitude of one shared action contrast. Interestingly, the same compression appears in Non-GM policies, where same-sign Fisher similarity reaches 0.952 at h_{fc}. The axis itself is therefore not specific to goal misgeneralization.

What the rich Euclidean structure encodes. The heterogeneous Euclidean texture raises the question of whether the policy also reads a finer spatial code within each sign. For each state we describe the spatial configuration of the two candidates by their first shortest-path directions and graph distances from the agent, and we compare pairwise spatial similarity with κ_E and κ_F within

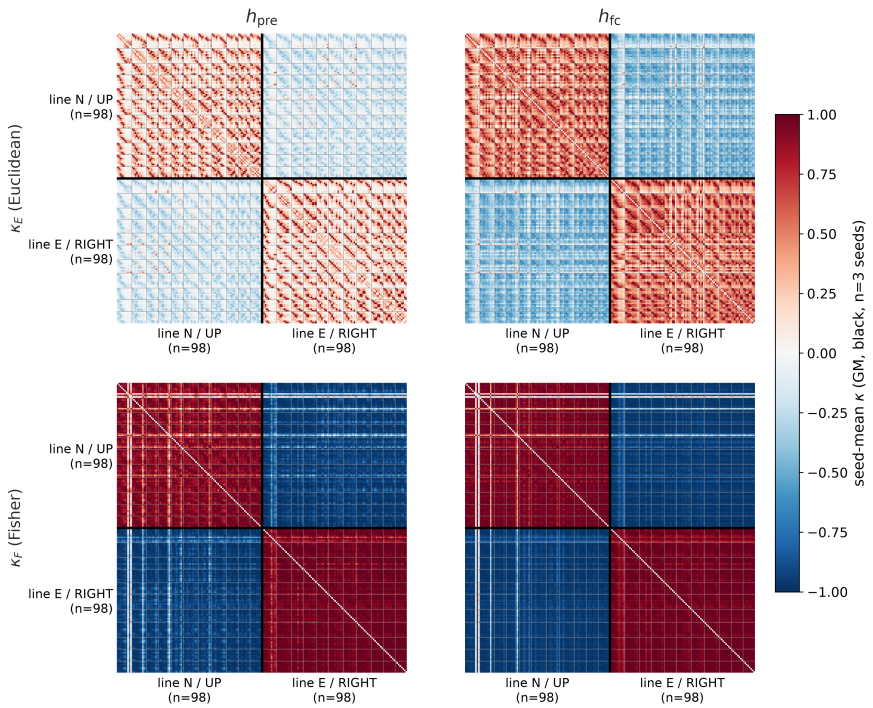

Figure 23: **Ablation-based competition geometry.** Same layout as Figure 22, with competition directions built from single-cue ablations instead of swaps, again averaged over the three GM policies on the black background with no spread shown. Ablations do not satisfy the mirror identity, so the cross-sign blocks are an empirical result here. At h_{pre} the Euclidean cross-sign blocks are close to zero, while the Fisher cross-sign blocks are strongly negative.

same-sign pairs. Spatial similarity tracks κ_E strongly, with correlations of 0.873 at h_{pre} and 0.735 at h_{fc} in GM policies on the black background. It tracks κ_F only moderately, at 0.296 and 0.460. Freshly initialized networks show the same pattern, with correlations of 0.737 and 0.733 for κ_E and 0.502 and 0.461 for κ_F. So, the fine spatial texture reflects the similarity of the inputs rather than a learned spatial code, and it lies mostly in directions the policy does not read.

Low-rank Fisher control. The policy has 15 actions, so the local Fisher metric has rank at most 14, and a low-rank metric can itself make cosine similarities more extreme. A matched random-vector control, evaluated under each pair's metric, confirms that Fisher low rank increases absolute cosine values but does not explain the observed alignment. For GM policies at h_{fc}, mean absolute Fisher similarity is 0.942 compared with 0.428 under the null, and 82% of real pairs exceed 0.9 compared with 2% of null pairs. The same separation holds in every trained cohort at both positions.

The axis is not an artefact of the swap construction. Under swaps, the antisymmetry between mirrored states is algebraic, so it cannot by itself show that the policy reads the two signs as opposite. We therefore project the competition directions obtained from single-cue ablations onto the swap axis u_{comp}, again evaluating only the 50 held-out states. Ablations do not inherit the mirror identity. Nevertheless, 92% of the Fisher energy of the ablation competition directions lies on the swap axis in GM policies on the black background, and between 82% and 90% in the other trained cohorts, compared with 9% to 10% at initialization. The sign agrees with $\sigma(s)$ in 92% to 100% of held-out states, and the ablation and swap axes coincide under the Fisher metric, with $\cos_F > 0.99$ in every trained cohort. The weakest cohort is GM on the texture background, where one of the three policies places less than 70% of the energy on the swap axis. The corresponding ablation heatmap does not inherit the algebraic antisymmetry and independently recovers the same policy-sensitive sign structure (Figure 23). In that heatmap the Euclidean cross-sign blocks at h_{pre} are close to zero, while the Fisher cross-sign blocks are close to -0.9. The two sides of the conflict are almost unrelated in raw activations, and only the policy metric makes them opposite.

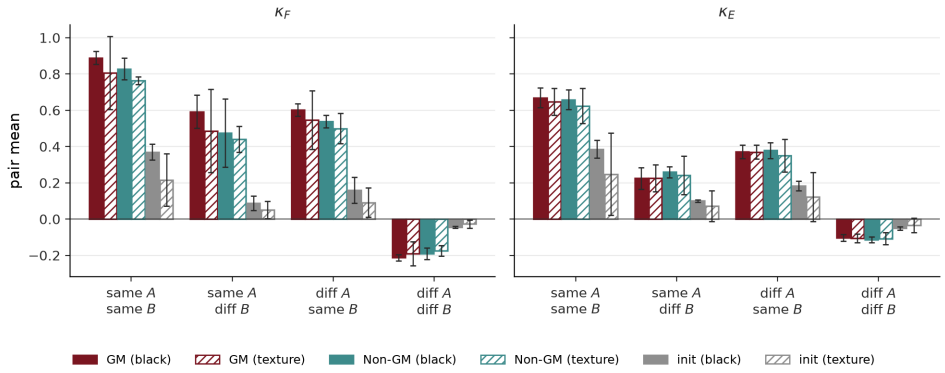

Figure 24: **With varied agent positions, competition directions are shared within action-conflict classes, not globally.** Mean pairwise Fisher (left) and Euclidean (right) similarity of competition directions at h_{fc} on the diversified bank of 480 strict conflict states. Pairs are grouped by whether they share the ordered action contrast A and the bearing B of the two targets. Filled bars show black-background policies and hatched bars show texture-background policies. Grey bars are freshly initialized networks. Error bars show 95% t-intervals across three seeds.

Scope: one axis or one axis per conflict? The fixed-spawn bank cannot tell us whether this is one universal goal-versus-proxy axis or simply the axis for the UP-versus-RIGHT conflict. Moving the agent breaks that confound. We therefore use the diversified bank of 480 strict conflict states (Appendix E), in which the agent is placed at all 21 walkable cells, so that every pair of movement directions occurs as a conflict. Within-state opposition persists there, with $\cos_F = -0.75$ at h_{fc} for GM policies on the black background.

We then group pairs of states by whether they share the ordered action contrast A and whether they share the bearing B of the two targets (Figure 24). Pairs that share both are strongly aligned in the Fisher geometry, with $\kappa_F = 0.889$ at h_{fc} in GM policies on the black background, compared with 0.370 at initialization. Pairs that share neither are weakly anti-aligned, with $\kappa_F = -0.213$. Pairs that share only one of the two lie in between, at approximately 0.6. An axis fit separately for each action-pair class transfers to held-out states of that class at $\cos_F = 0.83$, compared with 0.28 at initialization. So, the single shared axis of the conflict bank is the special case of a bank that contains only one conflict class. With varied agent positions, the policy-read competition geometry forms roughly one shared axis per action-conflict class, rather than a single global axis. Agent-relative bearing adds little beyond this. Once action contrast and absolute target configuration are controlled, the unique variance in held-out κ_F that bearing explains beyond initialization is only 0.028 ± 0.04 for GM policies on the black background. This scope check is lower-confidence than the conflict-bank analysis because it uses three seeds, many states are off-policy for the GM agents, and steering was not evaluated on this diversified population.

Implication for causal control. These results give two causal predictions. A state-specific push along $d_{\text{comp}}(s)$ should alter which cue controls the policy without assuming any shared geometry. Within a fixed action-conflict class, the compressed shared axis should provide a lower-dimensional route to the same intervention. We test both next.

B.3.3 WHY DOES THE PROXY WIN ON THE SHARED AXIS?

The previous section showed that the policy reads the cue competition along one signed axis u_{comp}, the same axis we later steer along (Appendix C.1.2). If both cues act on this one axis, the competition is a tug of war, and we can ask which cue pulls harder and whether this decides the agent's choice.

Measuring each cue's pull. Each swap moves one cue to the other candidate, so the displacement it causes is exactly the influence that cue had on the conflict state. To express this influence in the currency the policy uses, we project each swap displacement onto u_{comp} in the Fisher metric of the unedited state, as in Appendix A.3,

$$\alpha_F(\Delta) = \frac{u_{\text{comp}}^\top F(s)\,\Delta}{u_{\text{comp}}^\top F(s)\,u_{\text{comp}}}, \qquad \Delta \in \{\Delta_\star(s), \Delta_p(s)\}.$$

The projection keeps only the part of the displacement that lies along the competition axis, and the Fisher weighting counts it by how much it changes the policy. So $\alpha_F(\Delta_\star)$ measures how far the goal cue pulls the state along the axis, and $\alpha_F(\Delta_p)$ how far the proxy cue pulls it, both oriented so that a positive value points toward the candidate the swap favours. The axis is fit on the 146 fit states, and all pulls are reported on the 50 held-out states at h_{fc} on the black background.

The proxy pulls harder in GM policies. We find that in GM policies the colour swap moves the representation along the axis by 9.321 and the shape swap by 5.974. The ratio of the two is 1.564 (95% CI $[1.34, 1.79]$) and exceeds 1 in every seed, so the proxy clearly dominates the goal on the axis that decides the choice. Control policies show the mirror image, with a shape pull of 17.291, a colour pull of 1.012 and a ratio of 0.067. They are pulled almost only by the goal cue, while the proxy cue is nearly inert. Figure 25a shows that this holds state by state and not only on average: the proxy pull is the larger one in 79% of GM states and in none of the control states. The Euclidean projection gives the same ordering (1.631 against 0.094), although the Fisher projection is the one that measures relevance for the policy.

Where the unedited state sits. If the proxy pulls harder, the unedited conflict state should sit closer to the proxy's side of the axis, and its position should tell us what the agent does. The two swaps give us both ends of the axis for free. The shape swap puts both cues on the proxy's candidate (the proxy consensus), and the colour swap puts both cues on the goal's candidate (the goal consensus). The unedited state lies between these two consensus states, and the two pulls are its distances to them. Its position is therefore

$$t(s) = \frac{\alpha_F(\Delta_\star)}{\alpha_F(\Delta_\star) + \alpha_F(\Delta_p)},$$

with $t = 0$ at the proxy consensus and $t = 1$ at the goal consensus. The median held-out state sits at $t = 0.396$ in GM policies and at $t = 0.933$ in control policies. So GM conflict states lie on the proxy side of the midpoint, and control states lie close to the goal consensus.

This position also predicts what the GM agent does from each state. In Figure 25b, the fraction of GM rollouts that reach the intended goal stays below 0.2 in the bins on the proxy side and reaches 0.57 in the bin closest to the goal side. As a single number, the signed distance of the base state from the midpoint predicts whether a GM rollout reaches the goal or the proxy with an AUC of 0.774 over all 196 states (0.846 within each conflict orientation, timeouts excluded). We hold this at moderate confidence, because the held-out states contain only 5 to 11 goal outcomes per seed. Control policies never reach the proxy in these rollouts, so there is no outcome to predict for them.

The proxy pull decides. We can also ask which of the two pulls makes the difference between states (Figure 25c). The proxy pull alone predicts the GM outcome as well as both pulls together (0.779), while the goal pull alone is at chance (0.459). So the goal pulls with roughly the same strength in every state, and the states in which the GM agent still reaches the goal are the ones in which the colour cue happens to pull weakly.

We want to be clear about what this analysis adds. It is not a separate line of evidence but a confirmation of the swap results of Table 9, seen from the policy's own axis, where both swap effects are expressed in one shared unit. What it adds is that the two cues compete on one scale and that the

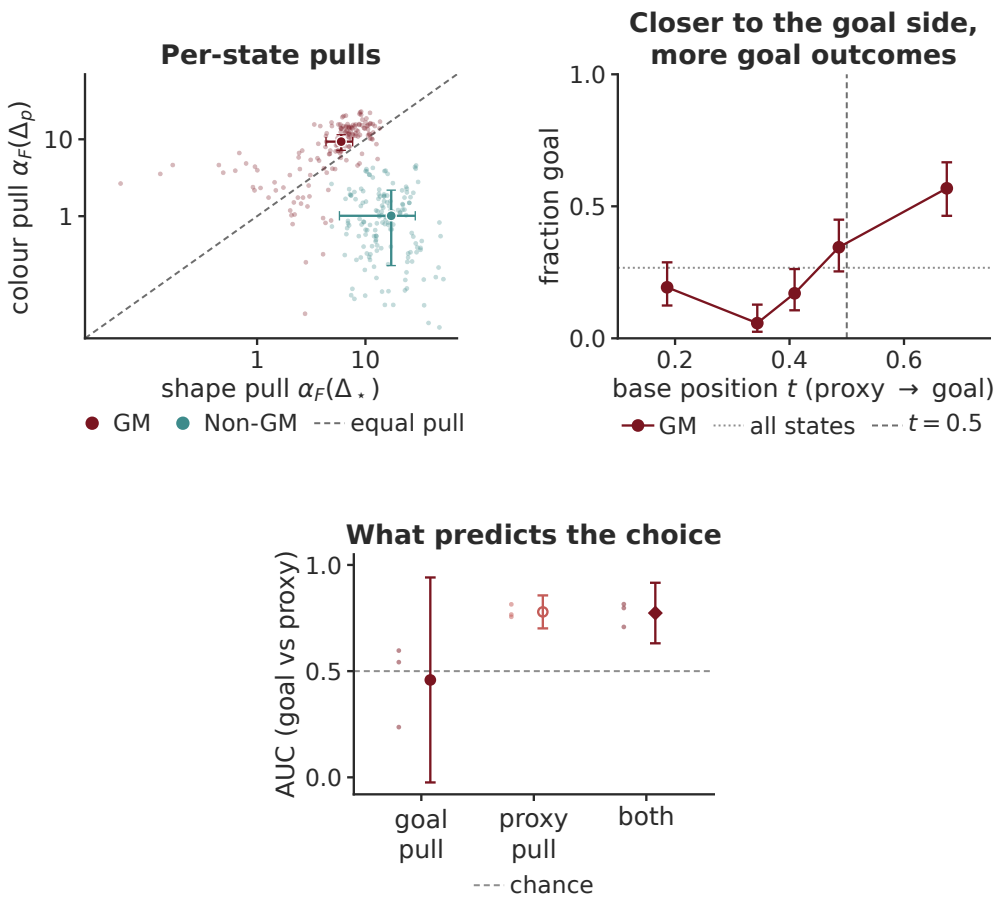

Figure 25: **GM policies weight the proxy cue more strongly on the shared competition axis** (h_{fc}, black background). (a) Goal pull against proxy pull per held-out state (small points) with cohort means and 95% t-intervals. States above the identity line are pulled harder by the proxy. Axes are symmetric-logarithmic. (b) Fraction of GM rollouts reaching the goal in five quantile bins of the base position t, with Wilson 95% intervals (all 196 states, timeouts excluded). (c) AUC for predicting a goal versus a proxy outcome from the goal pull, the proxy pull, or both.

balance on this scale predicts the choice, although it does not explain why training gives the colour cue about 1.5 times the pull of the shape cue. Together with the previous sections, the geometry of cue competition is therefore simple. Within a state the two cues pull in opposite directions, across states the policy reads this opposition as one signed axis per conflict, and on that axis the proxy pulls harder than the goal in GM policies. The goal is not lost in these agents but outweighed, which makes the shared axis a natural place to intervene (Appendix C.1).

C RECOVERABILITY: CAN GOAL-DIRECTED BEHAVIOR BE RECOVERED?

C.1 STEERING THE EXISTING WIRING

The preceding analyses identify internal goal-related directions, but geometry alone does not establish that these directions control behavior. A representation displacement can be decodable, policy-sensitive, or geometrically well organized without providing a useful intervention direction. We therefore ask whether the internal structure identified above can be used to redirect the existing policy without retraining its weights. This is deliberately a question about the wiring the policy already has. Rewiring, which we turn to next, instead asks what becomes possible once the downstream computation is relearned.

The geometry suggests two increasingly specific causal questions. *First, can the state-specific counterfactual direction available at the current state causally redirect behavior? Second, in the environment where this local intervention succeeds, how much of that causal effect survives compression into the shared competition axis identified across states?*

All interventions are applied at h_{fc}, after the final ReLU. This is the representation the policy and value heads read directly, so a push there changes the input to the head without passing through any further nonlinearity. The counterfactual direction used for steering is reconstructed at the same representation at which it is injected. In ATTRIBUTE MAZE, pushing at h_{pre} instead gives the same reach at every dose up to $\beta = 1$, so the choice of site does not drive the results below.

