BAD OUTCOMES NEED NOT MEAN BAD ACTIONS: COUNTERFACTUALLY VERIFIED PAIN MEMORY FOR POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Bad outcomes do not always mean bad actions. Policy-gradient methods update policies using advantage estimates and TD errors, while failure-memory approaches often retain experiences based on observed failures or their severity. When exogenous randomness dominates the effects of individual actions, these signals can mistake bad luck for poor decisions, while later policy updates can erase useful lessons from rare but consequential events. We propose *PAIN-PPO*, a two-channel framework built on Proximal Policy Optimization (PPO). The Advantage channel learns from current rollouts, while the Pain channel retains only corrections verified through counterfactual replay, without modifying rewards or advantage estimates. For a candidate failure, we intervene on the executed action and replay the episode in closed loop under the same recorded conditions. A correction enters memory only if the alternative action improves the outcome. Its influence grows when the same corrective direction recurs across distinct realizations. Each stored lesson defines a local, revisable, one-sided retention constraint whose gradient is combined with the PPO gradient only when needed to resist forgetting. We evaluate PAIN-PPO in both controlled simulations and real-world financial markets. In simulation, PAIN-PPO distinguishes actionable policy errors from adverse outcomes driven by exogenous randomness, and its gains largely disappear when counterfactual verification is removed, while selective admission further filters corrections that do not recur consistently. On held-out financial market data, PAIN-PPO demonstrates promising lower-tail risk performance under the prescribed mean-cost budget. Together, these results show that counterfactually verified memory can improve tail-sensitive policy learning by retaining rare but actionable corrections without conflating unfavorable outcomes with poor decisions.

1 INTRODUCTION

A reinforcement learning agent learns from the outcomes of its actions. In policy-gradient methods every return in a batch enters the update, so a severe outcome pushes the policy away from whatever preceded it. In many environments, however, most of what happens to an agent is not caused by its behavior: a market crash moves the cost of an order by far more than any placement decision could, and an intensive-care patient may die under the best available treatment. The question that matters is what, if anything, could have been done differently. A wrong answer is costly either way, since the agent can give up useful actions without preventing the failure or repeat a correctable mistake.

The previous literature handles severe outcomes in three ways (Appendix B). Risk-sensitive and constrained reinforcement learning reshapes the objective through CVaR criteria, distributional critics, or constraints (Rockafellar & Uryasev, 2000; Tamar et al., 2015; Chow et al., 2018; Achiam et al., 2017; Tessler et al., 2019; Bellemare et al., 2017; Dabney et al., 2018), so it decides which outcomes matter but not whose fault they were. Failure-memory methods keep the bad episodes and steer away from them, by sampling stored transitions by temporal-difference error (Schaul et al., 2016), reusing past experiences directly (Pritzel et al., 2017; Lin et al., 2018), or retrieving stored failures to avoid repeating them (Miao, 2026). Causal and counterfactual methods ask whether an action mattered at all, by inferring the exogenous scenario behind a logged episode (Buesing et al., 2019; Oberst &

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.

Sontag, 2019), relabeling goals in hindsight (Andrychowicz et al., 2017), or separating skill from luck inside the current estimator (Mesnard et al., 2021).

The difficulty is that a severe outcome does not say whose fault it was. No action could have changed it, or a different action may have changed it on that one realization of the environment but not on others, or the agent's action may have caused it and the same change would repair it again elsewhere. Only the third is worth remembering, and the return alone does not tell the three apart. Two properties of the environment, rather than of any particular algorithm, make the separation hard. One (P1) is low signal-to-noise with exogenous catastrophes, where severe failure does not imply poor action choices (Sections 5.1, 5.3, and 5.5). A failure is exogenous when no alternative action changes its outcome under the same realization of the randomness (Section 2.2), and in the settings we study such failures make up a large part of the tail. The other (P2) is sparse action value, where only a small fraction of states are action-critical, so the lesson from such a state is rare and an on-policy update (Schulman et al., 2017) averages it away within a hundred updates (Section 5.2). Both can be measured before any method is run (Appendix G), and the arenas of Section 5 are built around them. Storing failures is not enough, since a wrong lesson in memory can steer every later update and a right one can still be lost.

We introduce **PAIN** (Paired Attribution of Improvable Negatives), an attribution-and-memory framework for policy optimization, and instantiate it with PPO as **PAIN-PPO**. Its core principle is *attribute first, remember second*. Outcome severity is used only to select failures for inspection, after which PAIN-PPO holds the realized exogenous randomness fixed and uses same-realization counterfactual comparisons to identify actionable corrections. A correction receives initial support only if changing the agent's action would improve the outcome by a nontrivial margin (Pearl, 2009, Ch. 7), and gains lasting influence only when its direction recurs across distinct realizations. The Advantage channel learns from the current batch of policies, while a separate Pain channel retains verified corrections throughout the updates as bounded one-sided constraints that preserve part of each verified improvement without attaching the policy to a previous target (Section 3). PAIN-PPO therefore remembers *verified corrections*, rather than failure trajectories or reshaped rewards. Reward, critic, and advantage estimation remain unchanged, so PAIN-PPO is exactly reduced to standard PPO when the correction memory is empty.

Theoretically, we first show that the canonical PAIN-PPO update can be written as a PPO step with an adaptive one-sided regularizer induced by the current memory (Proposition 1). Although this regularizer evolves with memory and the effective objective is therefore time-varying, we still establish a finite-time bound on the update residual (Theorem 1). We further show that, in a shared-parameter policy class with gradient interference, retaining a verified correction changes the limiting policy (Proposition 2). Empirically, PAIN-PPO admits the avoidable failures with high precision where failures carry ground-truth labels, stays silent when nothing was avoidable, and reaches the risk-averse policy more reliably than Return Capping on the authors' own code. On real limit-order-book execution it lowers the tail of execution cost within a mean budget in three test months. There it beats a supervised selector trained on the same evidence even without the Pain channel, and the channel raises the tail gain further in every seed-month (Section 5).

2 Problem Setting

2.1 MDP and Policy Optimization Backbone

We consider a finite-horizon episodic Markov decision process (notation in Appendix A) with states s_t, actions a_t, rewards r_t, and discount factor $\gamma \in [0, 1]$, and a stochastic policy $\pi_\theta(a \mid s)$. The return from time t is $G_t = \sum_{k \geq t} \gamma^{k-t} r_k$. We use PPO (Schulman et al., 2017) as the base optimizer. At update u, it collects an on-policy batch $\mathcal{B}_u$ under π_{θ_u}. With importance ratio $r_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_u}(a_t \mid s_t)$, advantage estimates $\widehat{A}_t$, and clipping radius $\epsilon_{\text{clip}} > 0$, PPO optimizes

$$L^{\text{clip}}(\theta) = \mathbb{E}_{t \in \mathcal{B}_u} \left[\min \left(r_t(\theta)\, \widehat{A}_t,\ \text{clip}(r_t(\theta),\, 1 - \epsilon_{\text{clip}},\, 1 + \epsilon_{\text{clip}})\, \widehat{A}_t \right) \right].$$

Using the loss convention $L_{\text{PPO}} = -L^{\text{clip}}$, we write $g_{\text{A}} = \nabla_\theta L_{\text{PPO}}$.