C.1.1 CAN THE STATE-LOCAL COUNTERFACTUAL DIRECTION REDIRECT THE POLICY?

The intervention. The most direct causal use of the counterfactual analyses is to retain the complete state-dependent direction. This makes no assumption that the relevant geometry is shared across states or low-dimensional. At each rollout state s_t, we reconstruct the counterfactual representations associated with the intended and proxy-supported alternatives and use their contrast as the intervention direction $d_{\mathrm{local}}(s_t)$. Its positive orientation always points toward the counterfactual in which the evidence supports the intended target. In ATTRIBUTE MAZE, this is exactly the competition direction defined above,

$$d_{\mathrm{local}}(s_t) = d_{\mathrm{comp}}(s_t) = \Delta_p(s_t) - \Delta_\star(s_t).$$

In MAZE, the two counterfactuals place the cheese at its own cell and in the top-right corner, $d_{\mathrm{local}}(s_t) = h_{\mathrm{fc}}(s_t^\star) - h_{\mathrm{fc}}(s_t^p)$. In COIN, they place the coin at its position and on the shortcut side, ahead of the agent, and rollouts start from the edited conflict state in which the coin lies behind the agent, off the shortcut path. The important commonality is that the intervention is reconstructed from the current state at every step rather than replaced by one fixed direction learned across states.

For steering dose β, the corrective intervention, a state-local form of activation addition (Turner et al., 2023b), is

$$\delta^+_{\mathrm{local}}(s_t;\beta) = \beta\, d_{\mathrm{local}}(s_t),$$

giving

$$h'_{\mathrm{fc}}(s_t) = h_{\mathrm{fc}}(s_t) + \delta^+_{\mathrm{local}}(s_t;\beta).$$

The anticorrective intervention reverses the same direction,

$$\delta^-_{\mathrm{local}}(s_t;\beta) = -\beta\, d_{\mathrm{local}}(s_t).$$

This sign reversal is important. A direction that merely perturbs the policy may occasionally improve goal reach without being a meaningful causal axis. A genuine target-selection direction should instead have an ordered effect. Moving along the corrective orientation should favour intended-goal behavior, while reversing the same intervention should move behavior in the opposite direction.

For sufficiently small interventions, this follows the local first-order prediction

$$C_s(h+\delta) - C_s(h) \approx \nabla_h C_s(h)^\top \delta,$$

where C_s is the target-selection contrast defined above. The rollout experiment asks the stronger question of whether this local effect remains useful after repeatedly intervening as the state changes.

Table 12: **State-local steering at $\beta = 1$.** Intended-target reach, mean $\pm$ SD across three checkpoints, on the states of Figure 26. Each entry is read against the unsteered baseline in its own row, not across cohorts.

Environment	Cohort	Baseline	Corrective	Anticorrective	Random
ATTRIBUTE MAZE (black)	GM	0.160 ± 0.06	0.953 ± 0.05	0.007 ± 0.01	0.171 ± 0.06
	Non-GM	0.813 ± 0.05	0.587 ± 0.06	0.000 ± 0.00	0.504 ± 0.05
ATTRIBUTE MAZE (texture)	GM	0.040 ± 0.07	0.840 ± 0.18	0.000 ± 0.00	0.049 ± 0.07
	Non-GM	0.567 ± 0.13	0.507 ± 0.05	0.000 ± 0.00	0.418 ± 0.07
MAZE	GM	0.001 ± 0.00	0.002 ± 0.00	0.000 ± 0.00	0.001 ± 0.00
	Non-GM	0.567 ± 0.03	0.569 ± 0.03	0.002 ± 0.00	0.340 ± 0.04
COIN	GM	0.018 ± 0.01	0.016 ± 0.01	0.020 ± 0.01	0.020 ± 0.01
	Non-GM	0.121 ± 0.05	0.259 ± 0.08	0.019 ± 0.01	0.171 ± 0.07

Controls and dose response. We compare four conditions,

baseline, matched random, corrective, anticorrective.

The baseline leaves the representation unchanged. The random control injects three random directions with norm matched to the corresponding treatment intervention. It therefore controls for the possibility that sufficiently large perturbations alter behavior regardless of their orientation. The corrective and anticorrective conditions use opposite signs of the same state-local direction. Together with the random control, this separates directional causal control from generic intervention effects.

We evaluate the dose grid $\beta \in \{0.125,\ 0.25,\ 0.5,\ 1.0\}$ in MAZE and ATTRIBUTE MAZE, extend it to $\beta \in \{0.75,\ 1.25,\ 1.5,\ 2.0\}$ in ATTRIBUTE MAZE, and use $\beta \in \{0.25,\ 0.5,\ 1,\ 2,\ 4\}$ in COIN, where the direction is smallest in norm. This lets us distinguish a stable dose-response relationship from an effect that occurs only at one arbitrary intervention magnitude.

Evaluation states. Every steering state is a fixed conflict state in which the intended and proxy targets require different behavior, and the same states are used for every checkpoint of both cohorts. Steering trains nothing, so no condition is fit to these states. The one fitted quantity, the shared axis below, is estimated on states disjoint from the evaluation states. In ATTRIBUTE MAZE we report the 50 evaluation states of the conflict bank, the same states on which the shared axis below is tested. In MAZE we use 320 decision-fork states, the first fork of 160 GM and 160 Non-GM greedy rollouts, shared by every checkpoint. In COIN we use the states of the conflict bank, chosen by level geometry alone, with 300 reached by rollouts of each cohort (Appendix E). Each rollout starts from the edited conflict state, with the coin at the bank's goal placement left of the agent and no coin on the shortcut side. At every step, $d_{\text{local}}(s_t)$ is h_{fc} with the coin at its current cell minus h_{fc} with the coin moved to the nearest free cell ahead of the agent, so the corrective push points toward the coin the agent has to turn back for. Reach counts rollouts that collect the coin. The coin to be collected starts on the shortcut side, so at the start state reaching it does not by itself separate the coin from the rightward shortcut.

Outcomes. The primary outcome is intended-target reach. Where an episode can also end at the proxy target, reach factorizes exactly into decisiveness and conditional preference,

$$\text{reach} = D \cdot P_{\text{goal|dec}}, \qquad D = \frac{N_{\text{goal}} + N_{\text{proxy}}}{N}.$$

D asks whether the policy still completes the task at all, and $P_{\text{goal|dec}}$ asks which target it chooses when it does. This lets us tell a change in target selection apart from a policy that has simply been pushed into timeouts. The factorization applies in ATTRIBUTE MAZE and COIN. In MAZE, the episode ends only at the cheese, so reach is the only outcome.

Results in ATTRIBUTE MAZE. The local competition direction is a strong causal handle on the GM policy (Figure 26, Table 12). On the black background, intended-goal reach rises from 0.160 without steering to 0.953 at $\beta = 1$. On the textured background it rises from 0.040 to 0.840. This was not guaranteed by the construction. A direction built from counterfactual edits at one state could have shifted the first action and then washed out, broken competent navigation, or worked only on

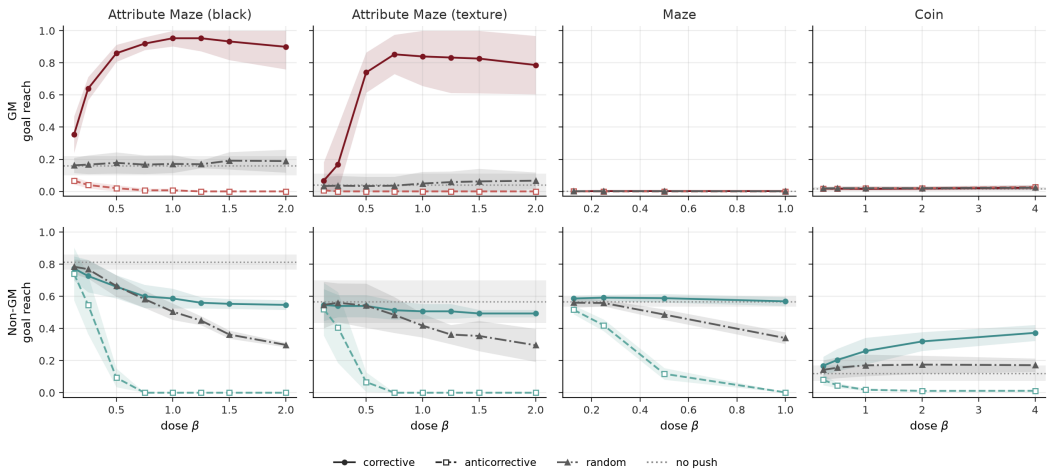

Figure 26: **State-local steering across environments.** Intended-target reach under corrective, anticorrective, and norm-matched random pushes along the state-local counterfactual direction at h_{fc}, as a function of dose β. The upper row shows GM policies and the lower row Non-GM policies. The dotted line and grey band show the unsteered baseline. Lines show the mean and shaded bands $\pm$ SD across three checkpoints. ATTRIBUTE MAZE panels use the 50 conflict-bank evaluation states, MAZE panels the 320 decision-fork states, and COIN panels 591 conflict-bank states.

a narrow subset of states. Instead it survives closed-loop rollout across the evaluation states. Norm-matched random pushes leave reach at its baseline across the full dose range, and the anticorrective push drives it to zero. So, the effect is directional and ordered, not a consequence of perturbing the representation.

The decomposition shows what the push changes. For GM policies on the black background, the conditional preference $P_{\text{goal}|\text{dec}}$ rises from 0.235 to 0.993 at $\beta = 1$, while decisiveness also rises, from 0.687 to 0.960. On the textured background, conditional preference rises from 0.069 to 1.000. Steering therefore does not merely make the agent more active. It changes which competing target wins. The texture policies need a larger dose before this happens, with reach of only 0.167 ± 0.24 at $\beta = 0.25$ before rising to 0.740 at $\beta = 0.5$.

Non-GM policies behave differently, and we treat them as a positive control for the direction rather than as a second recovery target. They already choose the intended target whenever they complete the task, with $P_{\text{goal}|\text{dec}} = 1.00$ at baseline on both backgrounds, and this stays at 1.00 under every corrective dose. Their reach nevertheless falls, from 0.813 to 0.587 on the black background, because a large push into an already working policy costs decisiveness and produces timeouts. The anticorrective push, by contrast, removes intended-goal reach entirely, so the same direction still carries target selection in these policies.

Results in MAZE. In MAZE, the intervention is asymmetric. No condition moves the GM policy, which reaches the cheese in 0.001 of the 320 states without steering and in at most 0.002 under any push at any dose. This is not primarily a matter of push size. The GM direction is about thirteen times smaller in norm than the Non-GM direction at the same β (median $\|d_{\text{local}}(s_0)\|_2$ of 1.40 against 17.97), so a separate run pushes GM along the same direction at fixed norms, on the first 48 of these states. Only very large pushes change behavior at all: at fixed norms of 16–32, both corrective and anticorrective pushes occasionally reach the cheese (at most 7 of 48 states for one seed), so these rare successes reflect large perturbations rather than a goal direction. In Non-GM policies, the same construction is clearly causal in the other direction. The anticorrective push removes cheese pursuit almost completely, from 0.567 to 0.002 at $\beta = 1$, while random pushes of equal norm only reduce it to 0.340. The corrective push leaves Non-GM reach near its baseline. So, removing the cheese-associated direction from a policy that already uses the cheese destroys goal pursuit, but adding the same kind of direction to GM does not create cheese pursuit. The intervention can suppress an existing computation, but it cannot instantiate one that the native GM policy does not already use. This matches the MAZE Fisher analysis, computed on a separate set of decision

states, where the policy-sensitive energy of the GM cheese edit is roughly three orders of magnitude below that of Non-GM policies and its signed effect on target selection is indistinguishable from zero. This asymmetry is consistent with earlier interventions on MAZE policies, where subtracting a cheese vector suppressed cheese-seeking while adding it left cheese-seeking unchanged, and editing goal-tracking channels retargeted the policy only partially (Mini et al., 2023). Those policies were trained with the cheese anywhere in the top-right 5×5 region and pursued it at baseline, whereas our GM agents are trained with the cheese on the exact corner cell (`rand_region`$= 0$) and our Non-GM agents with a randomized position (`rand_region`$= 25$), so the agreement is in the direction of the effect rather than a replication of our setting.

Results in COIN. Rollouts start from the edited conflict state, in which the coin lies behind the agent and off the shortcut path, on 591 of the 600 conflict-bank states. The remaining 9 states are excluded before any rollout because replaying the bank harvest does not reproduce them, and the same states are excluded for every checkpoint. In GM policies, no push changes coin reach. It is 0.018 without steering and stays between 0.016 and 0.027 under corrective, anticorrective and random pushes at every dose up to $\beta = 4$. At $\beta = 4$, the corrective push exceeds the random push by -0.001 (95% CI ± 0.013 across checkpoints). Target choice is equally flat, with the conditional preference $P_{\text{goal}|\text{dec}}$ at 0.028 without steering and 0.035 under the corrective push at $\beta = 4$. Because the GM baseline is close to zero, the anticorrective push cannot show a decrease, so the evidence for GM rests on the corrective push matching the random push. This is not a matter of push size. The GM direction is about four times smaller in norm than the Non-GM direction (mean $\|d_{\text{local}}(s_0)\|_2$ of 1.93 against 7.85), so at $\beta = 4$ the GM push has about the norm of the Non-GM push at $\beta = 1$, where Non-GM policies already respond. This matches the COIN Fisher analysis on the same states, where the GM signed response to relocating the coin is indistinguishable from zero.

In Non-GM policies, the same construction is causal in both directions and grows with dose. The corrective push raises reach from 0.121 to 0.259 at $\beta = 1$ and 0.373 at $\beta = 4$, against 0.171 for random pushes, and the anticorrective push lowers it to 0.019 and 0.012. Conditional preference rises from 0.263 to 0.733 at $\beta = 4$ and falls to 0.025 under the anticorrective push, while decisiveness stays near its baseline of 0.450. All three Non-GM checkpoints are monotone in both directions. So, in COIN the direction is causally live where the policy reads it, and inert in GM policies.

C.1.2 HOW MUCH CONTROL SURVIVES COMPRESSION ONTO A SHARED AXIS?

Motivation. State-local steering retains the full counterfactual direction $d_{\text{comp}}(s)$ at every state. It therefore provides the strongest test of causal accessibility, but it does not tell us how much of the intervention is actually necessary.

The geometry analysis gives us a more specific hypothesis in ATTRIBUTE MAZE. Within the canonical UP-versus-RIGHT conflict class, the policy-relevant component of $d_{\text{comp}}(s)$ is well described by one signed axis, $d_{\text{comp}}(s) \approx \sigma(s)\beta(s)u_{\text{comp}}$ (Section B.3.2). If the successful local intervention acts primarily through this shared axis, then much of its causal effect should survive after the remaining state-specific structure is removed. This test is specific to ATTRIBUTE MAZE, the only environment where local steering succeeds and where the cross-state geometry supports a shared axis.

Fitting the shared axis. We fit the axis only on the 146 fit states of the conflict bank, which are disjoint from the 50 evaluation states used for steering, with mirrored states always kept on the same side of the split. Within the fit states, 118 determine the axis and 28 serve as a held-out check before any steering. The axis is the sign-folded mean of the competition directions,

$$u_{\text{comp}} = \frac{\bar{d}_+ - \bar{d}_-}{\|\bar{d}_+ - \bar{d}_-\|_2}, \qquad \bar{d}_\pm = \frac{1}{|S_\pm|} \sum_{s \in S_\pm} d_{\text{comp}}(s),$$

where S_+ contains the fit states in which the line candidate lies along UP and S_- those in which it lies along RIGHT. The sign $\sigma(s) \in \{-1, +1\}$ used at a steering state is determined only by this orientation of its conflict. The rollout outcomes used to evaluate steering do not enter this construction.

Table 13: **Compression test at $\beta = 0.25$.** Intended-goal reach of GM policies on the 50 conflict-bank evaluation states, mean $\pm$ SD across three checkpoints. The unsteered baselines are 0.160 (black) and 0.040 (texture). Bold entries mark the three fixed pushes whose equivalence carries the compression result. The texture row is shown in grey because at this dose all arms remain close to baseline and the variation across checkpoints is as large as the effects.

Background	Local, fixed at s_0	Sign-gated	Fisher-projected	Unsigned	Unsigned, reversed	Local, rebuilt
Black	$\mathbf{0.560 \pm 0.05}$	$\mathbf{0.587 \pm 0.11}$	$\mathbf{0.560 \pm 0.04}$	0.300 ± 0.08	0.307 ± 0.14	0.640 ± 0.07
Texture	0.173 ± 0.27	0.200 ± 0.25	0.180 ± 0.29	0.093 ± 0.09	0.107 ± 0.15	0.167 ± 0.24

Intervention hierarchy. We compare the complete local intervention with progressively more compressed versions of it. Every compressed intervention is a fixed push, computed once at the start state s_0 and held constant during the rollout. The fair reference for compression is therefore the local direction fixed in the same way,

$$\delta_{\text{fixed}}(\beta) = \beta\, d_{\text{comp}}(s_0).$$

The fully local intervention, which rebuilds $d_{\text{comp}}(s_t)$ at every step, remains the positive control.

The sign-gated shared-axis intervention removes the state-specific direction while preserving its local magnitude and geometry-derived orientation,

$$\delta_{\text{shared}}(\beta) = \beta\, \sigma(s_0)\, \|d_{\text{comp}}(s_0)\|_2\, \hat{u}_{\text{comp}}.$$

This is the central compression hypothesis. If it approaches the effect of δ_{fixed}, then the causal component of $d_{\text{comp}}(s_0)$ is largely captured by one shared direction together with its sign.

We then remove the sign gate,

$$\delta_{\text{unsigned}}(\beta) = \beta\, \|d_{\text{comp}}(s_0)\|_2\, \hat{u}_{\text{comp}}.$$

This preserves the shared axis and the intervention magnitude but uses the same orientation in every state. It therefore tests whether the sign identified by the geometry is causally necessary.

Finally, we retain only the component of the local direction that the policy-sensitive geometry assigns to the shared axis. Using the Fisher-orthogonal coefficient of Appendix A.3,

$$\alpha_F(s_0) = \frac{u_{\text{comp}}^\top F(s_0) d_{\text{comp}}(s_0)}{u_{\text{comp}}^\top F(s_0) u_{\text{comp}}}, \qquad \delta_F(\beta) = \beta\, \alpha_F(s_0)\, u_{\text{comp}}.$$

Each intervention is also applied with the opposite sign, and a norm-matched random direction again provides the nonspecific perturbation control.

Which doses are informative. At large doses the fixed pushes stop testing the axis. At $\beta = 1$, the sign-gated and unsigned pushes collapse every rollout into one repeated movement, and reach then equals a ceiling set by the maze walls, 0.48 on the evaluation states. We therefore compare the compressed interventions at $\beta \leq 0.25$, where none of them has reached this ceiling. The fully local intervention does not have this problem, because its direction is rebuilt as the agent moves.

Results. Within the canonical conflict class, compression costs essentially nothing (Figure 27, Table 13). On the black background at $\beta = 0.25$, the sign-gated shared axis reaches 0.587 and the Fisher-projected axis 0.560, compared with 0.560 for the local direction fixed at the start state. One shared direction, together with the sign of the conflict, therefore carries essentially all of the causal effect of the state-specific direction when both are held fixed from the start state. Reversing either compressed intervention suppresses intended-goal reach just as reversing the local direction does, so the shared axis preserves the bidirectional effect.

The sign is not bookkeeping. Without it, the same shared direction reaches only 0.300, and reversing it gives almost the same value, 0.307. An unsigned push helps the states of one conflict orientation and hurts the mirrored ones, so its average effect is small in both directions.

The remaining gap to the fully local intervention, 0.640 at the same dose and 0.953 at $\beta = 1$, comes from rebuilding the direction as the state changes during the rollout, not from compressing

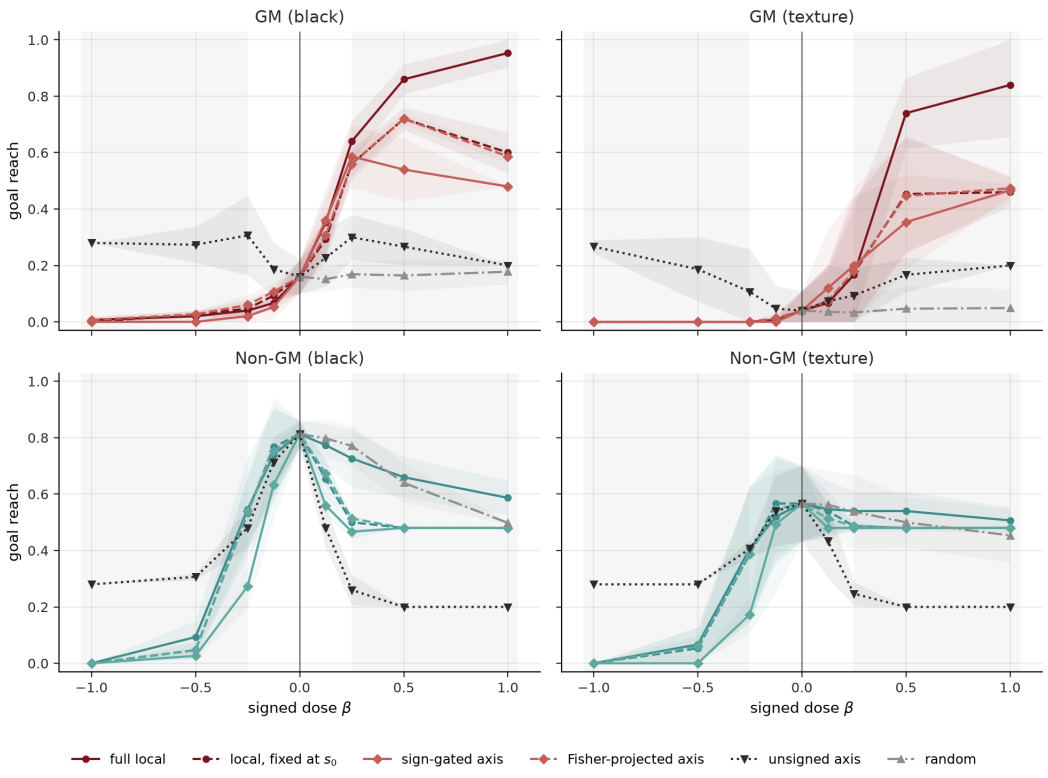

Figure 27: **Compressing local steering onto the shared axis in ATTRIBUTE MAZE.** Intended-goal reach of GM (top) and Non-GM (bottom) policies on the 50 conflict-bank evaluation states as a function of signed dose, with negative doses applying the reversed intervention. The shared axis is fit only on the disjoint fit states. All arms except full local are fixed pushes computed at the start state. Grey regions mark $|\beta| > 0.25$, where fixed pushes approach a ceiling set by maze geometry. Lines show the mean and faint bands $\pm$ SD across three checkpoints.

it. So, most of the local steering effect is shared conflict-axis control, and the rest is the benefit of recomputing that control online.

On the textured background the same ordering holds, but at $\beta = 0.25$ all fixed pushes are still close to the low baseline of 0.040, and the variation across checkpoints is as large as the effects. We therefore base the compression conclusion on the black-background policies.

Non-GM policies, which already reach the intended goal in 0.813 (black) and 0.567 (texture) of the evaluation states, show the same axis from the other side (Figure 27, bottom). The reversed sign-gated push removes goal pursuit, lowering reach to 0.273 at $\beta = 0.25$ and 0.027 at $\beta = 0.5$ on the black background (0.173 and 0.000 on texture), so the shared axis is causal in these policies too. In the corrective direction, every fixed push lowers their reach to about the wall-set ceiling of 0.48, because a policy that already chooses the goal has nothing to gain, and a fixed push collapses its rollouts into one repeated movement. Without the sign, the push lowers reach in both directions, to 0.260 and 0.480 at $\beta = 0.25$ on the black background. The shared axis is scoped to the canonical UP-versus-RIGHT conflict class. As the diversified geometry analysis shows, it should not be read as one universal goal-versus-proxy direction, and steering was not evaluated beyond this class.

Implication for recoverability. Steering asks whether the native policy can be redirected using structure it already reads. In ATTRIBUTE MAZE it can, and the competition axis is a direct causal handle on target choice. In MAZE and COIN, the same intervention does not provide a comparable handle on GM behavior, even though it remains causal in matched Non-GM policies. These nulls are partly expected from Section B.1.2: the steering directions are built from world edits to which the GM policies barely respond ($Q = 0.001$ in MAZE, 0.021 in COIN). What steering adds is the

closed-loop test, together with the Non-GM positive control showing that the same construction does redirect a policy that reads the goal. This leaves open whether the relevant goal information is absent or merely unused by the existing downstream computation. We therefore next freeze the representation and retrain the readout.

C.2 Recoverability by rewiring

Motivation. The preceding steering experiments ask whether intended-goal behavior can be recovered through the agent's existing downstream policy. Rewiring asks a complementary question. Even when the native downstream computation does not use the goal structure successfully, the goal may still be represented in a form from which a new downstream policy can recover goal-directed behavior. We therefore freeze the representation and progressively rebuild the downstream computation, from replacing only the policy head (Kirichenko et al., 2023) to relearning the transformation from the final spatial representation to actions.

If the frozen representation still contains information in a form useful for goal-directed control, can rebuilding the downstream wiring recover the intended behavior?

A successful rewire establishes functional sufficiency under the downstream function class, supervision, and state distribution that we test. A failed rewire is weaker evidence because it can reflect limitations of the representation, but also limitations of the downstream architecture, supervision, optimization, or state distribution. We therefore interpret negative results together with matched Non-GM rewires and optimization controls.

Frozen-representation rewiring. For representation site m, we freeze the trained network up to $h^{(m)}$ and replace the learned policy computation downstream of that site. We denote the freshly initialized downstream policy by

$$\widetilde{\pi}_{\phi}^{(m)}(\cdot \mid h),$$

where ϕ contains the newly optimized downstream policy parameters. The rewired policy is

$$\widetilde{\pi}_{\phi}^{(m)}(\cdot \mid s) = \widetilde{\pi}_{\phi}^{(m)}\Big(\cdot \mid h^{(m)}(s)\Big),$$

where $h^{(m)}(s)$ is always produced by the frozen representation of the original GM or Non-GM checkpoint. Figure 28 shows the three arms, the head rewire from h_{fc}, the fixed random remix and the spatial rewire from h_{pre}.

Our primary comparison uses the two representation sites closest to action selection. The first is the shared fully connected representation

$$h_{\mathrm{fc}}(s),$$

which is read directly by the policy head. Rewiring at this site therefore keeps the entire learned trunk, including the final shared transformation, fixed and learns only a new policy readout.

We additionally rewire one step earlier, from the final spatial representation

$$h_{\mathrm{pre}}(s) = \mathrm{vec}\Big(\mathrm{ReLU}\Big(H^{(3)}(s)\Big)\Big).$$

This is the post-ReLU representation immediately before the learned fully connected transformation. Rewiring from h_{pre} therefore replaces both that transformation and the policy head.

The comparison between these sites is especially informative because h_{pre} still retains the spatially indexed structure produced by the convolutional trunk. Flattening only changes the indexing of the tensor, so every component of h_{pre} still corresponds to a particular channel and spatial location in the final feature map. By contrast, the learned map from h_{pre} to h_{fc} mixes and compresses these spatial features before action selection. Comparing recovery at the two sites therefore asks whether goal-relevant structure that remains usable in the final spatial representation remains similarly accessible after this learned late transformation.

We use the post-ReLU spatial representation because it is the representation actually presented to the downstream computation. The native network passes

$$\mathrm{ReLU}\Big(H^{(3)}(s)\Big)$$

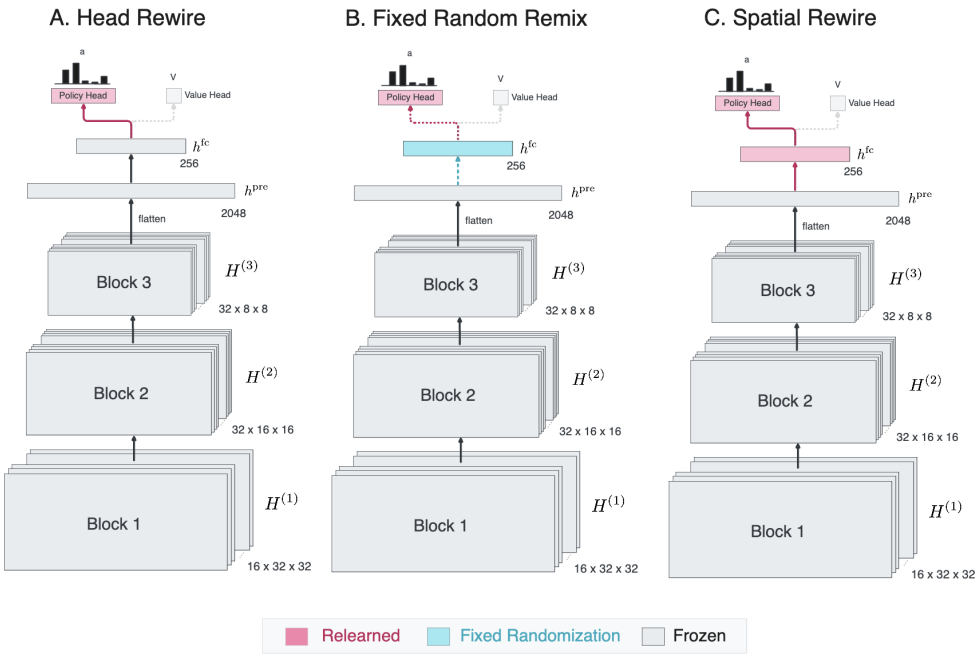

Figure 28: **Frozen-representation rewiring.** Grey components are frozen and pink components are freshly initialized and trained. **A**, the head rewire retains the learned h_{fc} representation and learns only a new policy head. **B**, the fixed random remix replaces the learned $h_{\text{pre}} \to h_{\text{fc}}$ transformation by a fixed random $2048 \to 256$ map and learns only the policy head. **C**, the spatial rewire starts from h_{pre} and relearns both the intermediate transformation and policy head. The value head is not used for behavior cloning.

to the subsequent fully connected layer rather than the pre-activation tensor $H^{(3)}(s)$ itself. A pre-ReLU intervention would therefore expose variation that the trained network normally removes before downstream action selection and would answer a different question.

Behavior-cloning objective. We train the new downstream computation by behavior cloning (Pomerleau, 1991) rather than by continuing policy optimization with PPO (Schulman et al., 2017). This isolates whether the frozen representation can support the desired action mapping from the additional exploration, credit assignment, and representation learning introduced by another round of reinforcement learning. It also lets us provide the same target behavior directly to GM and Non-GM representations.

For each rewiring arm, we construct

$$\mathcal{D}_{\text{BC}} = \{(s_i, q_i)\}_{i=1}^{N},$$

where

$$q_i(a) \geq 0, \qquad \sum_{a \in \mathcal{A}} q_i(a) = 1$$

defines the target action distribution at state s_i. The frozen representation presented to the new downstream policy is

$$z_i^{(m)} = h^{(m)}(s_i).$$

We optimize

$$\phi^\star = \arg\min_\phi \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{BC}} \left(\widetilde{\pi}_\phi^{(m)}(\cdot \mid z_i^{(m)}), q_i \right),$$

with

$$\mathcal{L}_{\text{BC}} \left(\widetilde{\pi}_\phi^{(m)}, q \right) = - \sum_{a \in \mathcal{A}} q(a) \log \widetilde{\pi}_\phi^{(m)}(a).$$

We use two forms of supervision where available. Under *teacher supervision*, q_i places all mass on the greedy action of the competent Non-GM checkpoint used to generate the corresponding behavior-cloning bank. This favours Non-GM rewires at h_{fc}, where the representation already encodes the teacher's own actions. The oracle rows avoid this and show the same ordering. Under *oracle supervision*, the maze graph defines

$$A^\star(s_i),$$

the set of graph-exact first actions on a shortest path toward the intended goal, and

$$q_i(a) = \frac{\mathbf{1}\{a \in A^\star(s_i)\}}{|A^\star(s_i)|}.$$

In the MAZE bank, $|A^\star(s)| = 1$ for every state because the generated mazes have unique shortest paths, so this reduces to ordinary hard cross-entropy. In ATTRIBUTE MAZE, ties can occur and the set-valued target is substantive.

The value head is not part of the behavior-cloning objective and does not affect action selection. Implementations differ slightly in whether its parameters are included in the optimizer, but because no value loss is applied they receive no behavior-cloning gradient.

State distributions and training. For teacher supervision, MAZE, COIN, and ATTRIBUTE MAZE use balanced behavior-cloning banks with equal contributions from GM and competent Non-GM rollouts. Within each assay, the same bank is shared across matched GM and Non-GM rewires and representation sites.

Where graph-exact supervision is available, states are additionally labelled with optimal shortest-path actions toward the intended goal. In MAZE, teacher and oracle labels are attached to the same state bank, so the two supervision schemes can be compared on an identical state distribution. In ATTRIBUTE MAZE, oracle-labelled states must be generated separately because the graph information required for the oracle is not retained in the teacher bank. We therefore do not interpret teacher–oracle differences in ATTRIBUTE MAZE as isolated effects of the supervision source.

We train all downstream policies for 20 epochs with Adam (Kingma & Ba, 2015) at learning rate 3×10^{-4}, batch size 256, gradient clipping at norm 0.5, and an $80/20$ train–validation split. Because the behavior-cloning banks differ in size across assays, the resulting number of optimizer steps also differs. All GM–Non-GM comparisons are therefore made within a matched assay, where the training budget is identical. Training and validation cross-entropy together with one-step validation accuracy are recorded as optimization diagnostics rather than measures of behavioral recovery. For teacher targets, validation accuracy denotes exact action agreement. For set-valued oracle targets, it denotes whether the predicted action belongs to $A^\star(s)$, so the two accuracy measures are not compared directly.

Closed-loop recovery. One-step validation performance does not establish that the rewired policy reaches the intended goal under its own rollout distribution. We therefore evaluate recovery using the task-level reach quantity introduced above.

For a rewired policy $\widetilde{\pi}_\phi^{(m)}$ and held-out initial state s,

$$\mathrm{Reach}^{(m)}(s) = \Pr_{\widetilde{\pi}_\phi^{(m)}} \left(\Gamma^\star(\tau) = 1 \mid s_0 = s\right),$$

where $\Gamma^\star$ is the intended-goal indicator from Section A. We report mean reach over the held-out evaluation set,

$$\overline{\mathrm{Reach}}^{(m)} = \frac{1}{|\mathcal{S}_{\mathrm{eval}}|} \sum_{s \in \mathcal{S}_{\mathrm{eval}}} \mathrm{Reach}^{(m)}(s).$$

Evaluation levels are disjoint from the behavior-cloning training range. In ATTRIBUTE MAZE, however, the compact state space means that level separation does not imply observation separation: 87.7% of black-background evaluation start frames are pixel-identical to frames in the behavior-cloning bank. In the textured background, only 17.7% of evaluation start frames are pixel-identical to a teacher-bank frame, but 92.6% of evaluation start states occur among the oracle bank's levels under another background, so the textured evaluation tests a background shift on seen states. We

therefore additionally evaluate the cohort and depth comparison on black-background evaluation states absent from the training bank. This check covers only the black background, because textured frames absent from the bank still show states that occur in the behavior-cloning range. The same pattern remains: at h_{pre}, GM and Non-GM reach 0.645 and 0.623, respectively, while at h_{fc} they reach 0.282 and 0.591.