Standard policy optimization maximizes expected return, $J(\theta) = \mathbb{E}_{\pi_\theta}[G_0]$. Our tail-sensitive experiments use the lower-tail Conditional Value-at-Risk (CVaR) of the return at tail level $\alpha \in (0, 1)$,

$$\mathrm{CVaR}_\alpha(G) = \sup_{\nu \in \mathbb{R}} \left\{ \nu + \frac{1}{\alpha} \mathbb{E}[\min(G - \nu, 0)] \right\},$$

where larger returns are better and $\alpha = 0.05$. The execution experiment (Section 5.5) maximizes $\mathrm{CVaR}_\alpha(G_\theta)$ subject to the mean budget $\mathbb{E}[G_\theta] \geq J_{\mathrm{ref}} - \varepsilon$, where J_{ref} is the reference-policy return and $\varepsilon \geq 0$ is the allowed mean-performance loss. It uses the Lagrangian advantage $\widehat{A}_t^{\mathrm{risk}} = \kappa_{\mathrm{tail}} \widehat{A}_t^{\mathrm{CVaR}} + \lambda \widehat{A}_t^{\mathrm{mean}}$ with $\kappa_{\mathrm{tail}} > 0$ and scaled dual coefficient $\lambda \geq 0$, so the multiplier of the unit-CVaR objective is $\lambda / \kappa_{\mathrm{tail}}$. Appendix H gives the estimator and the dual update. The risk-sensitive objective decides which outcomes the PPO update emphasizes, while PAIN-PPO decides which corrections drawn from them stay active later, so PAIN works with either objective.

2.2 Counterfactual Replay Interface

PAIN-PPO requires a counterfactual replay interface during training. Write the dynamics as $s_{t+1} = F(s_t, a_t, \zeta_t)$, where ζ_t is exogenous randomness. For verification, we fix the continuation policy and rule of the inspected episode and reuse its recorded realization ω, including base sampling variables if the continuation is stochastic. Each replay is then deterministic given the intervention state.

For a candidate action a at time t, let $J(a; \omega, t)$ denote the return obtained by replacing the factual action a_0 with a and regenerating the subsequent trajectory under this fixed continuation rule. Replaying both actions under the same ω defines the *same-realization counterfactual gain*

$$\Delta_t(a; a_0, \omega) = J(a; \omega, t) - J(a_0; \omega, t). \tag{1}$$

The interface can be implemented with replayable simulators, structural causal models, or historical replay environments. Counterfactual queries are required only during training, with replay details and assumptions given in Appendix H. Pairing is an efficiency device, not a requirement (Appendix F.1.4).

3 PAIN: Counterfactually Verified Correction Memory

PAIN (Paired Attribution of Improvable Negatives) is a correction layer for policy optimization that verifies and retains reliable local action corrections across updates. It leaves the policy objective, reward, critic, and advantage estimator unchanged; combined with PPO, it yields **PAIN-PPO**. PAIN stores recurring verified corrections as one-sided constraints and modifies the PPO update only when these constraints are violated. As shown in Figure 1, it consists of verification and consolidation (Section 3.1), one-sided retention (Section 3.2), and selective update correction (Section 3.3).

3.1 Counterfactually Verified Corrections

Outcome severity determines which episodes are inspected, but not which actions are penalized. At a selected decision t, let a_0 be the executed action and $a_1, \ldots, a_K$ be K local candidate actions. The replay interface in Section 2.2 gives $G_i := \Delta_t(a_i; a_0, \omega)$ as the return improvement from replacing a_0 with a_i under the same recorded realization ω.

For continuous actions, we remove probe-set-wide improvement and changes below the verifier tolerance $\epsilon_{\mathrm{noise}} \geq 0$, then aggregate the remaining evidence:

$$\begin{aligned} h_i &= \Big[G_i - \max\big(0, \mathrm{median}_\ell\, G_\ell\big) - \epsilon_{\mathrm{noise}} \Big]_+, \\ a^\star &= a_0 + \frac{\sum_{i=1}^K h_i (a_i - a_0)}{\sum_{i=1}^K h_i + \epsilon_{\mathrm{num}}}. \end{aligned} \tag{2}$$

Here $[x]_+ = \max(x, 0)$ and $\epsilon_{\mathrm{num}} > 0$ stabilizes division. If all $h_i = 0$, this decision provides no correction. Otherwise, stronger excess gains receive more weight and opposing displacements cancel. Thus, $a^\star$ is a local corrective target, not an estimate of the optimal action. Discrete actions retain the verified action comparison directly, without averaging action labels.

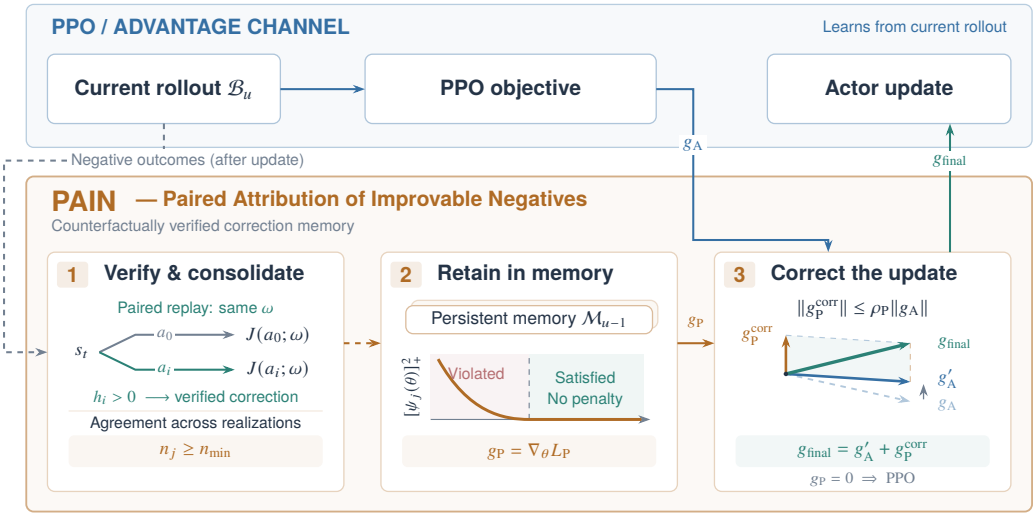


Figure 1: **PAIN-PPO as a two-channel policy update.** The PPO/Advantage channel learns from the current rollout. PAIN verifies and consolidates corrections, retains them as one-sided constraints, and modifies the PPO gradient when authorized constraints are violated through bounded correction and conflict attenuation. Dashed arrows denote evidence refresh after the actor update.

A single realization supplies only provisional evidence. Corrections at similar states with consistent directions are grouped into a memory entry. Let n_j count the distinct realizations supporting entry j, and authorize the entry only when $n_j \geq n_{\min} \geq 2$. Replaying the same realization does not increase n_j, and conflicting evidence weakens the entry. Pending entries have no direct teaching or veto authority. Once authorized, their influence depends on evidence strength and recurrence. Evidence scoring, state matching, and authority weights are specified in Appendix H.