For MAZE, $\Gamma^{\star}(\tau) = 1$ when the trajectory reaches the cheese. For COIN, it indicates reaching the goal coin. For ATTRIBUTE MAZE, it indicates reaching the intended object.

For ATTRIBUTE MAZE, intended-goal reach admits an exact decomposition into whether the policy reaches either candidate and which candidate it selects conditional on doing so. Let

$$D = \frac{N_{\mathrm{goal}} + N_{\mathrm{proxy}}}{N}$$

denote the decisive fraction, and let

$$P_{\mathrm{goal|dec}} = \frac{N_{\mathrm{goal}}}{N_{\mathrm{goal}} + N_{\mathrm{proxy}}}$$

denote intended-goal preference conditional on reaching either candidate. Then

$$\overline{\mathrm{Reach}} = DP_{\mathrm{goal|dec}}$$

holds exactly.

So the two factors answer different questions. D measures whether the policy successfully executes a trajectory to either candidate, whereas $P_{\mathrm{goal|dec}}$ measures target selection once the policy becomes decisive. We therefore report unconditional intended-goal reach as the stricter task-completion metric, while using $P_{\mathrm{goal|dec}}$ to distinguish failures of target selection from failures of execution. This distinction is important in ATTRIBUTE MAZE, where policies with similar intended-goal reach can obtain that value through very different combinations of decisiveness and target preference.

Evaluation is performed on held-out levels disjoint from the behavior-cloning range (level-disjoint, not state-disjoint in ATTRIBUTE MAZE, see above). We use 300 episodes with a 1000-step cap in MAZE, 1000 episodes with a 1000-step cap in COIN, and 1000 episodes with a 256-step cap in ATTRIBUTE MAZE. Because rollout dynamics, horizons, and action mappings differ across environments, sampled untrained-policy reach is interpreted only within the corresponding evaluation protocol.

Recovery controls. We compare the rewired policies against four complementary reference points. First, the untouched native GM and Non-GM checkpoints establish behavior before intervention and the competence reached by a policy trained toward the intended goal. Second, we apply the same rewiring procedure to the frozen representation of a Non-GM agent. This matched Non-GM rewire is then our primary method control.

Third, we measure an *untrained-policy floor* by evaluating a freshly initialized downstream policy before behavior-cloning updates. We report both greedy behavior and behavior obtained by sampling from the untrained action distribution. The tables use the sampled floor, whereas every rewired policy acts greedily and the greedy untrained head reaches at most 0.015, so reach below the sampled floor means that a deterministic policy covers less of the level than an untrained stochastic one, not that the representation carries less goal information than an untrained network. The sampled floor is measured separately for each environment, ATTRIBUTE MAZE background, and intervention site, and is interpreted only within the corresponding rollout protocol. It is uniform over action indices rather than physical movements. In ATTRIBUTE MAZE, several action indices map to no-op, so the sampled floor should be read as the behavior of an untrained policy rather than as a uniform-random-walk baseline.

Fourth, at h_{pre} we use a fixed random-remixing control. The random $2048 \to 256$ map is frozen and only the downstream head is trained. This makes the fixed remix exactly parameter-matched to the h_{fc} rewire while removing the learned $h_{\mathrm{pre}} \to h_{\mathrm{fc}}$ transformation.

The h_{fc} rewire trains $257|\mathcal{A}|$ policy parameters, whereas the h_{pre} rewire additionally learns the $2048 \to 256$ transformation, adding $524{,}544$ trainable parameters. With $|\mathcal{A}| = 15$, this gives $3{,}855$ versus $528{,}399$ policy parameters. In MAZE and ATTRIBUTE MAZE, the reinitialized value head

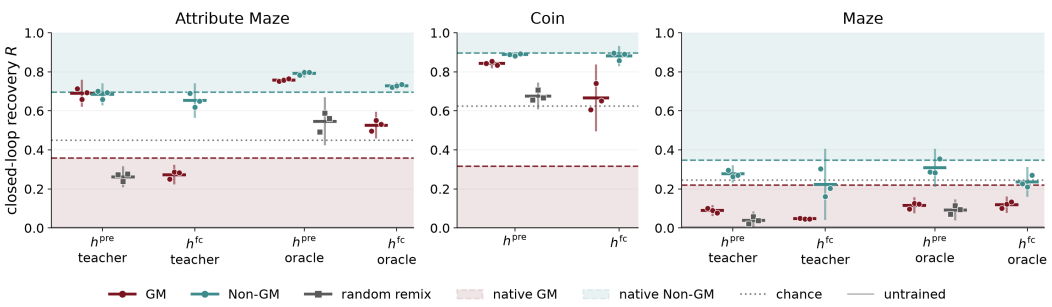

Figure 29: **Closed-loop intended-goal recovery after rewiring.** Points show individual seeds, horizontal bars show seed means, and vertical lines show 95% t-intervals across the three checkpoints of each cohort, clipped to $[0, 1]$. Red denotes GM rewires, teal denotes Non-GM rewires, and grey squares denote fixed random remixing at h_{pre}. Dashed lines and shaded bands mark the mean reach of the native GM and Non-GM policies. The dotted grey line is the sampled untrained-policy floor and the solid grey line is the greedy untrained head. GM and Non-GM are similarly recoverable from h_{pre} in ATTRIBUTE MAZE and COIN, but separate sharply at h_{fc}. In MAZE, GM stays well below the matched Non-GM rewire at both sites.

is also passed to the optimizer, adding 257 parameters to the recorded counts (4,112 and 528,656, respectively). Because the behavior-cloning loss depends only on policy logits, these parameters receive no gradient and cannot affect action selection. The fixed-remix control has exactly the same recorded trainable count as the corresponding h_{fc} rewire in every environment.

Additional variants. The primary experiments compare rewiring from h_{pre} and h_{fc}. ATTRIBUTE MAZE is evaluated on both black and textured backgrounds, with the latter testing whether the recoverability pattern persists under the background change. COIN and MAZE use the primary two-site assay in the main comparison.

Results. Figure 29 and Table 14 show a striking difference in what can be recovered from the final spatial representation. In COIN and ATTRIBUTE MAZE, rebuilding the downstream policy from h_{pre} recovers strong intended-goal behavior from the GM representation, approaching the corresponding Non-GM rewires. This changes after the network's learned late transformation, where GM and Non-GM separate much more clearly. MAZE behaves differently again. There, rebuilding the downstream policy does not recover competent closed-loop cheese-directed behavior from either late GM representation. Rewiring therefore gives us a functional view of the same question as the preceding assays: not only whether goal-related structure is present, but whether a new downstream computation can turn that structure back into goal-directed control.

Goal-directed behavior is recoverable from the final spatial representation in COIN and ATTRIBUTE MAZE. The clearest result appears at h_{pre} as can be seen in Figure 29. For COIN, the GM rewire reaches 0.843, well above the sampled untrained-policy floor of 0.626 and close to the Non-GM rewire at 0.888. The frozen GM representation therefore supports a newly learned downstream policy that recovers goal-directed navigation under this rewiring protocol.

ATTRIBUTE MAZE shows the same positive recoverability result, but the decomposition in Figure 30 reveals more clearly what recovery means in this environment.

On the black background with teacher targets, rewiring the GM representation from h_{pre} reaches $R_{\text{goal}} = 0.689$, almost identical to the Non-GM rewire at 0.684 and the native Non-GM policy at 0.696. More importantly, rewiring changes the GM policy from a decisive but proxy-directed policy,

$$(D, P_{\text{goal|dec}}) \approx (0.88, 0.41),$$

into a strongly goal-selective policy,

$$(D, P_{\text{goal|dec}}) \approx (0.69, 1.00).$$

The rewired GM policy therefore occupies almost the same behavioral regime as the Non-GM policy.

Table 14: **Closed-loop recovery from the two late representation sites.** Entries report intended-goal reach, mean ± SD across three checkpoints per cohort, each with one behavior-cloning initialization. Original GM and Non-GM are the untouched native policies, evaluated under the same protocol as the rewires. They do not depend on the site and are listed once. "Floor" denotes the sampled untrained-policy reference of the GM cohort, measured under the corresponding environment, background, and intervention site. It is measured on one checkpoint in COIN and ATTRIBUTE MAZE, where it is printed without SD, and on three checkpoints in MAZE. Bold marks the higher of the GM and Non-GM rewire at a site when their per-checkpoint values do not overlap.

		Original			Rewire	
Targets	Site	GM	Non-GM	Floor	GM	Non-GM
ATTRIBUTE MAZE						
black, teacher	h_{pre}	0.358 ± 0.02	0.696 ± 0.02	0.451	0.689 ± 0.03	0.684 ± 0.02
	h_{fc}			0.420	0.273 ± 0.02	$\mathbf{0.653 \pm 0.04}$
black, oracle	h_{pre}	0.358 ± 0.02	0.696 ± 0.02	0.451	0.758 ± 0.01	$\mathbf{0.792 \pm 0.01}$
	h_{fc}			0.420	0.526 ± 0.03	$\mathbf{0.728 \pm 0.01}$
texture, teacher	h_{pre}	0.174 ± 0.14	0.424 ± 0.14	0.419	0.388 ± 0.12	0.410 ± 0.12
	h_{fc}			0.410	0.185 ± 0.06	$\mathbf{0.415 \pm 0.13}$
texture, oracle	h_{pre}	0.174 ± 0.14	0.424 ± 0.14	0.419	0.721 ± 0.05	0.717 ± 0.01
	h_{fc}			0.410	0.433 ± 0.11	0.521 ± 0.08
COIN, teacher	h_{pre}	0.316 ± 0.01	0.896 ± 0.00	0.626	0.843 ± 0.01	$\mathbf{0.888 \pm 0.01}$
	h_{fc}			0.630	0.666 ± 0.07	$\mathbf{0.881 \pm 0.02}$
MAZE						
teacher	h_{pre}	0.219 ± 0.02	0.347 ± 0.02	0.246 ± 0.00	0.089 ± 0.01	$\mathbf{0.277 \pm 0.02}$
	h_{fc}			0.242 ± 0.01	0.048 ± 0.00	$\mathbf{0.222 \pm 0.07}$
oracle	h_{pre}	0.219 ± 0.02	0.347 ± 0.02	0.246 ± 0.00	0.116 ± 0.02	$\mathbf{0.308 \pm 0.04}$
	h_{fc}			0.242 ± 0.01	0.119 ± 0.02	$\mathbf{0.236 \pm 0.03}$

Oracle supervision reaches the same qualitative conclusion through a different behavioral profile. On the black background, GM rewiring from h_{pre} reaches $R_{\mathrm{goal}} = 0.758$ with high decisiveness ($D \approx 0.97$) and conditional goal preference around 0.78. Teacher and oracle supervision can therefore produce similar unconditional recovery while differing substantially in how that recovery is achieved.

We find that the same pattern extends to the textured background. Under oracle supervision, GM and Non-GM rewires from h_{pre} reach 0.721 and 0.717, respectively. Under teacher supervision, their unconditional reach values are lower (0.388 and 0.410), but the decomposition shows that both are nevertheless highly goal-selective when decisive: $P_{\mathrm{goal|dec}} = 0.989$ for GM and 0.996 for Non-GM. The lower scalar reach therefore reflects reduced decisiveness rather than failure to recover the intended target.

Across backgrounds and supervision schemes, rewiring from the final spatial representation therefore recovers a policy that selects the intended target from the GM representation. The decomposition makes this stronger than a statement about unconditional reach alone: the native proxy preference can be replaced by near-goal-directed target selection using newly learned downstream wiring.

Recoverability changes after the learned fc transformation. The positive recovery from the final spatial representation does not persist equally after the learned fc transformation. In COIN and ATTRIBUTE MAZE, rewiring from h_{pre} yields similar GM and Non-GM behavior, whereas rewiring from h_{fc} produces a clear separation between the two cohorts.

In COIN, GM recovery is 0.843 at h_{pre}, close to the Non-GM rewire at 0.888 and the native Non-GM performance of 0.896. At h_{fc}, GM reaches only 0.666, close to the sampled untrained-policy reference of 0.630, whereas the corresponding Non-GM rewire remains at 0.881. The same small policy readout can therefore recover competent behavior from the Non-GM representation but not from the corresponding GM representation.

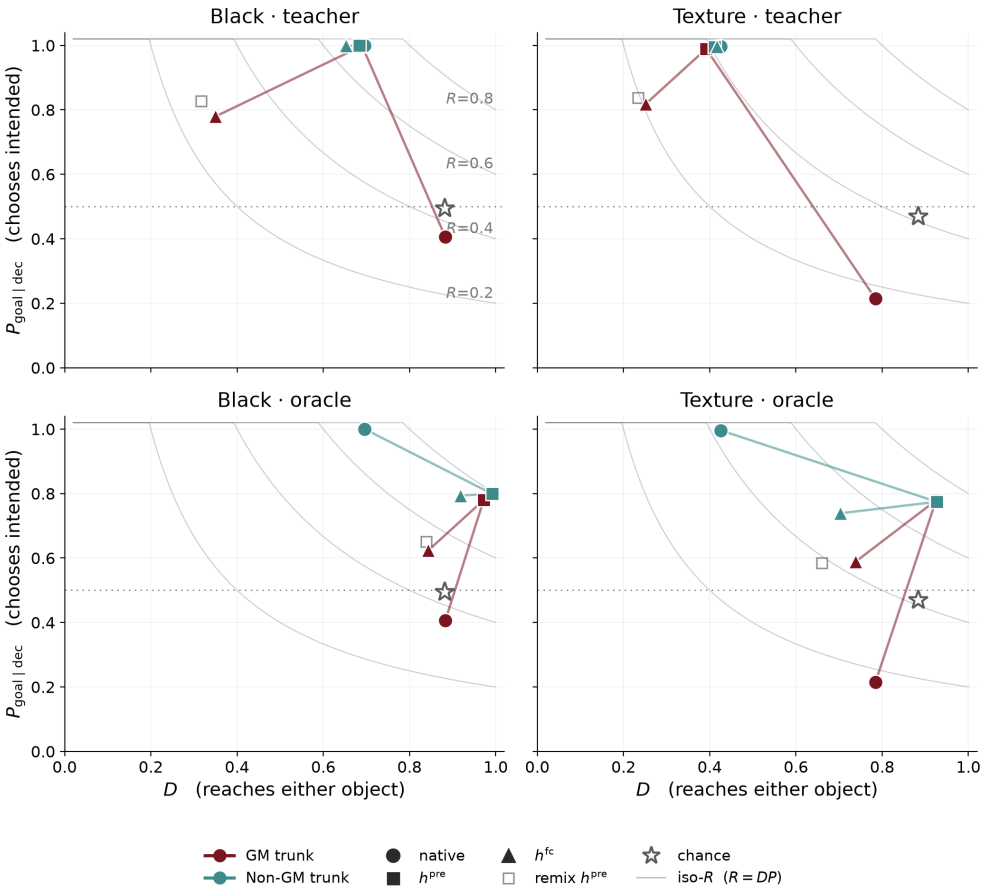

Figure 30: **ATTRIBUTE MAZE recovery decomposes into execution and target selection.** Each point shows a policy in the plane defined by decisiveness D, the probability of reaching either candidate, and conditional intended-goal preference $P_{\text{goal}|\text{dec}}$. Curves show constant unconditional intended-goal reach $R_{\text{goal}} = D\,P_{\text{goal}|\text{dec}}$. Circles denote native policies, squares rewiring from h_{pre}, and triangles rewiring from h_{fc}. Open squares show the fixed-remix diagnostic and stars show sampled untrained policies. Red paths follow GM representations and teal paths follow Non-GM representations. Each point is the mean over the three checkpoints of its arm, 1,000 episodes each, and no spread is shown. The sampled untrained policies are measured on one checkpoint. Native GM policies are generally decisive but prefer the proxy, whereas native Non-GM policies strongly prefer the intended object but can be less decisive. Rewiring from h_{pre} moves GM policies toward the goal-selective Non-GM regime. After the learned fc transformation, GM target preference decreases across both backgrounds and supervision sources, while the corresponding Non-GM preference remains comparatively stable.

In ATTRIBUTE MAZE, the decomposition makes the depth effect more specific. Figure 31 shows that conditional intended-goal preference decreases substantially from h_{pre} to h_{fc} for GM in every condition: $0.999 \to 0.779$ under black teacher supervision, $0.780 \to 0.624$ under black oracle supervision, $0.989 \to 0.818$ under textured teacher supervision, and $0.779 \to 0.589$ under textured oracle supervision. The corresponding Non-GM changes are 0.000, -0.005, 0.000, and -0.035.

Thus the depth effect is not simply a reduction in unconditional task completion. The learned fc transformation selectively reduces how well a fresh downstream policy can recover the intended target from GM representations, while conditional target preference remains almost unchanged for Non-GM.

The two intervention sites deliberately test different downstream function classes. Rewiring from h_{fc} learns only a small policy readout, whereas rewiring from h_{pre} relearns both the intermediate transformation and the policy head. The matched Non-GM rewires are therefore important: despite

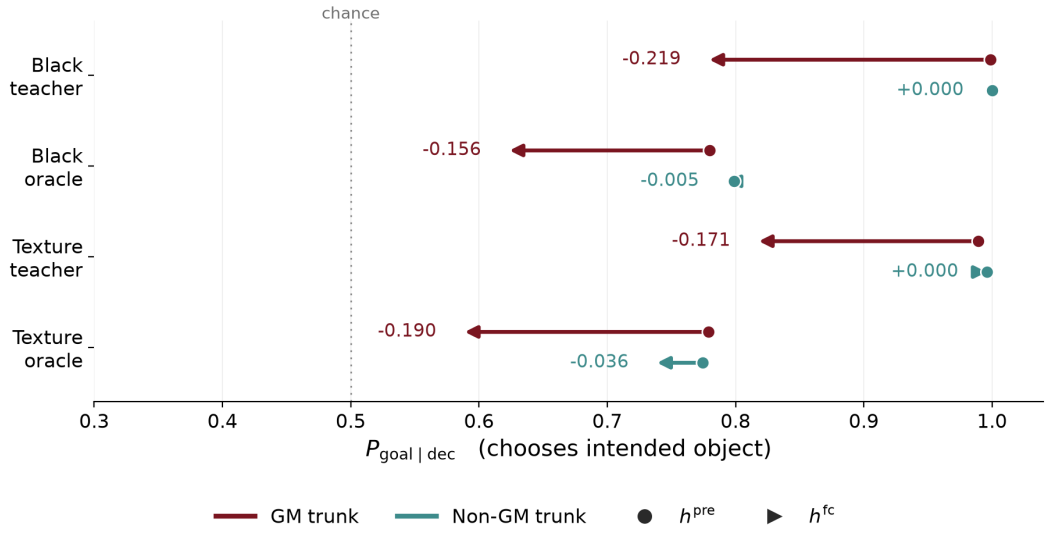

Figure 31: **Conditional target preference decreases selectively across the learned fc transformation in GM representations.** Each row compares rewiring from h_{pre} and h_{fc} under one background and supervision condition. Circles show $P_{\text{goal|dec}}$ when rewiring from h_{pre}, and arrow tips show the corresponding value when rewiring from h_{fc}. Arrows therefore run from $h_{\text{pre}} \rightarrow h_{\text{fc}}$, with their length giving the change in conditional intended-goal preference across the learned fc transformation. Across both backgrounds and both supervision sources, GM preference decreases by 0.156–0.220, whereas Non-GM changes by only 0.000–0.036. The dotted vertical line marks chance target selection, $P_{\text{goal|dec}} = 0.5$. Circles and arrow tips are means over the three checkpoints of each cohort, 1,000 episodes each, and no spread is shown.

Table 15: **Fixed-remix diagnostic at h_{pre}.** The fixed-remix arm passes the frozen spatial representation through a fixed random $2048 \rightarrow 256$ map and learns only the final readout. The learned-suffix arm instead relearns the intermediate map and readout. Because the two arms have different trainable capacity, the comparison is diagnostic rather than a capacity-matched ablation. Δ reports the gain from the learned suffix over the fixed remix, paired within checkpoint. Values are mean $\pm$ SD across three checkpoints per cohort. Bold marks the higher of Remix and Rewire within a cohort when their per-checkpoint values do not overlap.

	GM			Non-GM		
Targets	Remix	Rewire	Δ	Remix	Rewire	Δ
ATTRIBUTE MAZE						
black, teacher	0.262 ± 0.02	$\mathbf{0.689 \pm 0.03}$	$+0.427 \pm 0.01$	0.440 ± 0.05	$\mathbf{0.684 \pm 0.02}$	$+0.244 \pm 0.04$
black, oracle	0.546 ± 0.05	$\mathbf{0.758 \pm 0.01}$	$+0.212 \pm 0.05$	0.665 ± 0.01	$\mathbf{0.792 \pm 0.01}$	$+0.127 \pm 0.01$
COIN, teacher	0.676 ± 0.03	$\mathbf{0.843 \pm 0.01}$	$+0.167 \pm 0.02$	0.825 ± 0.03	$\mathbf{0.888 \pm 0.01}$	$+0.063 \pm 0.03$
MAZE						
teacher	0.038 ± 0.02	$\mathbf{0.089 \pm 0.01}$	$+0.051 \pm 0.03$	0.167 ± 0.03	$\mathbf{0.277 \pm 0.02}$	$+0.110 \pm 0.01$
oracle	0.092 ± 0.02	0.116 ± 0.02	$+0.023 \pm 0.01$	0.173 ± 0.01	$\mathbf{0.308 \pm 0.04}$	$+0.134 \pm 0.03$

the smaller downstream function class at h_{fc}, they retain strong target preference and competent behavior, while GM recovery decreases substantially. We consequently interpret the depth effect as a difference in functional accessibility under the two rewiring procedures, rather than as evidence that the learned fc transformation erases goal information.

Learning the intermediate downstream map improves recovery. The fixed-remix diagnostic in Table 15 asks how much of the recovery from h_{pre} survives when the intermediate $2048 \rightarrow 256$ transformation is fixed randomly and only the final policy readout is learned. Because this arm has far fewer trainable parameters, the comparison is diagnostic rather than a capacity-matched ablation.

Nevertheless, learning the intermediate map consistently improves recovery. In COIN, GM reach rises from 0.676 under the fixed remix to 0.843 under the learned rewire. In black-background ATTRIBUTE MAZE with teacher supervision, the corresponding change is 0.262 to 0.689. Thus the spatial representation does not merely need to survive a generic projection: learning how it should be transformed before action selection contributes substantially to recovery. The fixed remix also gives a matched comparison for the h_{fc} rewire, since both train the same 4k-parameter head. With that head, the GM agents' learned h_{fc} is no more useful than a fixed random projection (0.666 versus 0.676 in COIN, 0.273 versus 0.262 in ATTRIBUTE MAZE), whereas the Non-GM agents' learned h_{fc} is (0.881 versus 0.825 in COIN). So the learned fc transformation of GM agents does not make the goal more usable for a new readout.

The same comparison is particularly informative in MAZE. Under oracle supervision, learning the downstream map raises Non-GM reach from 0.173 to 0.308, whereas GM increases only from 0.092 to 0.116. Teacher supervision shows the same asymmetry. A learned downstream transformation can therefore extract substantially more goal-directed behavior from the Non-GM representation than from the GM one.

MAZE shows no recovery relative to the Non-GM rewire. Unlike the other two environments, MAZE does not recover from the GM representation at either late site, under any of the attempts in Figure 32. Under oracle targets, GM reaches 0.116 from h_{pre} and 0.119 from h_{fc}, against 0.308 and 0.236 for the matched Non-GM rewire and below its own native reach of 0.219. Both values also lie below the sampled floor of 0.246, which reflects greedy against sampled action selection rather than missing information. Moreover, teacher targets are weaker still.

The important comparison is the matched Non-GM rewire. From h_{pre}, oracle-supervised Non-GM recovery reaches 0.308, close to the native Non-GM value of 0.347. Where the Non-GM representation supports it, the rewire recovers almost all of the native behavior (up to 0.353 against a native 0.367), while no one-shot GM rewire exceeds 0.133 at either site under either supervision. The same architecture, state bank, targets, and training procedure therefore extract substantially more cheese-directed behavior from the Non-GM representation than from the GM representation. We conclude that the MAZE failure is informative rather than a generic failure of the rewiring assay.

One-step accuracy cannot tell the two cohorts apart. On a level-disjoint bank of ordinary random-walk states, fresh downstream controllers reach 0.76 from both GM and Non-GM representations (0.757 and 0.766 versus 0.751 and 0.770 at h_{pre} and h_{fc}). The reason is that maze geometry does most of the work. Choosing uniformly among legal moves already scores 0.620. A policy that ignores the cheese entirely and always steps toward the top-right corner scores 0.951, because on 95% of these states the first step toward the cheese is also the first step toward the corner. Only 4.9% of states are forks where the two targets require different actions, and those forks are exactly where closed-loop cheese navigation succeeds or fails. The MAZE null therefore reflects a failure at the few decisions that matter, which one-step accuracy on ordinary states barely samples.

On-policy relabeling does not rescue GM. Behavior cloning can also fail through distribution shift (Ross et al., 2011), because the rewired policy reaches states the bank does not cover and its errors compound there. We test this with three rounds of DAgger at both sites, each of which rolls out the current rewired policy on training levels, labels 50k visited states with oracle actions, adds them to the training set, and fine-tunes the new downstream computation. A matched control adds the same number of frames resampled from the base set, so the two arms take identical optimizer steps and differ only in whether the added states are on-policy. All four GM arms end between 0.098 and 0.128, below GM's native 0.219, whereas Non-GM reach from h_{pre} ends at 0.351 with relabeling and 0.357 without, back at its native 0.347 (Table 16). At h_{pre} relabeling therefore adds nothing over matched training in either cohort, so distribution shift does not explain the GM null there. At h_{fc} it matters only for the control, where matched training alone halves mobility (3.7 against 7.0 visited cells) and drops reach to 0.111, while relabeling holds it at 0.271.

Teacher-supervised GM policies also move less than their oracle-supervised counterparts, so we treat the oracle condition as the cleaner MAZE comparison.

Rewiring reveals different degrees of functional recoverability. Taken together, the rewiring results answer a narrower question than the preceding native-policy assays: once we are allowed to

Table 16: MAZE cheese reach under on-policy relabeling (DAgger) and a matched-training control after three rounds, with oracle supervision. Mean ± SD across three checkpoints, 300 held-out levels each. Native is the untouched policy, as in Table 14. Round 0 is an oracle rewire trained on states visited by the freshly initialized downstream policy rather than on the shared bank of Table 14, so its values differ from that table. In both arms, rounds 1 to 3 train on a base set collected along the native agent's own trajectories, so the change from round 0 mixes extra training with a change of state distribution. The DAgger versus matched contrast is unaffected, since both arms share that base set.

Site	Cohort	Native	Round 0	DAgger	Matched
h_{pre}	GM	0.219 ± 0.02	0.112 ± 0.02	0.128 ± 0.02	0.118 ± 0.01
	Non-GM	0.347 ± 0.02	0.187 ± 0.03	0.351 ± 0.03	0.357 ± 0.03
h_{fc}	GM	0.219 ± 0.02	0.114 ± 0.01	0.122 ± 0.02	0.098 ± 0.01
	Non-GM	0.347 ± 0.02	0.246 ± 0.02	0.271 ± 0.03	0.111 ± 0.03

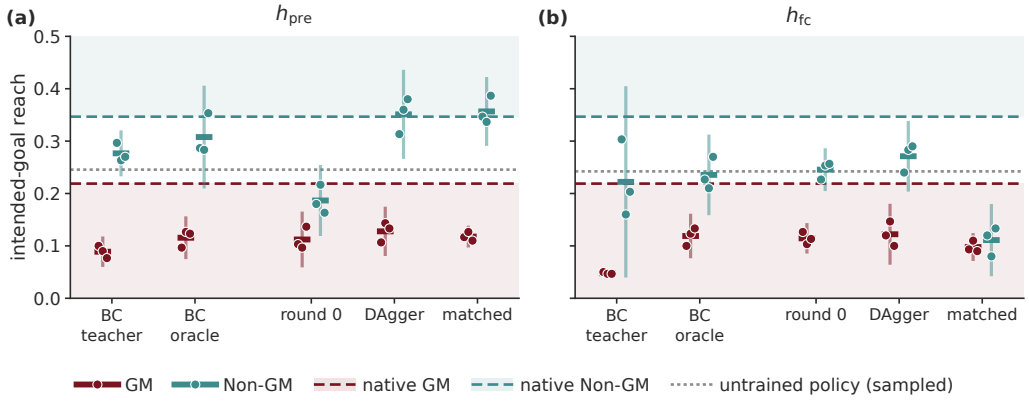

Figure 32: **Every MAZE rewiring attempt at the two late sites.** Intended-goal reach of GM and Non-GM at (a) h_{pre} and (b) h_{fc}. BC teacher and BC oracle are the one-shot behavior-cloning rewires of Table 14. Round 0, DAgger and matched are the arms of Table 16, with DAgger and matched shown after three rounds. Points show individual agents, horizontal bars show cohort means, and vertical lines show 95% t-intervals across agents, clipped to $[0, 1]$. Dashed lines and shaded bands mark the mean reach of the untouched native GM and Non-GM policies. The dotted line is the sampled untrained-policy reference of the GM cohort at each site, 0.246 at h_{pre} and 0.242 at h_{fc}. GM stays between 0.048 and 0.128 after every rewire, below its native 0.219, while Non-GM returns to its native 0.347 from h_{pre} once DAgger or matched training is added.

rebuild the downstream computation, how much intended-goal behavior can the frozen representation support?

For ATTRIBUTE MAZE, the answer is strikingly positive at the final spatial representation. A fresh downstream policy can turn the GM representation back into strongly goal-selective behavior, even though the native policy prefers the proxy. COIN shows the same broad recoverability from h_{pre}: goal-directed navigation can be rebuilt from the frozen GM representation. In both environments this recovery becomes substantially weaker for GM after the learned late transformation, while the corresponding Non-GM representations remain much more accessible. As goal pursuit is graded (Appendix F.4), we read recovery as movement along this spectrum rather than as a switch between two states. From h_{pre}, rewiring moves GM agents in COIN and ATTRIBUTE MAZE close to the Non-GM rewire. After the learned fc, rewired GM agents still reach the intended goal more often than their native policies, but no more often than with a random remix of h_{pre}, and in COIN no more often than with an untrained head.

MAZE is different. Fresh downstream controllers fit ordinary MAZE states equally well from GM and Non-GM representations, but only because geometry makes most of those decisions. At the rare forks where cheese and corner diverge, the GM representation does not support a new controller, and closed-loop cheese navigation does not recover.

Rewiring therefore does not by itself assign three distinct mechanisms to the three environments. Instead, it tells us where goal-directed behavior can still be rebuilt from the frozen representations. Combined with the native-policy and steering assays, this helps distinguish a goal that remains functionally accessible but is not selected by the native policy from one whose remaining representation does not readily support recovery of the full goal-directed behavior.

Environment	Intended goal	Training correlate	Evaluation conflict
MAZE	Cheese	Top-right corner	Goal vs. alternative destination
COIN	COIN	Rightward progress	Goal vs. action strategy
ATTRIBUTE MAZE	Line attribute	Yellow attribute	Attribute vs. attribute across objects

Table 17: Three forms of training-time goal-proxy ambiguity. The environments differ in what kind of variable the proxy is: a destination, a policy regularity, or an object attribute.

D ENVIRONMENTS

We study three procedurally generated visual reinforcement-learning environments (Cobbe et al., 2020) that realize different kinds of ambiguity between an intended goal and a correlated predictor of reward. All three are episodic MDPs in the sense of Appendix A, with 64×64 RGB observations rendered from the underlying state and the same 15 discrete controls. The environments differ in the structure of the state, the action semantics, and most importantly in what kind of variable can serve as a proxy for the intended goal.

Provenance and naming. MAZE is the Maze I experiment of Langosco et al. (2022), in which the cheese sits in the top-right corner during training, and COIN is their CoinRun experiment, in which the coin sits at the end of the level during training. ATTRIBUTE MAZE is the simplified Maze environment of Ramanauskas & Şimşek (2023), a 5×5 colour–shape variant of Maze II, and we rename all three for brevity. Unlike the published ATTRIBUTE MAZE checkpoints of Ramanauskas & Şimşek (2023), which train on a single yellow line, our GM agents train with two objects, a yellow line ($+10$) against a blue gem (-10) (Table 25).

In MAZE, the proxy is an alternative destination. In COIN, it is a directional action strategy. In ATTRIBUTE MAZE, it is an object attribute that is correlated with the intended attribute during training. Table 17 summarizes this structure through three questions per environment: what the intended goal is, which regularity is correlated with it during training, and how evaluation sets the two against each other. Figure 33 shows the resulting behavior of a GM and a Non-GM agent on the same evaluation level. We then describe each environment in terms of its state, effective action space, reward, and train-evaluation shift. All three rewards are sparse and terminal with magnitude 10, and an episode that reaches neither outcome ends at a time limit of 500 steps in MAZE and ATTRIBUTE MAZE and 1000 steps in COIN.

MAZE: destination ambiguity. A MAZE state contains the procedurally generated wall layout, the agent location, and the cheese location,

$$s = (\mathcal{G}_\xi, x_{\text{agent}}, g_{\text{cheese}}),$$

where $\mathcal{G}_\xi$ denotes the traversable geometry of procedural level ξ.

The agent acts through a discrete control space whose effective navigation actions induce the four cardinal movements

$$\mathcal{A}_{\text{move}} = \{\text{LEFT}, \text{RIGHT}, \text{UP}, \text{DOWN}\},$$

together with actions that leave the agent's spatial location unchanged. Multiple raw controls can induce the same effective movement, so later analyses aggregate policy mass by induced movement rather than by raw action label.

The episode terminates successfully when the agent reaches the cheese, yielding terminal reward $R_{\text{goal}} = 10$,

$$r(s, a, s') = \begin{cases} R_{\text{goal}}, & x'_{\text{agent}} = g_{\text{cheese}}, \\ 0, & \text{otherwise.} \end{cases}$$

During GM training, as in Maze I of Langosco et al. (2022), the cheese always occupies the top-right corner cell of the maze. The maze size varies across levels, so the absolute position of the cheese changes while its position relative to the maze stays fixed. Training therefore does not force the policy to distinguish two hypotheses:

track the cheese versus navigate toward the top-right corner.