3.2 Persistent One-Sided Constraints

Section 3.1 identifies a reliable change from the executed action toward a corrective target at a stored state s_j. We turn it into a constraint that preserves part of that change without requiring the policy to reproduce the target exactly. For continuous actions, let $z_\theta(s)$ be the deterministic policy output used to measure corrections (e.g., its mean action). Express the executed action and $a^\star$ in these same coordinates as z_j^{bad} and z_j^{target}, using a fixed action-to-output map. This map is the identity for direct action outputs; other parameterizations are detailed in Appendix H. Define $d_j = z_j^{\text{target}} - z_j^{\text{bad}}$. Exclude zero displacements; otherwise, set $\hat{d}_j = d_j / \|d_j\|$. For a retention fraction $\eta \in (0, 1]$, define

$$\psi_j(\theta) = \eta \|d_j\| - \left\langle z_\theta(s_j) - z_j^{\text{bad}}, \hat{d}_j \right\rangle,$$
$$L_{\text{P}}(\theta) = \frac{\sum_{j \in \mathcal{M}_{u-1}} w_j^{\text{auth}} [\psi_j(\theta)]_+^2}{\sum_{j \in \mathcal{M}_{u-1}} w_j^{\text{persist}} + \epsilon_{\text{num}}}. \tag{3}$$

The condition $\psi_j(\theta) \leq 0$ retains at least an η-fraction of the verified displacement. This constraint is one-sided: further progress along the verified direction incurs no penalty. Evaluating L_{P} on stored states allows corrections to remain active without an on-policy revisit. The nonnegative weights w_j^{auth} and w_j^{persist} control current authority and persistence, and Appendix H gives their updates and the discrete-action constraint.

3.3 PAIN-Corrected Policy Update

Let $g_{\text{P}} = \nabla_\theta L_{\text{P}}$ denote the PAIN correction gradient induced by memory and g_{A} the PPO gradient from the current rollout. If no authorized constraint is violated, $g_{\text{P}} = 0$ and the update is ordinary PPO. Otherwise, PAIN combines a bounded correction toward satisfying the constraints with attenuation of the PPO components that conflict with them.

For the canonical relative-norm-capped instantiation analyzed in Section 4, the update is

$$g_{\mathrm{P}}^{\mathrm{corr}} = \alpha_{\mathrm{P}} g_{\mathrm{P}} \min\left\{1, \frac{\rho_{\mathrm{P}}\|g_{\mathrm{A}}\|}{\|\alpha_{\mathrm{P}} g_{\mathrm{P}}\|}\right\}, \qquad g'_{\mathrm{A}} = g_{\mathrm{A}} + q\frac{[-\langle g_{\mathrm{A}}, g_{\mathrm{P}}\rangle]_+}{\|g_{\mathrm{P}}\|^2 + \epsilon_{\mathrm{num}}} g_{\mathrm{P}}, \tag{4}$$
$$g_{\mathrm{final}} = g'_{\mathrm{A}} + g_{\mathrm{P}}^{\mathrm{corr}}.$$

Set $g_{\mathrm{P}}^{\mathrm{corr}} = 0$ when $g_{\mathrm{P}} = 0$. The coefficient $\alpha_{\mathrm{P}} > 0$ sets the nominal correction strength, while $\rho_{\mathrm{P}} > 0$ limits its norm, giving $\|g_{\mathrm{P}}^{\mathrm{corr}}\| \le \rho_{\mathrm{P}}\|g_{\mathrm{A}}\|$. Under the loss-gradient convention, $\langle g_{\mathrm{A}}, g_{\mathrm{P}}\rangle < 0$ means PPO descent increases L_{P} to first order. Only then does g'_{A} attenuate the conflicting component, with strength $q \in [0, 1]$ set by memory authority and constraint violation. The veto is part of the full method and is on in execution, the sepsis simulator, and the Guarded Maze. It removes a median of at most about one percent of the PPO gradient norm, and in the sepsis simulator it carries corrections that are not yet learned across a shift in the training data (Appendix F.3). These operations modify the gradient, rather than enforce feasibility after a finite optimizer step. Appendix H.3 gives the more general scaling form and the arena-specific scaling choices, while Appendix H gives the rule for q and one full update iteration (Algorithm 1).

4 Theoretical Analysis

We characterize the canonical capped PAIN-PPO update in Eq. (4), establish a finite-time bound for this adaptive update, and give a shared-parameter instance in which retaining a verified correction changes the limiting return. Proofs and protocol details are in Appendix C.

4.1 Regularized Updates and a Finite-Time Bound

Let $g_{\mathrm{A},t} = \nabla_\theta L_{\mathrm{PPO},t}(\theta_t)$ denote the PPO gradient estimated from the on-policy batch at update t, and let $L_{\mathrm{P},t}$ be the one-sided constraint loss induced by the current correction memory.

Proposition 1 (Adaptive one-sided regularization)**.** *For $\alpha_{\mathrm{P}}, \rho_{\mathrm{P}} > 0$ and $q_t \in [0, 1]$, Eq. (4) can be written as*

$$\begin{aligned} g_{\mathrm{final},t} &= g_{\mathrm{A},t} + \beta_t \nabla L_{\mathrm{P},t}(\theta_t) \\ &= \nabla_\theta (L_{\mathrm{PPO},t} + \beta_t L_{\mathrm{P},t})\,(\theta_t), \qquad \beta_t \ge 0, \end{aligned}$$

where β_t is held fixed in differentiation. If all authorized constraints are satisfied, $\nabla L_{\mathrm{P},t}(\theta_t) = 0$ and the update reduces to PPO.

Appendix C.1 gives β_t, the norm bound from the relative cap, and the proof. PAIN therefore does not optimize a fixed objective, since each update has an adaptive one-sided regularizer while memory and authority evolve over training.

Let $f = -J$, with $f^\star = \inf_\theta f(\theta) = -J^\star$, and consider $\theta_{t+1} = \theta_t - \eta_t g_{\mathrm{final},t}$. We assume a finite-horizon MDP with bounded rewards, a smooth population objective f, bounded stochastic actor gradients $\|g_{\mathrm{A},t}\| \le G_{\mathrm{A}}$, and controlled estimation error $\mathbb{E}\|g_{\mathrm{A},t} - \nabla f(\theta_t)\|^2 \le v_t$. The error v_t includes sampling variance and actor-gradient bias. The nonnegative memory losses $L_{\mathrm{P},t}$ must have uniformly Lipschitz gradients on a common domain containing all update segments. Sufficient policy and memory regularity conditions are in Appendix C.1.

Let $g_t = \nabla f(\theta_t)$, and let g_t^{PAIN} be the direction produced by Eq. (4) with g_t replacing $g_{\mathrm{A},t}$, keeping the current memory and q_t fixed. Its norm measures stationarity of the full PAIN-PPO update, rather than of the unregularized return objective. Write $S_T = \sum_{t<T} \eta_t$ and $\bar{v}_T = S_T^{-1} \sum_{t<T} \eta_t v_t$. Define the cumulative positive variation of the adaptive memory regularizer as

$$\mathcal{V}_T = \mathbb{E} \sum_{t=1}^{T-1} \Big[\beta_t L_{\mathrm{P},t}(\theta_t) - \beta_{t-1} L_{\mathrm{P},t-1}(\theta_t)\Big]_+.$$

The initial regularized gap is $\Delta_0 = \mathbb{E}[f(\theta_0) + \beta_0 L_{\mathrm{P},0}(\theta_0)] - f^\star < \infty$. With empty memory and fixed initialization, it reduces to the usual return suboptimality $\Delta_0 = J^\star - J(\theta_0)$.

Theorem 1 (Finite-time bound for the adaptive PAIN update)**.** *Under these conditions, suppose the frozen objectives $f + \beta_t L_{\mathrm{P},t}$ have a common smoothness constant L. For deterministic step sizes $0 < \eta_t \le L^{-1}$,*

$$\frac{1}{S_T} \sum_{t<T} \eta_t \mathbb{E}\big\|g_t^{\mathrm{PAIN}}\big\|^2 \le \frac{4(\Delta_0 + \mathcal{V}_T)}{S_T} + 2\big[1 + (1 + \rho_{\mathrm{P}})^2\big]\bar{v}_T. \tag{5}$$

Corollary 1 (Vanishing weighted update residual). *Under Theorem 1, if* $S_T \to \infty$, $\mathcal{V}_T = o(S_T)$, *and* $\bar{v}_T \to 0$, *then* $S_T^{-1} \sum_{t<T} \eta_t \mathbb{E}\|g_t^{\mathrm{PAIN}}\|^2 \to 0$.

The bound separates the initial optimization gap, adaptive-memory variation, and gradient-estimation error. The residual measures the full PAIN update, while Section 4.2 studies limiting policy quality.

4.2 A Shared-Parameter Separation

We give a concrete instance showing that retaining a verified correction can change the limiting policy, rather than merely modify individual updates. Consider one-step episodes with equally likely contexts s_1, s_2, binary actions $a \in \{0, 1\}$, and reward $R = a + E$, where $E \sim \mathrm{Unif}[-\sigma, \sigma]$, $\sigma > 0$, is independent of context and action. For $\theta = (x, y)$, let

$$\pi_\theta(1 \mid s_1) = \ell(4x), \qquad \pi_\theta(1 \mid s_2) = \ell(y - x),$$

where $\ell(z) = (1 + e^{-z})^{-1}$. Its expected return $J(\theta) = \frac{1}{2}[\pi_\theta(1 \mid s_1) + \pi_\theta(1 \mid s_2)]$ has supremum $J^\star = 1$. Increasing x improves the policy at s_1 but hurts it at s_2, so the shared parameters interfere.

Both methods start from $\theta_0 = (-2, 0)$. At each iteration, each collects a fresh batch of B episodes under its own policy and takes one Euclidean PPO-Clip step with the exact state-value baseline, unnormalized advantages, and $\eta_t = \eta_0/(t + 1)$. PAIN-PPO follows Algorithm 1 with empty committed and pending memories. After updates $t = 0, U, 2U, \ldots$, for fixed integer $U \geq 1$, it inspects all episodes with $s = s_1$ and $R \leq 0$. Exact same-realization replay compares both actions, including the factual control, with zero verifier tolerance. Two distinct episodes supporting $0 \to 1$ authorize the single constraint $4x \geq 0$. The first record is retained until authorization; both records are re-verified every U updates thereafter, refreshing their weights without increasing support. No additional constraints are admitted. Let τ be the first actor update with this correction. Appendix C.2 gives the weights, proof, and query-cost bounds.

Proposition 2 (Shared-parameter separation). *For the model and protocol above, let* $\rho_{\mathrm{P}} = 1$, $\alpha_{\mathrm{P}} > 0$, $G = 4(1 + \sigma)$, *and* $\delta \in (0, 1)$. *If* $0 < \eta_0 \leq \min\{(8G)^{-1}, \sqrt{B\delta}/(16G)\}$, *then* $\tau < \infty$ *almost surely and*

$$J(\theta_t^{\mathrm{PAIN}}) \longrightarrow 1 \quad \textit{a.s.}, \qquad \Pr\big(J(\theta_t^{\mathrm{PPO}}) \longrightarrow \tfrac{1}{2}\big) \geq 1 - \delta. \tag{6}$$

Here PAIN-PPO discovers the correction from empty memory and prevents the useful action at s_1 from losing all sampling probability. Re-verification maintains the correction until it becomes inactive and ordinary policy improvement continues. PPO can instead suppress the useful action at s_1 while continuing to improve at s_2. This is an existence separation for this instance, not a convergence guarantee for neural PPO, and discovery and maintenance cost extra counterfactual queries. An anonymized implementation is available online.[1]

5 Experiments

Each arena isolates part of the method. The Clean Corridor tests admission against ground truth, without memory or arbitration. The sepsis simulator also tests admission, with one-sided constraints, the relative cap, and the veto. The Guarded Maze tests persistence of a revised memory. Section 5.4 adds a noisy verifier and a distribution shift, and execution tests the whole system (Appendix H).

5.1 Clean Corridor: the mechanism

The corridor has twenty steps, four observable regimes, a speed v earning a small per-step reward, and a benign control p. One disaster occurs at an unobserved step per episode, with its class set by the regime. In L, severity grows with average speed so far, so early slowing helps consistently across tapes. In U, it depends on an independent draw and is action-independent. In H, an unobserved sign makes the corrective direction tape-specific. The classes share a severity distribution, making severity-based admission chance-level; class labels provide ground truth for scoring admission (Appendix F.1, Figure 5). Figure 2a varies the non-actionable failure share at a fixed total failure rate. Severity-based admission retains classes at their base rates and stays at chance. Same-tape replay rejects unavoidable

[1] https://anonymous.4open.science/r/PAIN-PPO-iclr2027/

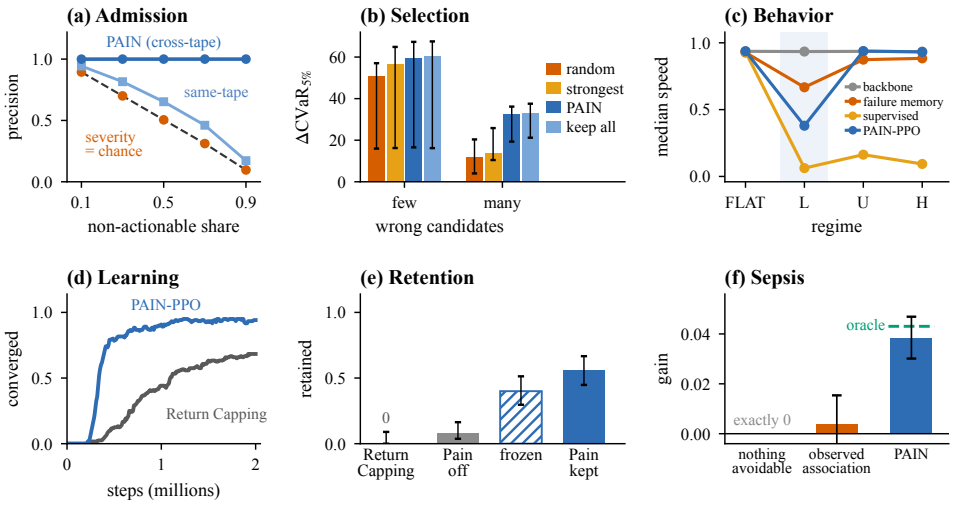


Figure 2: **Three simulator arenas.** (a–c) Corridor: admission precision, $\Delta\mathrm{CVaR}_{5\%}$ over the same-seed backbone, and median speed by regime. (d–e) Guarded Maze: success rate (120 seeds) and share of converged seeds that keep the route under plain PPO. (f) Sepsis: gain over the same backbone. (b) Median and interquartile range over 20 seeds; (e–f) 95% intervals (Wilson in e). Dashed: chance (a), oracle (f).