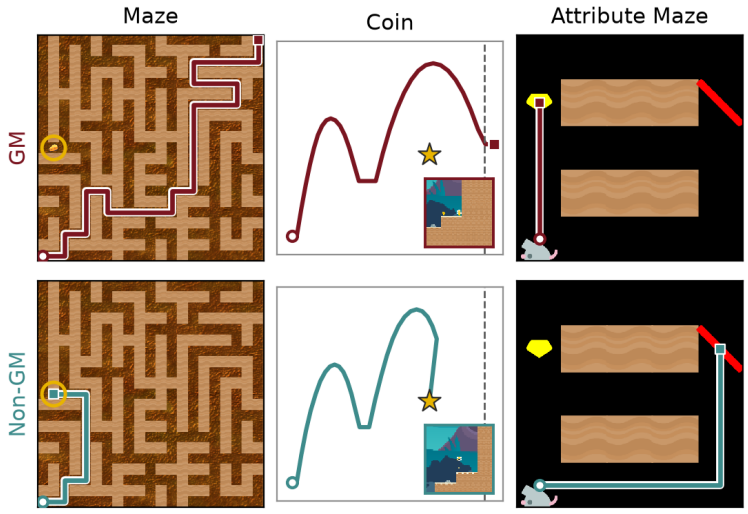

Figure 33: GM and Non-GM agents on the same OOD level, from the same reset, with greedy actions. Circles mark the start and squares the end of each trajectory. MAZE: the GM agent runs to the top-right corner and stays there until the time limit, and the Non-GM agent reaches the cheese (ringed) in 17 steps. COIN: agent paths in level coordinates, with the coin as a star, the end wall dashed and the last frame inset. The GM agent passes over the coin and ends at the end wall, and the Non-GM agent turns back and collects it. ATTRIBUTE MAZE (black background): the GM agent ends on the yellow gem (-10) in 3 steps, and the Non-GM agent takes the 7-step path to the red line ($+10$). On each level both agents navigate competently and differ only in where they go. Levels were selected by inspection for a GM-proxy and Non-GM-goal pair within the first 40 held-out levels (MAZE, ATTRIBUTE MAZE) or among levels whose coin is away from the end wall (COIN), so the figure shows the phenotype and does not estimate its rate (Table 23). The checkpoints are those of Appendix F.

At evaluation time, the cheese is placed away from the corner, turning the latent ambiguity into a competition between two spatial targets.

COIN: strategy ambiguity. A COIN state, as in Procgen CoinRun (Cobbe et al., 2019), contains the procedurally generated level geometry, the agent state, the coin location, and other level elements such as obstacles or hazards,

$$s = (\xi, x_{\text{agent}}, g_{\text{coin}}, z_{\text{level}}).$$

The agent again acts through a discrete control space whose effective movement component includes leftward and rightward motion together with vertical or non-spatial actions. For the analyses below, the important partition is between actions that move the agent to the right,

$$A_{\text{right}} \subseteq \mathcal{A},$$

and actions selected by the environment geometry as immediately coin-directed.

Reaching the coin produces terminal reward $R_{\text{coin}} = 10$,

$$r(s, a, s') = \begin{cases} R_{\text{coin}}, & \text{coin reached,} \\ 0, & \text{otherwise,} \end{cases}$$

while hazardous interactions can terminate the episode without reward.

During training, the coin sits at the right end of every level, so persistent rightward progress is a reliable strategy for eventually reaching it (Langosco et al., 2022). Unlike MAZE, the competing proxy is therefore not another object or destination. It is a policy regularity:

track the coin versus keep moving right.

At evaluation time, the coin is generated in a random section of the level, so the two can require different immediate actions.

ATTRIBUTE MAZE: cue ambiguity. ATTRIBUTE MAZE, the simplified Maze environment of Ramanauskas & Şimşek (2023) derived from Maze II of Langosco et al. (2022), augments the maze navigation problem with two candidate objects whose identities are defined by multiple visual attributes. Unlike MAZE, every level is a fixed 5×5 grid. A state contains the maze geometry, agent position, the locations of the two objects, and the attribute assignment of each object,

$$s = \left(\mathcal{G}_\xi, x_{\text{agent}}, \{(g_i, \sigma_i, \kappa_i)\}_{i=1}^{2}\right),$$

where g_i is the location of object i, σ_i its shape, and κ_i its colour.

The effective navigation action space is again cardinal movement plus non-spatial aliases. Reward is assigned according to which candidate object the agent reaches. In the canonical task, reaching the object carrying the intended attribute yields terminal reward $+R$, while reaching the competing object yields $-R$, with $R = 10$,

$$r(s, a, s') = \begin{cases} +R, & \text{intended candidate reached,} \\ -R, & \text{competing candidate reached,} \\ 0, & \text{otherwise.} \end{cases}$$

During training, the intended shape cue and the correlated colour cue occur together on the same rewarded object, a yellow line, while the other object is a blue gem. A successful policy therefore need not distinguish

$$\text{shape-based target selection} \qquad \text{from} \qquad \text{colour-based target selection.}$$

At evaluation time, the two attributes are assigned to different candidate objects, a red line ($+R$) and a yellow gem ($-R$). The conflict can therefore be created while holding the underlying maze geometry and candidate locations fixed. This differs qualitatively from MAZE and COIN, as the evaluation intervention changes which object carries a cue rather than where an object is located.

E EVALUATION BANKS

Every analysis scores the policy on a bank of states that is fixed before any outcome is computed on it. We count every bank in *states*. A state is one complete environment configuration, including the agent position, together with the observation it renders. Unless stated otherwise below, all checkpoints of both cohorts and the randomly initialized networks are scored on the identical states, so a cohort difference cannot come from a difference in the states. Table 18 lists the banks and the analyses that use them.

Bank	States	Probes	World edits	Fisher	Role	Geometry	Steering	Rewiring
MAZE								
Probe bank	2,787	✓						
Conflict bank	128[a]		✓	✓			✓	
Detour bank	1,500				✓			
Cloning bank	200,004							✓
COIN								
Probe bank	320	✓						
Conflict bank	600		✓	✓	✓		✓	
Cloning bank	500,352							✓
ATTRIBUTE MAZE								
Probe bank	4,608[b]	✓						
Conflict bank	196		✓	✓	✓	✓	✓	
Diversified bank	480					✓		
Cloning bank	499,998							✓

Table 18: **Evaluation banks.** Check marks show which analyses use each bank. Every conflict bank, the detour bank and the diversified bank hold states in strict conflict, $A_\star(s) \cap A_p(s) = \emptyset$ (Definition 9). [a]Shared sample. The same construction also gives 140–142 states per checkpoint for the world-edit table and 48 states for steering. [b]Per checkpoint and rollout seed.

E.1 CONFLICT BANKS

A conflict state is a state in strict conflict (Section A.4), where the routes to the intended goal and to the proxy need different first actions. Only at such states does the chosen action show which target the policy follows, so the world edits, the Fisher analysis, functional role, cue-competition geometry and steering all use conflict banks. Figure 34 shows one state from each conflict bank, and Figure 11 shows the world edits applied to them.

The MAZE conflict bank follows a greedy rollout and keeps the steps at which the shortest-path first actions toward the cheese and toward the reachable top-right extreme differ. It is drawn in three ways that differ only in whose rollout is followed and how many states each level contributes. The shared sample follows GM checkpoint 1080 from level 1,000,000 and keeps the first conflict of each level, which gives 128 states on which every policy is scored. The Fisher analysis and the world-edit KL of Table 1 use it, so that every policy is compared on identical states. The per-checkpoint sample follows each checkpoint's own rollout and keeps at most three conflicts per level, which gives 140–142 states per checkpoint on 47 to 55 levels. The world-edit table (Table 3) uses it, because it asks whether each policy reacts to an edit at the decisions it actually faces. The steering sample keeps the first conflict of each level on the rollouts of one GM and one Non-GM checkpoint, 24 states each, so that neither cohort is steered only on the other cohort's states.

The COIN conflict bank holds 600 states reached by rollouts of one GM and one Non-GM checkpoint (300 states each) and kept by level geometry alone, so no evaluated policy's response decides which states enter. Each state has two coin placements, the nearest free cell to the right of the agent and the nearest free cell to its left, both visible in the frame. The base state s holds the coin at the right placement and the edited state s_e at the left placement, which is one cell from the agent in every state.

The ATTRIBUTE MAZE conflict bank is read from the serialized game state of 40,000 levels. Their 2,041 strict-conflict levels reduce to 196 unique states, because the generator has only 11 wall layouts. In every state the agent stands at the $(0, 0)$ spawn, and the two candidates are a red line and

a yellow gem at equal distance. Black and textured observations are two renderings of the same states. The shared steering axis is fit on 146 of them and evaluated on the other 50, with line and gem mirror pairs kept on the same side of the split.

E.2 SPECIAL BANKS

Why MAZE has a detour bank. Functional role scores which target the policy's action favours, so it needs states that no cohort's behavior has chosen. The detour bank therefore takes its cells from level geometry alone. In each level it keeps up to 40 reachable cells at which the shortest-path first actions toward the cheese and toward the reachable top-right extreme differ and the two path lengths differ by at most two steps, and it reaches each cell by a scripted walk from the spawn. This gives 1,500 strict-conflict states on 618 levels, including the 168 detour states of Appendix B.2.1 after which the bank is named.

Why ATTRIBUTE MAZE has a diversified bank. Every conflict-bank state starts at $(0, 0)$, so every conflict is UP versus RIGHT, and bearing, immediate action and the sign of $d_{\text{comp}}(s)$ are one variable (Figure 34). The diversified bank writes the agent position into the serialized state, so that the agent stands at each of the 21 walkable cells and every pair of movement directions occurs as a conflict. This gives 480 strict-conflict states, on which cue-competition geometry can be tested beyond the single contrast of the conflict bank (Appendix B.3.2).

E.3 PROBE BANKS

A probe learns a concept only if its bank contains labelled variation in that concept, such as the cheese present and absent, the coin left and right of the agent, or an attribute on an object that does not carry the rewarded role. Conflict banks hold this variation fixed, so each environment has one probe bank built to make its probe targets separable (Table 19). The MAZE probe bank is scripted and independent of any policy. It places the agent at four cells in each of 200 levels and, for each placement, puts the cheese at up to three cells (the reachable top-right cell, a distant cell and a random cell) or removes it, so 800 of its 2,787 states hold no cheese. The COIN probe bank holds 320 states from 81 levels, collected on rollouts of GM checkpoint 6033 and balanced so that the coin lies left and right of the agent equally often at every stage of level progress. The ATTRIBUTE MAZE probe bank is a grid of 12 training and 6 held-out object configurations on disjoint level ranges, and each checkpoint is evaluated on its own sampled rollouts in these configurations. Its labels are constant within a configuration, so its effective sample is 12 and 6 configurations. In MAZE and COIN, level-grouped splits assign whole levels to either the fitting or the held-out partition, over five split seeds.

E.4 CLONING BANKS

Rewiring fits fresh downstream layers by behavior cloning on rollouts of GM and Non-GM driver policies, pooled across drivers. Each environment has one cloning bank (Table 18), on levels disjoint from the levels on which the rewired policies are evaluated (Appendix C.2).

E.5 PROPERTIES OF THE BANKS

The shared sample of the MAZE conflict bank was collected from a GM rollout, so Non-GM policies and initialized networks are evaluated partly outside their own state distribution. The COIN conflict bank holds a single left-versus-right contrast with the goal one cell away, so COIN results describe that contrast. The ATTRIBUTE MAZE diversified bank places the agent at cells that GM agents rarely reach in their own rollouts, so it tests the scope of the shared competition axis partly outside their state distribution.

Concept	Label	States	Split	Sites
MAZE				
Cheese presence	cheese present or removed	2,787	level-grouped	$h^{(1)}$–h_{fc}
Cheese bearing	cheese left or right of the agent	1,987	level-grouped	$h^{(1)}$–h_{fc}
Corner bearing	top-right extreme left or right of the agent	2,787	level-grouped	$h^{(1)}$–h_{fc}
COIN				
Coin presence	coin visible in the frame	320	level-grouped	$h^{(1)}$–h_{fc}
Coin bearing	horizontal offset from the agent, residualized on level progress[a]	320	level-grouped	$h^{(1)}$–h_{fc}
ATTRIBUTE MAZE				
Line presence L_{pres}	line shape present	4,608[b]	configuration grid	$h^{(1)}$–h_{fc}
Yellow presence Y_{pres}	yellow colour present	4,608[b]	configuration grid	$h^{(1)}$–h_{fc}
Line role L_{role}	rewarded object is a line	4,608[b]	configuration grid	$h^{(1)}$–h_{fc}
Yellow role Y_{role}	rewarded object is yellow	4,608[b]	configuration grid	$h^{(1)}$–h_{fc}

Table 19: **Probe banks.** Binary labels are scored by held-out ROC–AUC, and COIN bearing by held-out R^2. Bearing is taken relative to the agent. Sites run over $h^{(1)}, h^{(2)}, h^{(3)}, h_{\text{pre}}, h_{\text{fc}}$ (Appendix B.1). Cheese bearing uses the 1,987 states that hold the cheese. [a]The residualization is fit on the training partition only. [b]Per checkpoint and rollout seed, 3,072 training and 1,536 held-out states.

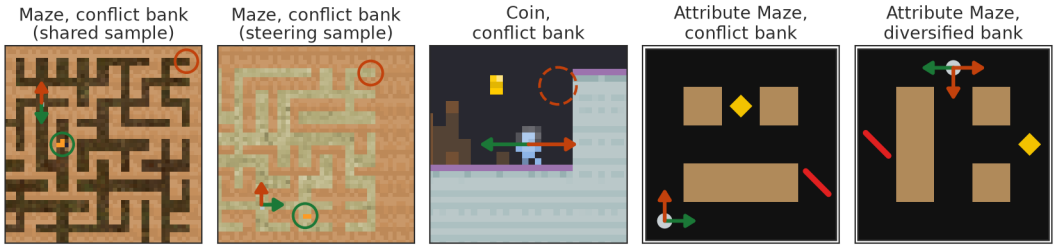

Figure 34: **One state from each conflict bank and from the diversified bank.** From left to right, the shared and steering samples of the MAZE conflict bank, the COIN conflict bank, the ATTRIBUTE MAZE conflict bank and the diversified bank. Green arrows are progress actions toward the intended goal, orange arrows toward the proxy (Definition 9). Circles mark the cheese and the top-right target. The COIN panel draws the coin at its edited placement left of the agent, and the dashed circle marks its base placement to the right. Each panel shows a single state. ATTRIBUTE MAZE panels are schematics of the stored state (red line, yellow gem, brown walls).

F BEHAVIORAL QUALIFICATION OF GOAL MISGENERALIZATION

Every experiment in this paper compares a GM cohort with a control cohort. This appendix shows from rollout outcomes alone, before any representation is inspected, that each checkpoint behaves as its label claims. We find a wide margin. Every GM checkpoint is capable and follows the proxy in 27% to 73% of out-of-distribution episodes, while every control follows it in at most 0.5%.

F.1 DEFINITIONS

Langosco et al. (2022) define goal misgeneralization as a failure in which an agent's capabilities generalize out of distribution while its goal does not, so that it competently pursues a goal other than the intended one. Shah et al. (2022) stress that this happens although the reward is specified correctly. Both definitions describe a training setup, and our experiments compare individual checkpoints, so we state a version that can be checked for one checkpoint at one training step. We use the notation of Appendix A, in which $\Gamma^\star$ and Γ^{prox} coincide on Λ_{train} and come apart on Λ_{OOD}.

Definition 10 (GM checkpoint)**.** *Let π be a checkpoint at a fixed training step in environment e, trained on Λ_{train}. Then π is* goal-misgeneralizing (GM) *if it meets both conditions below, with the statistics and thresholds of Table 21.*

- *(C)* Capability. *Its capability statistic meets the threshold of environment e, so that a low intended-goal success $J^\star(\pi; \Lambda_{\mathrm{OOD}})$ cannot be put down to incompetence.*
- *(P)* Proxy pursuit. *On Λ_{OOD} its proxy-reach rate is $\rho \geq 0.25$, and it passes the check of environment e.*

Definition 11 (Control (Non-GM) checkpoint)**.** *A checkpoint π in environment e is a* control*, written Non-GM, if it meets three conditions.*

- *(T)* Training. *It is trained on a distribution on which Γ^{prox} is uninformative about reward.*
- *(C)* Capability. *It meets condition (C) of Definition 10.*
- *(N)* No proxy pursuit. *On Λ_{OOD} its proxy-reach rate is $\rho \leq 0.01$.*

As (P) and (N) bound ρ on opposite sides, no checkpoint is both GM and a control. The two classes share the capability condition, so a GM checkpoint and its control differ in target and not in competence. A checkpoint that is incapable, or follows the proxy only occasionally ($0.01 < \rho < 0.25$), is neither and enters no cohort. The main text calls Non-GM checkpoints control agents.

F.2 METRICS

We roll out each checkpoint for N episodes from level resets on Λ_{OOD}, with greedy actions unless stated otherwise. Each episode falls into exactly one of three classes. It reaches the intended goal ($\Gamma^\star = 1$), it satisfies the proxy without the goal ($\Gamma^{\mathrm{prox}} = 1$, $\Gamma^\star = 0$), or it does neither. We write N_{goal} and N_{proxy} for the first two counts.

Proxy events. The proxy is a different kind of variable in each environment (Table 17), so N_{proxy} is counted per environment.