failures but admits hindsight ones because an alternative improves the recorded outcome. Cross-tape agreement removes these as well, reaching precision 1.000 at every mixture with recall just under one half. With per-batch entry counts matched, PAIN selects corrections that reproduce more often on new tapes than random or strongest-evidence selection in all four settings (Appendix F.1.3). When most candidates do not reproduce, its tail gain is $+32$ versus $+12$ and $+14$, respectively, and $+33$ for keeping every verified entry, the upper reference (Figure 2b). Entry choice rather than count accounts for about 90% of the gap over random selection. Sparse admission keeps the policy feasible under a binding budget (Appendix F.1.3). PAIN-PPO slows only in L, retaining backbone speed in FLAT, U, and H (Figure 2c), whereas failure memory and supervised controls also slow in U and H, adding cost without benefit. It improves tail performance over the same-seed backbone on every seed and meets the mean budget for all 40 seeds on validation tapes and 39 of 40 on evaluation tapes; the supervised control violates it on most seeds (Table 3, Figure 6).

5.2 Guarded Maze: an independent environment

Guarded Maze (Greenberg et al., 2022) is a risk-averse policy-gradient benchmark on an 8×8 maze (Figure 8). The goal pays $+16$ and each of the first 32 steps costs 1, and in the Return Capping configuration a guarded shortcut incurs a penalty of 64 in one episode in twenty. The long route is CVaR-optimal and the shortcut mean-optimal, so the maze tests whether corrections learned from rare guard penalties persist while most episodes favor the shortcut. We use the authors' implementation of Return Capping (Mead et al., 2025), which does not converge to the risk-averse policy on every seed of the discrete maze (Luo & Delage, 2026). PAIN-PPO adds the Pain channel through two hooks, with the one-sided constraint of Section 3.2 on the probability of the verified action and the veto on (Appendix H). Everything else is unchanged. Both arms train for two million environment steps, and PAIN-PPO's replays add about one million more. The pre-registered criterion is a mean $\mathrm{CVaR}_{5\%}$ of at least -10 over the last five training batches, and it holds exactly when the final policy avoids the guarded cell. After two million steps on 120 seeds (Figure 2d), PAIN-PPO converges on 37, 36, and 40 seeds against 24, 28, and 30 for Return Capping (Fisher $p = 0.0012, 0.048, 0.001$). The lead is larger earlier, and with replays counted it stands at 109 of 120 seeds against 76 after 1.5 million interactions (Appendix F.2). To test persistence, converged seeds continue from one million steps under plain PPO, whose mean objective favors the shortcut. With the Pain channel off, 6 of 75 converged PAIN-PPO seeds and none of the 39 converged Return Capping seeds keep the route after 100 updates. Keeping the channel keeps it on 42 of 75, at a cost in mean return (Figure 2e). Freezing the stored entries instead of revising them keeps it on 30 of 75 (Appendix F.2.1).

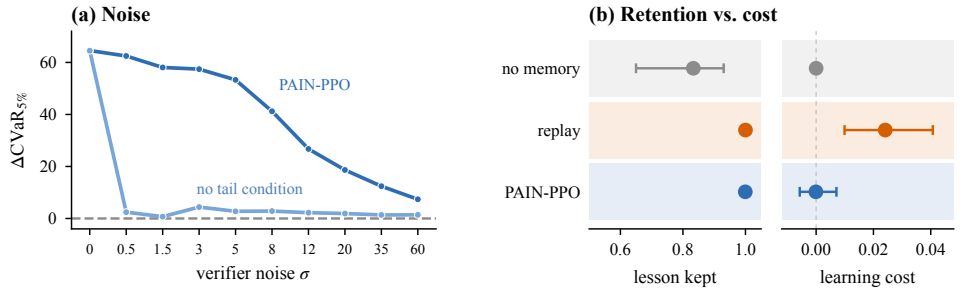


Figure 3: **Noisy verification and distribution shift.** (a) Corridor: median $\Delta\mathrm{CVaR}_{5\%}$ over the same-seed backbone under Gaussian verifier noise (20 seeds); $\sigma = 5$ matches a true correction's nominal gain. (b) Sepsis, with stored states still appearing in training: verified-action probability at these states (median and interquartile range, 24 seeds) and new-group return loss versus no memory (95% bootstrap intervals).

5.3 Sepsis simulator: PAIN rejects unavoidable failures

Following Oberst & Sontag (2019), we treat the transitions of their sepsis simulator as a Gumbel-Max structural causal model and draw its noise ourselves, so replaying an episode with one action changed gives the exact counterfactual under this model. We use the authors' code and learned parameters with terminal states made absorbing. A dial λ mixes each action's transition law toward the average over actions, and the share of deaths that some treatment could have prevented falls with λ and reaches zero at $\lambda = 1$. Every arm shares the same PPO backbone, warm start, and evidence budget and differs only in what enters memory. The memory holds nothing, corrections verified by counterfactual replay (PAIN), corrections read off observed associations at the same failure moments, or corrections from the value-iteration optimum, an oracle that fixes the ceiling (Appendix F.3).

Figure 2f shows how the gain changes with the share of avoidable deaths. When no death is avoidable, PAIN admits no correction on any evaluated seed and its gain over the backbone is exactly zero. When most deaths are avoidable, it recovers 88% of the gap between the backbone and the oracle. Corrections read off observed associations at the same failure moments gain about a tenth as much on the same seeds, with an interval that includes zero, so the improvement comes from counterfactual verification rather than from which failures are selected.

5.4 PAIN under noisy verification and distribution shift

For attribution under a noisy verifier, we add Gaussian noise to the corridor's counterfactual estimates and compare PAIN-PPO with and without tail-conditioned admission (Figure 3a). The two perform alike under exact verification. With noise the variant without the tail condition falls to near zero, while PAIN-PPO keeps most of its gain even when the noise equals the improvement from a true correction. For retention, every arm first learns the same verified corrections on low-oxygen admissions, to probability 0.97 for the verified action at the stored states, and then trains on normal-oxygen admissions only (Figure 3b). Without memory that probability drifts to 0.83, and to 0.68 once the stored states leave the training data. Replaying the same corrections to saturation keeps it at 1.00 but costs 0.024 of return on the new group and lowers the old one as well. PAIN-PPO keeps it above 0.99 at no detectable cost (Appendix F.3).

5.5 Execution on historical market data

One episode is one stock on one trading day. A parent order of 12% of the window's market volume is sold between 09:31 and 10:31, and every 30 seconds the policy sets b, the speed relative to a volume-proportional schedule, and p, the share posted passively. Queues are reconstructed from tick-level orders and trades, so each stock-day is a tape on which a different policy replays deterministically. The reference is POV (participation of volume), which sells a fixed 15% of the market volume as it arrives, and the policy is a residual on POV that reduces to it exactly when its output is zero. Δ is the implementation-shortfall difference against POV on the same stock-day, in bp with positive

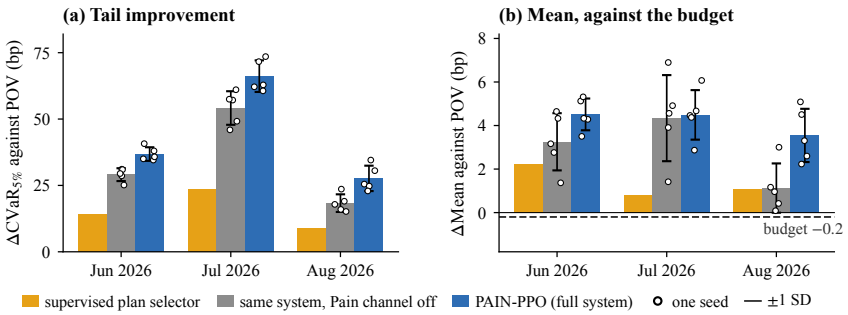


Figure 4: **A-share execution, three test months** (Section 5.5). (a) Tail improvement and (b) mean change against POV at 120 updates. Dashed, the -0.2 bp budget. Bars are means over 5 seeds, dots are seeds, and whiskers ± 1 SD. The supervised selector is a single model.

better. The target is $\Delta\mathrm{CVaR}_{5\%}$ and the pre-registered budget is $\Delta\mathrm{Mean} \geq -0.2$ bp. No fixed (b, p) schedule is feasible in this window. Both properties of Section 1 were measured on these data before the method was built (Appendix G). The data cover 67 trading days from January to August 2026, about 346,000 stock-days. Network weights and thresholds are fit on the 41 days up to May, and June, July, and August are the test months. The system has three layers, all decided by 09:31. An admission classifier trained on same-tape counterfactual labels chooses the stock-days the policy acts on. PAIN-PPO sets the size of (b, p) with $b \leq 0.2$, and a regime layer conditions on the first-minute market return and applies the budget per regime bucket. The b upper bound of 0.2 and the 120-update checkpoint were selected using only the January–May training months and fixed for all seeds and both arms before the June–August test-month evaluation (Appendix F.4).

Figure 4 shows the three test months, and all fifteen seed-months stay within the pre-registered mean budget on the point estimate, the criterion that 8 to 10 trading days per month can resolve. In every month the worst 5% of stock-days improves against POV, by 66, 28, and 37 bp in July, August, and June, a tenth to a fifth of POV's own worst-case loss. PAIN-PPO's tail gain exceeds the supervised selector's in every seed-month, and so does that of the same system retrained without the Pain channel, which makes the retrain a strong baseline for the channel. That retrain stays feasible but keeps only two-thirds to four-fifths of the tail gain, and the full system's tail exceeds it in all fifteen seed-months. Without admission, three of the five August seeds break the budget. An open-loop market-impact penalty, permanent or transient and up to 10 bp per 1% of window volume, leaves the order of the three systems unchanged in every seed-month (Appendix F.4). The gain works like insurance. The system gives up one or two basis points of mean on ordinary days and recovers far more in both mean and tail on predicted sharp-drop days. Appendix F.4 gives per-seed and trading-day results, development arms, and test-month reads.

6 Conclusion

When exogenous randomness dominates, a severe outcome does not say whether the agent's action was at fault. Policy-gradient and failure-memory methods then mistake bad luck for poor decisions, and later updates erase the rare lessons that were the agent's fault. In such environments the useful lessons are the few that survive verification, recur across futures, and are enforced only in context. PAIN builds this principle into policy optimization by connecting counterfactual attribution with persistent, revisable memory. A correction enters memory only when replay under the same realization shows that another action would have done better, and it gains influence only when its direction recurs across realizations. Each stored correction then acts as a local one-sided constraint, while rewards and advantage estimates stay unchanged. PAIN-PPO applies wherever two actions can be compared under the same realization of the world, as in replayable simulators, structural causal models, and historical replay. Extending admission to worlds that answer back, through calibrated response models or paired randomized deviations, is the next step. One target is the continual training of tool-using LLM agents, where alternative tool calls can be replayed under matched external conditions and the verified corrections retained during later training.

AI USE STATEMENT

We used generative AI tools to assist with implementation and debugging, data preparation, feedback on experimental design, hypothesis suggestions during debugging, result interpretation, figure preparation, manuscript drafting and editing, and literature search and reference formatting. The authors reviewed all AI-assisted work and take full responsibility for the final manuscript, code, and results.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Eitan Altman. *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999.

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

Nicole Bäuerle and Jonathan Ott. Markov decision processes with average-value-at-risk criteria. *Mathematical Methods of Operations Research*, 74(3):361–379, 2011.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

David A. Bruns-Smith. Model-free and model-based policy evaluation when causality is uncertain. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Lars Buesing, Théophane Weber, Yori Zwols, Nicolas Heess, Sébastien Racanière, Arthur Guez, and Jean-Baptiste Lespiau. Woulda, coulda, shoulda: Counterfactually-guided policy search. In *International Conference on Learning Representations (ICLR)*, 2019.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations (ICLR)*, 2019.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18 (167):1–51, 2018.

Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026): 768–774, 2024.

Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2004.

Ido Greenberg, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Efficient risk-averse reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Xiangkun He, Jingda Wu, Zhiyu Huang, Zhongxu Hu, Jun Wang, Alberto Sangiovanni-Vincentelli, and Chen Lv. Fear-neuro-inspired reinforcement learning for safe autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):267–279, 2024.

Arthur Juliani and Jordan T. Ash. A study of plasticity loss in on-policy deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Łukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. In *International Conference on Learning Representations (ICLR)*, 2017.

Chinmaya Kausik, Yangyi Lu, Kevin Tan, Maggie Makar, Yixin Wang, and Ambuj Tewari. Offline policy evaluation and optimization under confounding. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

Dohyeong Kim and Songhwai Oh. TRC: Trust region conditional value at risk for safe reinforcement learning. *IEEE Robotics and Automation Letters*, 7(2):2621–2628, 2022.

Dohyeong Kim, Kyungjae Lee, and Songhwai Oh. Trust region-based safe distributional reinforcement learning for multiple constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Dohyeong Kim, Taehyun Cho, Seungyub Han, Hojun Chung, Kyungjae Lee, and Songhwai Oh. Spectral-risk safe reinforcement learning with convergence guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

Zichuan Lin, Tianqi Zhao, Guangwen Yang, and Lintao Zhang. Episodic memory deep Q-networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Yudong Luo and Erick Delage. Boosting CVaR policy optimization with quantile gradients. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Yudong Luo, Guiliang Liu, Pascal Poupart, and Yangchen Pan. An alternative to variance: Gini deviation for risk-averse policy gradient. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Harry Mead, Clarissa Costen, Bruno Lacerda, and Nick Hawes. Return capping: Sample efficient CVaR policy gradient optimisation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Thomas Mesnard, Théophane Weber, Fabio Viola, Shantanu Thakoor, Alaa Saade, Anna Harutyunyan, Will Dabney, Tom Stepleton, Nicolas Heess, Arthur Guez, Éric Moulines, Marcus Hutter, Lars Buesing, and Rémi Munos. Counterfactual credit assignment in model-free reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Chenyang Miao. Learning from failures: Efficient reinforcement learning control with episodic memory. *arXiv preprint arXiv:2603.07110*, 2026.