- COIN: the episode reaches the end wall of the level without collecting the coin (Cobbe et al., 2019; Langosco et al., 2022). Deaths and timeouts are neither goal nor proxy.
- ATTRIBUTE MAZE: the episode ends on the yellow gem (Ramanauskas & Şimşek, 2023).
- MAZE: the corner is a region, not a terminal event, so a corner-following episode runs until the time limit. We count an episode as proxy when it does not reach the cheese and either spends at least 12% of its steps in the top-right 28% of the observation along each axis or times out there. Levels on which the cheese itself lies in that region never count as proxy.

The MAZE count is used only for qualification and for Figure 35. The recovery analyses report MAZE reach alone (Appendices C.1 and C.2), since no episode ends at the corner.

Reach and its factorization. The intended-goal reach

$$\overline{\text{Reach}} = \frac{N_{\text{goal}}}{N} \tag{6}$$

estimates $J^\star(\pi; \Lambda_{\text{OOD}})$. It is the level-reset case of $\overline{\text{Reach}}^{(m)}$ in Appendix C.2, and Reach_e in Appendix B.1.1 is the same quantity started from an edited state. Reach factorizes exactly into two parts,

$$\overline{\text{Reach}} = D\, P_{\text{goal|dec}}, \qquad D = \frac{N_{\text{goal}} + N_{\text{proxy}}}{N}, \qquad P_{\text{goal|dec}} = \frac{N_{\text{goal}}}{N_{\text{goal}} + N_{\text{proxy}}}. \tag{7}$$

The decisiveness D is the share of episodes that commit to either target, and it reads capability on the conflict. The conditional preference $P_{\text{goal|dec}}$ is the share of committed episodes that pick the intended goal, and it reads target choice. These are the quantities of the recovery analyses (Appendices C.1.1 and C.2), where $P_{\text{goal|dec}}$ is also written $P(\text{goal} \mid \text{decisive})$.

Proxy-reach rate. The remaining share of committed episodes gives the proxy-reach rate

$$\rho = \frac{N_{\text{proxy}}}{N} = D\left(1 - P_{\text{goal|dec}}\right) = D - \overline{\text{Reach}}, \tag{8}$$

the share of all episodes that follow the proxy. It reads, on a single checkpoint, the half of the definition of Langosco et al. (2022) in which a capable agent pursues the proxy. $P_{\text{goal|dec}}$ asks which target wins when the policy commits, whereas ρ asks how often the proxy is pursued at all.

Capability statistics. In MAZE, a policy that follows the corner and a policy that cannot navigate both run out the clock on Λ_{OOD}. Capability is therefore read on held-out Λ_{train} levels, where the cheese sits in the corner, as the success rate S_{train}. The reward is 10 per success, so a mean return of at least 4.0 is the same as $S_{\text{train}} \geq 0.40$. In COIN it is read on Λ_{OOD} from the failure rate

$$F = \frac{N_{\text{died}} + N_{\text{timeout}}}{N}, \tag{9}$$

which in COIN equals $1 - D$. In ATTRIBUTE MAZE it is read from D directly.

COIN dependence. COIN adds one within-checkpoint intervention. We roll out the checkpoint on the same levels twice, once as usual and once with the coin sprite removed from the observation while the coin stays in the level, collectible and rewarded (Appendix H), and define

$$\text{dep} = \overline{\text{Reach}}(\text{coin visible}) - \overline{\text{Reach}}(\text{coin hidden}). \tag{10}$$

A policy that pursues the coin has $\text{dep} > 0$, since seeing the coin helps it get there. A negative value means the policy collects the coin less often when it can see it than when it is blind to it, so its coin collection is not directed at the coin.

Intervals. Proportions carry 95% Wilson intervals (Wilson, 1927) and dep carries a 95% Newcombe interval for a difference of two proportions (Newcombe, 1998). Cohort summaries are mean $\pm$ SD across checkpoints.

Native reach across protocols. The paper reports the reach of the unedited native policies in four places (Table 20). The values differ because each assay starts from different states and runs under its own action mode and step cap. Episodes that start from a level reset include the whole route to the conflict, whereas steering and world-edit rollouts start at a conflict state. In MAZE, for example, Non-GM reach is 0.415 from greedy resets but 0.871 from its own conflict states under sampled actions, and the steering forks give 0.694 under greedy actions. Each value is therefore compared only with values from the same row. In the two COIN rows that start in the conflict bank, the base state has the coin on the shortcut side, and the edited state has it one cell behind the agent (Appendix E).

Env.	Where used	GM	Non-GM	Start states per checkpoint	Actions	Cap
MAZE	Table 23	0.277 ± 0.02	0.415 ± 0.02	level resets, 100 OOD levels (Non-GM 1,000)	greedy	500
	Table 23		0.937 ± 0.01	level resets, 1,000 OOD levels	sampled	500
	Table 14	0.219 ± 0.02	0.347 ± 0.02	level resets, 300 held-out levels	greedy	1,000
	Reach_e, Appendix B.1.1	0.080 ± 0.00	0.871 ± 0.03	own 140 to 142 conflict states, cheese relocated	sampled	150
	Table 12	0.001 ± 0.00	0.567 ± 0.03	320 shared decision forks	greedy	200
COIN	Table 23	0.277 ± 0.03	0.885 ± 0.01	level resets, 200 OOD levels (Non-GM 50 to 200)	greedy	1,000
	Table 14	0.316 ± 0.01	0.896 ± 0.00	level resets, 1,000 held-out levels	greedy	1,000
	Reach_e, Appendix B.1.1	0.019 ± 0.01	0.179 ± 0.09	600 conflict-bank states, edited (coin one cell left)	greedy	200
	Table 12	0.018 ± 0.01	0.121 ± 0.05	591 conflict-bank states, edited (coin behind)	greedy	150
ATTR. MAZE	Tables 23, 14	0.358 ± 0.02	0.696 ± 0.02	level resets, 1,000 held-out levels	greedy	256
	Table 12	0.160 ± 0.06	0.813 ± 0.05	50 conflict-bank evaluation states	greedy	60

Table 20: **Native intended-goal reach under each protocol.** Intended-goal reach of the native policies, with weights and activations left unchanged, mean $\pm$ SD across the three checkpoints of each cohort, black background for ATTRIBUTE MAZE. Start states are counted per checkpoint, with one rollout per start state. Cap is the rollout step limit. The MAZE Reach_e row averages first within each level and then over levels.

Env.	Capability (C)	Proxy (P)	Check	Reported
MAZE	$S_{\mathrm{train}} \geq 0.40$ on held-out Λ_{train}	$\rho \geq 0.25$	none	D, $P_{\mathrm{goal\|dec}}$
COIN	$F < 0.45$ or $\overline{\mathrm{Reach}} \geq 0.35$	$\rho \geq 0.25$	$\mathrm{dep} < -0.05$	D, $P_{\mathrm{goal\|dec}}$
ATTR. MAZE	none	$\rho \geq 0.25$	$P_{\mathrm{goal\|dec}} < 0.5$, Wilson upper bound below 0.5, $N = 1000$	D
Control	as for GM	$\rho \leq 0.01$	trained with the proxy uninformative (T)	D, $P_{\mathrm{goal\|dec}}$

Table 21: **Qualification gates.** A checkpoint is GM when it meets the capability condition, the proxy condition and the check of its environment (Definition 10), and a control when it meets (T), (C) and $\rho \leq 0.01$ (Definition 11). S_{train} is the success rate on held-out training levels, $F = 1 - D$ the COIN failure rate, and every other statistic is read on Λ_{OOD} (Section F.2). In ATTRIBUTE MAZE the proxy condition already requires a quarter of all episodes to commit to the proxy, and every GM checkpoint has $D \geq 0.673$, so no separate capability threshold is set. Reported statistics enter no gate.

F.3 GATES

Table 21 gives the gate of each environment and Table 22 the per-checkpoint values for MAZE and COIN. The margin on ρ is wide. Every GM checkpoint has ρ between 0.270 and 0.729 with Wilson lower bound at least 0.193, and every control has $\rho \leq 0.005$ with Wilson upper bound at most 0.028 (0.071 for 7001@40M, scored on 50 episodes). So any cut between 0.005 and 0.270 gives the same cohorts. The corner count does not absorb incompetence, since the greedy MAZE controls run out the clock on about 60% of OOD levels and still have $\rho \leq 0.001$. In COIN, rendering the coin lowers its collection on all three GM checkpoints, with dep intervals below zero, whereas it raises collection for the control 7001@40M. $P_{\mathrm{goal|dec}}$ is reported and not gated in MAZE and COIN. The GM checkpoints there lack the intended-goal preference of the controls, with upper bounds at most 0.643 against control values at least 0.994. With 56 to 125 decisive episodes per checkpoint the data cannot show that each of them prefers the proxy, and we do not claim it.

F.4 GOAL PURSUIT IS GRADED

Figure 35 places every scored checkpoint on the $(D, P_{\mathrm{goal|dec}})$ plane, and goal pursuit there is graded rather than binary. Labels change within a run, as COIN seed 6033 moves from $P_{\mathrm{goal|dec}} = 0.384$ at 20M to 0.788 at 200M at similar D (0.625 and 0.685). Labels also spread across seeds of one

Env.	Checkpoint	Capability	ρ	dep	$P_{\text{goal}\mid\text{dec}}$
MAZE	1080@200M	0.710	0.270 [0.193, 0.364]		0.518 [0.390, 0.643]
	1081@200M	0.840	0.370 [0.282, 0.468]		0.439 [0.326, 0.559]
	1082@200M	0.740	0.350 [0.264, 0.447]		0.417 [0.301, 0.543]
	Non-GM, greedy	0.370 to 0.416	≤ 0.001		≥ 0.998
	Non-GM, sampled	0.926 to 0.938	≤ 0.002		≥ 0.998
COIN	6033@20M	0.375	0.385 [0.320, 0.454]	-0.135 [-0.222, -0.045]	0.384 [0.303, 0.472]
	6037@120M	0.400	0.300 [0.241, 0.367]	-0.130 [-0.221, -0.036]	0.500 [0.412, 0.588]
	6051@55M	0.395	0.315 [0.255, 0.382]	-0.100 [-0.190, -0.007]	0.479 [0.392, 0.568]
	Non-GM, 7001–7003	≤ 0.120	≤ 0.005	$+0.219$ [$+0.184$, $+0.254$]	≥ 0.994

Table 22: **Per-checkpoint evidence for the MAZE and COIN labels.** Point estimates with 95% Wilson intervals, and a 95% Newcombe interval for dep. $P_{\text{goal}\mid\text{dec}}$ is reported and enters no gate. The capability column is S_{train} for MAZE (threshold 0.40) and F for COIN (threshold 0.45). GM checkpoints use $N = 100$ (MAZE) and $N = 200$ (COIN) OOD episodes, and the COIN dep compares 200 episodes with the coin visible and 200 with it hidden on the same levels. Control rows give the range or extreme over the three checkpoints, and the control dep is available for 7001@40M only.

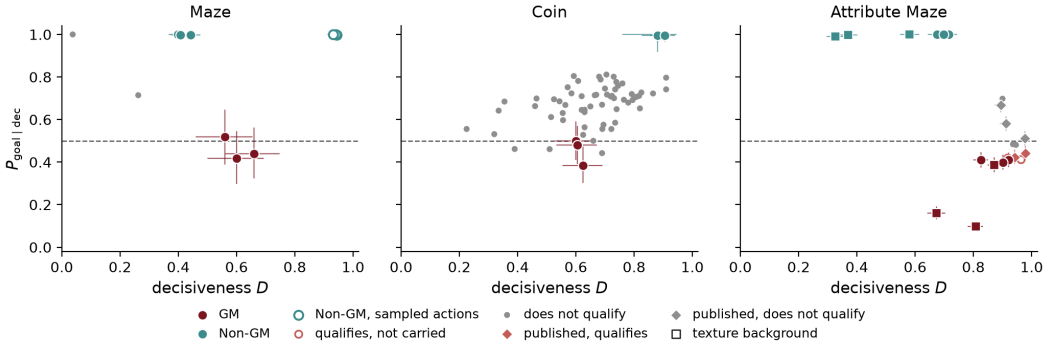

Figure 35: Goal pursuit is graded. Each point is one checkpoint at one training step, placed by decisiveness D and conditional intended-goal preference $P_{\text{goal}\mid\text{dec}}$ on Λ_{OOD}, with 95% Wilson intervals on both axes for cohort and published checkpoints. The dashed line marks $P_{\text{goal}\mid\text{dec}} = 0.5$, the ATTRIBUTE MAZE check. MAZE: GM seeds at 200M ($N = 100$), seed 1080 at 20M and 80M ($N = 80$), and Non-GM checkpoints under greedy and sampled actions ($N = 1000$). COIN: every greedy checkpoint scored in the GM training regime (eight seeds, 1M to 200M steps, $N = 200$ or 1000) and the Non-GM checkpoints at 40M. ATTRIBUTE MAZE: all eight black seeds of the GM training condition, the three texture seeds, the six Non-GM checkpoints and the five published checkpoints ($N = 1000$ each). Open red circles are checkpoints outside the cohorts that pass every gate of their environment (Table 21), namely ATTRIBUTE MAZE seeds 6312 and 6314. COIN 6036@5M has $\rho \geq 0.25$ and $P_{\text{goal}\mid\text{dec}} < 0.5$ but fails the dependence check (dep $= -0.010$), so it is marked as not qualifying.

condition, as the eight black ATTRIBUTE MAZE seeds span 0.399 to 0.698 at $D \geq 0.826$. Strict GM is rare, and of the 63 greedy COIN checkpoints scored in the GM training regime only the three cohort checkpoints pass the gate. The grading has a simple origin. As $\Gamma^\star$ and Γ^{prox} coincide on Λ_{train}, reward never selects between them and nothing forces a policy to commit to one, so intermediate values are what training should produce. Therefore selecting on the gate is part of the definition and not a filter laid over it, since a cohort drawn from one training condition would mix chance-level and goal-preferring checkpoints, and a mechanistic null on it would reflect that mixture.

F.5 PUBLISHED CHECKPOINTS

Ramanauskas & Şimşek (2023) released five ATTRIBUTE MAZE checkpoints (Ramanauskas, 2023), trained on a single yellow line on a black background for 5M steps, whereas our GM agents train on a yellow line against a blue gem (Appendix D). Scored on our evaluation, two of the five qualify, 7257 at 0.424 [0.392, 0.455] and 3561 at 0.443 [0.412, 0.474]. 3959 is at chance (0.511 [0.480, 0.543]),

Env.	Cohort	Checkpoints	D	$P_{\text{goal}\mid\text{dec}}$	ρ	Pass
MAZE	GM	1080–1082	0.607 ± 0.05	0.458 ± 0.05	0.330 ± 0.05	3/3
	Non-GM (greedy)	1090–1092	0.416 ± 0.02	0.998 ± 0.00	0.001 ± 0.00	1/3
	Non-GM (sampled)	1090–1092	0.938 ± 0.01	0.999 ± 0.00	0.001 ± 0.00	3/3
COIN	GM	6033, 6037, 6051	0.610 ± 0.01	0.454 ± 0.06	0.333 ± 0.05	3/3
	Non-GM	7001–7003	0.888 ± 0.01	0.996 ± 0.00	0.003 ± 0.00	3/3
ATTR. MAZE	GM, black	6313, 6315, 6317	0.883 ± 0.05	0.406 ± 0.01	0.525 ± 0.03	3/3
	Non-GM, black	4371–4373	0.696 ± 0.02	1.000 ± 0.00	0.000 ± 0.00	3/3
	GM, texture	6301–6303	0.784 ± 0.10	0.214 ± 0.15	0.610 ± 0.10	3/3
	Non-GM, texture	4361–4363	0.426 ± 0.14	0.996 ± 0.00	0.001 ± 0.00	3/3
	Published	five seeds	0.941 ± 0.04	0.525 ± 0.10	0.449 ± 0.11	2/5

Table 23: **Behavioral cohorts** on Λ_{OOD}, greedy actions unless marked sampled. Mean $\pm$ SD across the three checkpoints of each cohort (five for the published seeds). Pass counts checkpoints that meet the gate of their label (Table 21). Intended-goal reach is $D\,P_{\text{goal}\mid\text{dec}}$ and $\rho = D\,(1 - P_{\text{goal}\mid\text{dec}})$ (Section F.2). Episodes per checkpoint: 100 (MAZE GM), 1000 (MAZE Non-GM and ATTRIBUTE MAZE), 200 (COIN, 50 for 7001). MAZE training-level success, the capability statistic of its gate, is 0.763 ± 0.07 for GM and 0.394 ± 0.02 (greedy) or 0.933 ± 0.01 (sampled) for Non-GM. The published seeds are a reference population and enter no other analysis.

and 683 and 8038 prefer the intended object ($0.581\ [0.549, 0.613]$ and $0.667\ [0.636, 0.697]$). All five are highly decisive ($D \geq 0.895$) and follow the proxy often ($\rho \geq 0.298$), so the three that do not qualify fail on target choice and not on capability. The difference lies in the criterion. Ramanauskas & Şimşek (2023) establish goal misgeneralization for a training setup, as Langosco et al. (2022) and Shah et al. (2022) do, whereas we test each checkpoint on its own and require the interval to exclude chance, so the published labels stand under their own criterion. Every ATTRIBUTE MAZE GM checkpoint we study meets ours, with $P_{\text{goal}\mid\text{dec}}$ from 0.098 to 0.409.