Andrew Y. Ng and Michael I. Jordan. PEGASUS: A policy search method for large MDPs and POMDPs. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with Gumbel-Max structural causal models. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adrià Puigdomènech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000.

R. Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471, 2002.

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

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 (ICLR)*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Maximilian Seitzer, Bernhard Schölkopf, and Georg Martius. Causal influence detection for improving efficiency in reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ben Seymour. Pain: A precision signal for reinforcement learning and control. *Neuron*, 101(6): 1029–1041, 2019.

Aviv Tamar, Yonatan Glassner, and Shie Mannor. Optimizing the CVaR via sampling. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations (ICLR)*, 2019.

Sean Vaskov, Wilko Schwarting, and Chris Baker. Do no harm: A counterfactual approach to safe reinforcement learning. In *Proceedings of the 6th Annual Learning for Dynamics and Control Conference (L4DC)*, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block MDPs. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

A NOTATION

Backbone and objective (Section 2)		*Verified corrections and memory (Section 3.1)*	
$s_t,\ a_t,\ r_t$	state, action, and reward at step t	a_0	executed action at the inspected step
γ	discount factor	$a_1, \dots, a_K$	the K local probe actions
$\pi_\theta,\ \theta$	policy and its parameters	G_i	gain of probe i, $\Delta_t(a_i; a_0, \omega)$
$G_t,\ G_0$	return from step t, episode return	$h_i,\ \epsilon_{\text{noise}}$	excess gain and verifier tolerance, Eq. (2)
$J(\theta)$	expected return $\mathbb{E}_{\pi_\theta}[G_0]$	ϵ_{num}	numerical stabilizer
$u,\ \mathcal{B}_u,\ \theta_u$	update index, on-policy batch and actor at update u	$a^\star$	corrective target, Eq. (2)
$\widehat{A}_t$	advantage estimate	$\mathcal{M}_u,\ j$	correction memory after update u, one entry
$r_t(\theta),\ \epsilon_{\text{clip}}$	PPO importance ratio and clip radius	$n_j,\ n_{\min}$	distinct realizations supporting entry j, support floor
$L^{\text{clip}},\ L_{\text{PPO}},\ g_{\text{A}}$	clipped surrogate, PPO loss $-L^{\text{clip}}$, its gradient	$\bar{P}_j,\ P_j,\ n_{\text{scale}}$	mean and consolidated evidence strength, saturation scale
$\text{CVaR}_\alpha,\ \alpha$	lower-tail CVaR and its level	$w_j^{\text{auth}},\ w_j^{\text{persist}}$	authority and persistence weights of entry j
$J_{\text{ref}},\ \varepsilon$	reference-policy return, mean-budget slack	$\gamma_j,\ u_j$	retention factor, last re-verification update
$\widehat{A}_t^{\text{risk}},\ \kappa_{\text{tail}},\ \lambda$	Lagrangian advantage, tail weight, scaled dual coefficient	U	memory-refresh interval in updates
Counterfactual interface (Section 2.2)		*Constraints and update (Sections 3.2–3.3)*	
$F,\ \zeta_t$	transition map and exogenous randomness	$z_\theta(s)$	policy output in which corrections are measured
ω	recorded realization of an episode's exogenous randomness	$s_j,\ z_j^{\text{bad}},\ z_j^{\text{target}}$	stored state, executed output, target output
$J(a; \omega, t)$	return after replacing a_0 by a at step t under ω	$d_j,\ \hat{d}_j,\ \eta$	corrective displacement, its direction, retention fraction
$\Delta_t(a; a_0, \omega)$	same-realization counterfactual gain, Eq. (1)	$\psi_j(\theta),\ L_{\text{P}},\ g_{\text{P}}$	constraint value and loss, Eq. (3), and its gradient
		$\alpha_{\text{P}},\ \rho_{\text{P}}$	nominal correction strength, relative norm cap
		$g_{\text{P}}^{\text{corr}}$	bounded correction, Eq. (4)
		$q,\ F_{\text{P}},\ R_{\text{P}}$	veto strength $q = R_{\text{P}} F_{\text{P}}$, violation fraction, memory reliability
		$g'_{\text{A}},\ g_{\text{final}}$	PPO gradient after attenuation, final update, Eq. (4)
		$a^{\text{ref}},\ s_{\text{act}}$	reference action and action scaling of the bounded residual
Analysis (Section 4, Appendix C)			
t	update index in Section 4 and Appendix C	$f,\ f^\star,\ J^\star$	$f = -J$ and the optimal values
$\eta_t,\ \eta_0$	step size at update t and its initial value	$G_{\text{A}},\ v_t,\ \bar{v}_T$	bound on $\|g_{\text{A},t}\|$, gradient-error bound, its weighted average
$g_{\text{A},t},\ L_{\text{P},t},\ q_t$	PPO gradient, constraint loss, and veto strength at update t	$T,\ S_T$	number of updates, accumulated step size
β_t	adaptive regularization weight, Proposition 1	$\mathcal{V}_T,\ \Delta_0$	variation of the regularizer, initial regularized gap
$g_t,\ g_t^{\text{PAIN}}$	population gradient and the PAIN direction built from it	$L,\ H$	common smoothness constant, episode horizon
$\tau,\ B$	first actor update with the correction, batch size (Section 4.2)		

Upright subscripts A and P label the advantage (PPO) channel and the Pain channel. A few letters change meaning with their argument or scope. The reward is r_t and the PPO ratio is $r_t(\theta)$. The counterfactual gain $\Delta_t(\cdot)$ always carries its arguments, and Δ_0 is the initial gap of Section 4. The index t counts steps within an episode except in Section 4 and Appendix C, where it counts updates, and the step size η_t always carries its index, unlike the retention fraction η. In Appendix H, H in the formation formulas is the verified headroom rather than the horizon, and τ_u is the coefficient of g_{P} in the general update, unrelated to τ. The complete instance of Appendix C.2 has a single entry, so its strength P and retention factor γ drop the index j. A subscript e indexes the evidence records of an entry, whereas e raised to a power is Euler's number. In the sepsis simulator λ also names the mixing dial of Section 5.3. Symbols used only inside one arena (Appendix F), the audit (Appendix G), or a single proof are defined where they appear.

B Related Work

Risk-sensitive and constrained RL. PAIN-PPO builds on risk-sensitive and constrained reinforcement learning, where CVaR and related criteria are used to control tail risk through policy gradients, state augmentation, or constrained optimization (Rockafellar & Uryasev, 2000; Tamar et al., 2015; Chow et al., 2018; Bäuerle & Ott, 2011; Altman, 1999; Achiam et al., 2017; Tessler et al., 2019). Trust-region methods constrain CVaR directly, for example under a Gaussian approximation of the cost return (Kim & Oh, 2022). Distributional critics represent the return distribution directly, with quantile critics representing its quantiles, and provide a natural representation for tail-sensitive objectives (Bellemare et al., 2017; Dabney et al., 2018; Kim et al., 2023; 2024); see Section 2.1. This literature primarily asks how a policy should respond to risky outcome distributions, rather than whether a particular realized tail event was actually caused by the agent's action. A related distinction arises in memory-based RL. Prioritized replay emphasizes surprising transitions (Schaul et al., 2016), while episodic-control methods retain and reuse selected experiences (Pritzel et al., 2017; Lin et al., 2018), building on memory modules for rare events in supervised one-shot learning (Kaiser et al., 2017). FEMA (Miao, 2026), the closest method in spirit, stores short segments of failure episodes and retrieves similar experiences to discourage repeated failures. Such methods generally use observed outcomes to determine which experiences deserve emphasis or retention. PAIN-PPO instead inserts counterfactual verification and cross-realization consolidation between an observed failure and its admission into persistent memory. This distinction matters when severe outcomes are frequently driven by exogenous uncertainty rather than by avoidable actions, as explicitly tested in Section 5.1.