F.6 THE COHORTS WE STUDY

Table 23 lists the cohorts used in every experiment that follows, and Figure 33 in Appendix D shows one GM and one Non-GM checkpoint from each on the same OOD level. Each Non-GM checkpoint is trained with the proxy uninformative about reward. The black ATTRIBUTE MAZE cohort holds the three lowest $P_{\text{goal}\mid\text{dec}}$ of the five black seeds that pass the gate. The MAZE controls meet the capability gate under sampled actions, whereas under greedy actions two of the three fall just below it (S_{train} of 0.370 and 0.396), so the capability match between the MAZE cohorts rests on sampled actions. The texture ATTRIBUTE MAZE controls commit less often than their GM counterparts ($D = 0.426 \pm 0.14$ against 0.784 ± 0.10), so comparisons with them rest on $P_{\text{goal}\mid\text{dec}}$ rather than on reach. In COIN the learning rate decays linearly to zero, so the GM checkpoints, taken 20M to 120M steps into 200M-step runs, carry a higher learning rate than the Non-GM checkpoints at the end of 40M-step runs.

G Related work in detail

Goal misgeneralization. Langosco et al. (2022) give the first explicit demonstrations of goal misgeneralization in deep reinforcement learning, and all three of our environments build on theirs. We rename the environments for readability. MAZE is their Maze I (cheese in the top-right corner), COIN is their CoinRun experiment (Procgen CoinRun with the coin at the right end during training), and ATTRIBUTE MAZE is the simplified Maze environment of Ramanauskas & Şimşek (2023), itself a simplification of their Maze II (Appendix D). Shah et al. (2022) argue that a correct specification does not guarantee a correct goal, with examples well beyond reinforcement learning. Langosco et al. (2022) also sort their failures into directional proxies (CoinRun), location proxies (Maze I, our MAZE) and observation ambiguity (Maze II, colour versus shape). Our three regimes line up with these kinds one to one: the policy shortcut in COIN with the directional proxy, functional target substitution in MAZE with the location proxy, and cue competition in ATTRIBUTE MAZE with observation ambiguity. Their CoinRun critic values the end of the level while the actor keeps moving right past it, so a directional and a location proxy can coexist in one agent, and we do not claim that COIN separates a rightward rule from endpoint-directed behavior. Dorn et al. (2023) frame goal misgeneralization as implicit goal conditioning, and mitigation work changes training or oversight, through minimax-regret level design (Abdel Sadek et al., 2025) or a mentor the agent can ask for help (Trinh et al., 2024). We build on the behavioral split of Langosco et al. (2022) and supply its internal counterpart: for each kind of proxy we locate where inside the policy the intended goal drops out, and show that this location matches which post-hoc recovery works.

Goals in reinforcement learning and reward hacking. The word "goal" is used in different senses across the reinforcement-learning literature. In goal-conditioned reinforcement learning a goal is an input to the value function or policy (Schaul et al., 2015), often a parametrized region of the state space to be reached (Florensa et al., 2018), and Colas et al. (2022) define a goal as a goal embedding paired with a goal-achievement function, which is a goal-conditioned reward. There the reward defines the goal. In the goal-misgeneralization literature the goal is a task-level outcome, and the reward is a learning signal that marks where that outcome was reached, so a correct reward can leave the pursued goal underdetermined (Langosco et al., 2022; Shah et al., 2022). We use the second sense (Definition 1). Our agents receive no goal input, and the goal we look for is whatever the trained network treats as its target. Reward hacking, also called specification gaming, is an adjacent but different failure, in which agents exploit gaps in a misspecified reward and gain proxy reward at the cost of true reward (Pan et al., 2022). Laidlaw et al. (2025) define it through a proxy reward that correlates with the true reward on the states of a reference policy and stops correlating as optimization moves the policy away. In goal misgeneralization the reward is fixed and correct, the coupling that breaks is between the intended goal and another regularity of the training levels, it breaks because the test levels change, and the agent stays capable while it pursues the wrong target. Our diagnostics follow this distinction. They keep the reward fixed and change the world, through shifted levels and world edits, rather than comparing a proxy reward with a true one.

Interpretability of RL agents. Hilton et al. (2020) interpret and edit a CoinRun agent with a different convolutional architecture from ours. Mini et al. (2023) dissect a MAZE policy trained with the cheese anywhere in the top-right 5×5 region. They find channels that track the cheese location, partially retarget the policy by editing them, and show that subtracting a cheese vector suppresses cheese-seeking while adding it does not increase it. Our GM agents are trained with the cheese on the exact corner cell, a regime they set aside because such agents did not pursue cheese outside the top-right 5×5 region, and our network is more than five times smaller. Our probe null therefore concerns a more strongly misgeneralizing regime and our probe sites, several of them globally average pooled, and does not contradict their finding. Their asymmetry is consistent with our MAZE steering, where the anticorrective push removes Non-GM cheese pursuit and no push creates it in GM agents (Appendix C.1.1). Ramanauskas & Şimşek (2023) simplify Maze II into a 5×5 maze without dead ends and with pure-colour sprites, the simplified Maze environment that we call ATTRIBUTE MAZE, and find that the preferred cue depends on the seed and the background, as in our cohorts. They attribute the colour preference to goal detection through one specific colour channel. Our swaps do not rule out that account. Exchanging sprites also changes which RGB channels carry line-shaped pixels, so our shape and colour cues could be one channel-specific shape detector rather than two separate features. Unlike theirs, our GM agents also train with a penalized

second object. Few studies of RL agents test whether an internal representation is used rather than only decodable. Bush et al. (2025) probe a model-free Sokoban agent for planning concepts and show through interventions on its representations that the resulting plans change its actions, and the edits of Hilton et al. (2020) and Mini et al. (2023) test use in the same causal sense. Trinh et al. (2024) report evidence that the internal state of their CoinRun agent does not represent the coin at all. We find coin bearing decodable in both COIN cohorts, well above randomly initialized networks, with different checkpoints and a different probe, so the two findings are not in direct conflict. We differ from these single-agent case studies by applying one set of assays to three environments with matched Non-GM controls, which is what shows that one behavioral label covers different internal failures.

Representation versus use: probes, interventions and Fisher sensitivity. Linear probes measure what an intermediate layer makes linearly available (Alain & Bengio, 2017). Hewitt & Liang (2019) show that probe accuracy means little without a control task, and untrained networks already expose some features linearly (Hermann & Lampinen, 2020). Our randomly initialized networks and input-level controls follow that logic, as control models and control inputs rather than control tasks. Decodability does not establish use. Elazar et al. (2021) test use by removing a property from the representation and measuring behavior, and interchange interventions set an internal variable to its value under another input (Geiger et al., 2021). Our world edits are the input-level analogue, with swaps as the canonical counterfactual in ATTRIBUTE MAZE. The Fisher information is the local metric of the KL divergence (Amari, 1998; Martens, 2020), and the natural policy gradient uses the Fisher of a policy with respect to its parameters (Kakade, 2001). Trust-region policy optimization bounds the KL divergence between successive policies and builds its Fisher matrix as the Hessian of that divergence (Schulman et al., 2015), and PPO, which trains all our agents, keeps the same constraint in approximate form (Schulman et al., 2017). The policy Fisher is therefore the natural metric for whether the policy moves. To second order it measures the change of the action distribution itself, independently of the coordinates in which the change is made, and it is the quantity that these methods keep small during training. We build on this geometry but take the Fisher with respect to the activations read by the policy head, evaluate it along world-edit displacements, and split the sensitivity into how strongly the policy reads the edited direction and how far the edit moves the representation.

Steering and readout recovery. Activation addition steers a model by adding the activation difference between two contrasting inputs (Turner et al., 2023b), and Mini et al. (2023) apply it to the MAZE agent. Our direction is state-local, built from the world-edit pair at each conflict state, with anticorrective and random controls. Shortcut learning names decision rules that succeed in training and fail to transfer (Geirhos et al., 2020). We use "policy shortcut" more narrowly, for an action regularity rather than an input feature. In supervised classification with spurious correlations, retraining only the last layer restores robustness because the core features survive (Kirichenko et al., 2023). COIN is a counterexample in our setting. A fresh head on h_{fc} reaches the coin at a rate of 0.666, close to the untrained floor of 0.630, while relearning from h_{pre} reaches 0.843 (Appendix C.2). Depth and capacity change together ($528{,}399$ versus $3{,}855$ trainable parameters). A fixed random $2048 \to 256$ projection of h_{pre} followed by a head the size of the h_{fc} readout reaches 0.676 for GM and 0.825 for Non-GM agents (Table 15), so GM coin information does not survive compression to the bottleneck. Lee et al. (2023) show that the subset of layers that is best to fine-tune depends on the type of distribution shift. They compare layer subsets by accuracy after fine-tuning on a small labelled target dataset, whereas we freeze the representation at h_{pre} or h_{fc}, relearn only the computation downstream of it by behavior cloning, and evaluate the result in closed loop. Prior repairs of a proxy-following agent either continue RL training under a new reward, on the training levels with a reward model fit to LLM preferences for a maze agent biased toward the top-right region and paused at 32% of training (Nait El Barj & Sautory, 2024) or on unlabeled test levels with an inferred reward (Armstrong et al., 2023), or prune the proxy concept from a concept-bottleneck agent, which requires that architecture (Delfosse et al., 2024). Subtracting a top-right vector from a region-trained maze agent has little effect on its pull toward the top-right (Turner et al., 2023a), consistent with our MAZE steering, where no push creates cheese pursuit in GM agents. To our knowledge, no prior work compares post hoc repairs of a goal-misgeneralizing agent or relates which repair succeeds to where the intended goal drops out inside the agent. We differ from these methods by using recovery as a test of the diagnosis. Steering works only under cue competition, where the intended goal still

reaches the decision, and rewiring from h_{pre} works only in the two regimes where our probes find the goal at that depth.

H IMPLEMENTATION AND TRAINING DETAILS

Why we reuse the published recipes. Our aim is to explain the goal misgeneralization that Langosco et al. (2022) and Ramanauskas & Şimşek (2023) observed, not to find it again under new conditions. We therefore train every agent with the recipe of the paper that reported the failure, the code of Langosco et al. (2022) for MAZE and COIN and the code of Ramanauskas & Şimşek (2023) for ATTRIBUTE MAZE. A different recipe would add a second difference between our agents and theirs, and any mechanism we found could then be a property of that recipe rather than of the phenomenon. Within each environment, GM and Non-GM agents share this recipe and differ only in the training distribution.

Environments. All environments are Procgen games (Cobbe et al., 2020) with 64×64 RGB observations and 15 discrete actions. MAZE and COIN are the variants of Langosco et al. (2022). MAZE trains on their modified Maze game (hard mode, maze sides 3 to 25), in which `rand_region`$= k$ confines the cheese to the top-right $k \times k$ block. Its OOD game, the original Procgen Maze, puts the cheese on any free cell. COIN trains on CoinRun (Cobbe et al., 2019) with the coin at the level end (`random_percent`$= 0$), and is evaluated on the variant of Langosco et al. (2022) that puts the coin in a random section. ATTRIBUTE MAZE uses the 5×5 easy-mode two-object maze of Ramanauskas & Şimşek (2023), in which one object gives $+10$ and the other -10 and the agent starts in the bottom-left cell.

Training. Every agent is trained from scratch with PPO (Schulman et al., 2017) and the `hard-500` preset (Table 24). MAZE and COIN use the code of Langosco et al. (2022), derived from Lee (2020), and ATTRIBUTE MAZE uses the code of Ramanauskas & Şimşek (2023). The two codebases share the network, the PPO update and the preset. The learning rate decays linearly to zero over the default run length, so COIN GM checkpoints, taken 20M to 120M steps into 200M-step runs, saw higher learning rates than Non-GM checkpoints at the end of 40M-step runs (Appendix F).

Checkpoints and controls. Non-GM controls change only the training distribution (Table 25, where `rand_region`$= 25$ covers every maze). ATTRIBUTE MAZE controls train on eight object pairs, 32 environments each, with line presence and yellow presence uncorrelated and two same-colour pairs that only shape separates. The line is rewarded whenever it appears, and no training pair sets a non-yellow line against a yellow gem, so the trap is new to both cohorts. COIN checkpoints were chosen by step during qualification. Every other checkpoint is the final save of its run.

Evaluation levels. MAZE qualification uses levels from 200,000 (training distribution) and 700,000 (OOD). COIN qualification uses levels from 1 (GM) and 700,000 (Non-GM). All other MAZE and COIN evaluation uses levels from 1,000,000. ATTRIBUTE MAZE trap preference and rewiring evaluation use levels from 100,000. The 196-state conflict bank reuses layouts of levels 2,003 to 41,991, inside the training range, so for these states only the object assignment is held out.

Network implementation. Section A.1 describes the network conceptually. Its implementation is the feed-forward IMPALA CNN of Espeholt et al. (2018) as written by Lee (2020). Each of the three blocks applies a 3×3 convolution with stride 1 and padding 1, a 3×3 max-pool with stride 2 and padding 1, and two residual units, with 16, 32 and 32 output channels, so the spatial size falls from 64 to 32, 16 and 8. Each residual unit computes $x + \mathrm{conv}(\mathrm{ReLU}(\mathrm{conv}(\mathrm{ReLU}(x))))$ with 3×3 convolutions that keep the channel count, and no ReLU sits between a block's first convolution and its max-pool. The only nonlinearities outside the residual units are the ReLU applied to $H^{(3)}$ before flattening into h_{pre} and the ReLU after the $2048 \to 256$ linear layer that gives h_{fc}. The policy head ($256 \to 15$, orthogonal initialization with gain 0.01) and the value head ($256 \to 1$, gain 1) are linear. The value head enters PPO training only through the value loss, and it is unused in the behavior-cloning rewiring of Appendix C.2. The convolutional trunk holds 97,600 parameters (9,728, 41,632 and 46,240 per block), the fully connected layer 524,544, the policy head 3,855 and the value head 257, for 626,256 in total.

	MAZE	COIN	ATTRIBUTE MAZE
Training game	cheese top-right	coin at level end	duel, see Table 25
OOD game	cheese anywhere	coin anywhere	duel, red line vs. yellow gem
Distribution mode	hard	hard	easy, 5×5 maze
Reward, time limit	$+10$, 500	$+10$, 1000	± 10, 500
Default run length	200M	200M (GM), 40M (Non-GM)	5M
Training levels	$[0, 10^5)$		
Environments $\times$ rollout	256×256, so 65,536 transitions per update		
Epochs, minibatches	3 epochs of 8 minibatches of 8,192		
γ, GAE (Schulman et al., 2016) λ, clip	0.999, 0.95, 0.2		
Optimizer	Adam, $\epsilon = 10^{-5}$, learning rate 5×10^{-4} decayed linearly to 0, no weight decay		
Loss coefficients	value 0.5, entropy 0.01, gradient-norm clip 0.5		
Normalization	advantages per rollout, rewards by running return scale (clip ± 10)		

Table 24: PPO training recipe. The lower block is the `hard-500` preset, identical for every agent. $\mathrm{M} = 10^6$ environment steps.

Env.	Cohort	Seed: checkpoint step	Training distribution
MAZE	GM	1080–1082: 200,015,872	cheese in top-right cell (`rand_region`$= 0$)
	Non-GM	1090–1092: 200,015,872	cheese anywhere (`rand_region`$= 25$)
COIN	GM	6033: 20,054,016, 6037: 120,061,952, 6051: 55,050,240	coin at level end
	Non-GM	7001–7003: 40,042,496	coin anywhere
ATTR. MAZE	GM black	6313, 6315, 6317: 5,046,272	yellow line ($+$) vs. blue gem ($-$), black
	Non-GM black	4371–4373: 5,046,272	eight decorrelated pairs, black
	GM texture	6301–6303: 5,046,272	yellow line ($+$) vs. blue gem ($-$), texture
	Non-GM texture	4361–4363: 5,046,272	eight decorrelated pairs, texture

Table 25: Checkpoint provenance for the cohorts of Table 23, in environment steps. Every run trained on a single GPU. Where the run logged it, wall-clock per run is 39.8 to 47.3 hours for MAZE GM, 55.0 for MAZE 1090, 35.5 and 61.3 for COIN 6033 and 6037, 10.0 for COIN 7002 and 0.6 to 1.5 for ATTRIBUTE MAZE. These times depend on the hardware, but a 200M-step MAZE or COIN run takes about two days on one GPU.

H.1 MODIFICATIONS TO THE TRAINING CODE

The analyses in this paper need to read and edit the state of a running episode, which neither fork supports beyond saving and restoring it. Procgen already generates each level deterministically from its index and can serialize and restore the full game state, so a matched counterfactual starts from a restored copy of the unedited state and changes only the edited object. We added the edits themselves to the game code, and none of them changes an episode that is not edited. In MAZE a call places or removes the cheese at a given cell and another turns a wall cell into a free cell, which gives the cheese relocations and the structural edit that opens the top-right region (Appendix B.1.1). The reward is then tied to the cell that currently holds the cheese, which in an unedited level is the cell it was generated on. In COIN a call places or removes the coin, and a game option hides the coin sprite while keeping its reward, which gives the coin relocations of Appendix B.1.1 and the visible-versus-hidden dependence score of Appendix F. Both games also report the agent cell and the cheese or coin position at every step, which supply the probe targets and the agent-position controls of Appendix B.1. In ATTRIBUTE MAZE the duel game of Ramanauskas & Şimşek (2023) already takes the rewarded and the penalized object as parameters, and we added calls that read and set the cell of each object. These calls build the shape and colour swaps on a fixed layout, and the diversified bank places the agent by writing its position into the restored state (Appendix E). Our ATTRIBUTE MAZE trainers split the 256 parallel environments into equal groups, one per object pair, so that a single PPO run sees all pairs of Table 25 at once. Beyond this, the training code gained checkpoint resumption and loss logging, which leave the PPO update unchanged. All code needed to reproduce the agents and analyses will be released with the paper.