Causal and counterfactual RL. The verification step connects PAIN-PPO to causal and counterfactual reinforcement learning. Structural-causal approaches use logged trajectories to construct counterfactual scenarios for policy evaluation, policy search, or high-risk decision review (Buesing et al., 2019; Oberst & Sontag, 2019). Work on confounded MDPs studies when policy values remain identifiable, or can only be bounded, under latent confounding (Bruns-Smith, 2021; Kausik et al., 2024), while causal-invariance methods identify the state variables that causally determine future return (Zhang et al., 2020), and influence detection identifies when and what the agent can influence with its actions (Seitzer et al., 2021). Counterfactual safe RL similarly distinguishes harm induced by the learner from violations that would have occurred under a default policy (Vaskov et al., 2024). Related ideas also appear in hindsight relabeling and counterfactual credit assignment, which use alternative outcomes or actions to improve learning signals (Andrychowicz et al., 2017; Mesnard et al., 2021), as well as in paired-rollout and common-random-number constructions that reduce variation when alternative policies or actions are compared (Ng & Jordan, 2000; Glasserman, 2004, §7.1.1).

These approaches address whether, and in what sense, an action mattered for an observed outcome. PAIN-PPO focuses on the downstream question: *when should an episode-level counterfactual correction acquire persistent authority over future policy updates?* This distinction becomes important once counterfactual evidence is stored rather than used only for a local update. An erroneous causal estimate used once affects a limited learning step; once admitted into persistent memory, the same error may influence many subsequent updates. PAIN therefore couples within-realization counterfactual verification with recurrence across distinct realizations before a lesson is allowed to constrain future learning.

Memory protection and continual learning. The mechanism used to enforce an admitted lesson is related to gradient-based continual learning. GEM and A-GEM protect episodic memories by constraining new-task gradients (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019), while PCGrad removes components of gradients that conflict across objectives or tasks (Yu et al., 2020). PAIN-PPO uses a related projection principle but changes what is permitted to generate the constraining gradient. Its memory is formed from counterfactually verified failures within a single task, represents local corrective directions rather than simply replaying stored samples, and acquires authority through recurrence across distinct realizations. Thus, the central question differs from standard continual learning: rather than preserving performance on previous tasks, PAIN asks when a specific verified lesson should be protected from subsequent policy updates. This perspective is also distinct from the broader plasticity literature, which studies how neural networks lose their capacity to adapt during continued training (Lyle et al., 2023; Dohare et al., 2024; Juliani & Ash, 2024). Our concern is narrower: how much a previously verified corrective lesson may be overwritten while the underlying policy remains plastic. Fear-inspired safe RL anticipates danger before an action is taken (He et al., 2024); PAIN is retrospective, asking which realized failures were actually avoidable and which of those lessons should persist. We use the pain metaphor only in this computational sense (Seymour, 2019).

C Proofs

All norms are Euclidean, and $\ell(v) = (1 + e^{-v})^{-1}$.

C.1 Proofs for the Adaptive PAIN Update

Write $H_t = L_{\mathrm{P},t}$, $h_t = \nabla H_t(\theta_t)$, $\widehat{g}_t = g_{\mathrm{A},t}$, and $d_t = g_{\mathrm{final},t}$.

Proof of Proposition 1. For $h_t \neq 0$, Eq. (4) gives $d_t = \widehat{g}_t + \beta_t h_t$, where

$$\beta_t = \min\left\{\alpha_{\mathrm{P}}, \frac{\rho_{\mathrm{P}}\|\widehat{g}_t\|}{\|h_t\|}\right\} + q_t \frac{[-\langle \widehat{g}_t, h_t\rangle]_+}{\|h_t\|^2 + \epsilon_{\mathrm{num}}} \geq 0.$$

Set $\beta_t = 0$ when $h_t = 0$. Holding this coefficient fixed in differentiation proves the regularization identity. Moreover, $\langle h_t, -d_t\rangle = \langle h_t, -\widehat{g}_t\rangle - \beta_t\|h_t\|^2$. For the norm bound, set $u_t = h_t/\|h_t\|$ and $\bar{q}_t = q_t\|h_t\|^2/(\|h_t\|^2 + \epsilon_{\mathrm{num}}) \in [0, 1]$. The veto maps the component $a = \langle \widehat{g}_t, u_t\rangle < 0$ to $(1 - \bar{q}_t)a$ and leaves orthogonal components unchanged. It therefore cannot increase the norm. The added correction has norm at most $\rho_{\mathrm{P}}\|\widehat{g}_t\|$, so

$$\|d_t\| \leq (1 + \rho_{\mathrm{P}})\|\widehat{g}_t\|.$$

Finally, $\nabla[\psi]_+^2 = 2[\psi]_+\nabla\psi$ vanishes at every satisfied constraint, including its boundary. Thus $h_t = 0$ when all authorized constraints are satisfied. □

Remark 1. *The inner-product comparison controls the first-order change of the aggregate memory loss; individual constraint gradients may cancel.*

We first record sufficient conditions for the common smoothness constant used below. Assume $f \geq f^\star$, $\|\widehat{g}_t\| \leq G_{\mathrm{A}}$, $\mathbb{E}\|\widehat{g}_t - \nabla f(\theta_t)\|^2 \leq v_t$, and nonnegative memory losses with uniformly Lipschitz gradients on a common domain containing all update segments. For example, in a horizon-H MDP, bounded discounted return $|G_0| \leq R_H$, parameter-independent policy support, valid differentiation under expectation, and bounds $\|\nabla \log \pi\| \leq C_s$, $\|\nabla^2 \log \pi\|_{\mathrm{op}} \leq C_{ss}$ give $L_f = R_H(H^2C_s^2 + HC_{ss})$. For $H_t = \sum_j a_{t,j}[\psi_{t,j}]_+^2$ with $a_{t,j} \geq 0$, $\sum_j a_{t,j} \leq 1$, bounds $[\psi_{t,j}]_+ \leq C_0$, $\|\nabla\psi_{t,j}\| \leq C_1$, and $\mathrm{Lip}(\nabla\psi_{t,j}) \leq C_2$ give $L_H = 2(C_1^2 + C_0C_2)$. Since

$$0 \leq \beta_t \leq \bar{\beta} := \alpha_{\mathrm{P}} + \frac{G_{\mathrm{A}}}{2\sqrt{\epsilon_{\mathrm{num}}}},$$

one may take $L = L_f + \bar{\beta}L_H$ for the frozen objectives. Estimation bias is included in v_t.

Proof of Theorem 1. Set $\Phi_t = f + \beta_t H_t$, $r_t = \nabla\Phi_t(\theta_t)$, and $e_t = \widehat{g}_t - \nabla f(\theta_t)$, holding β_t fixed in differentiation. The update is $\theta_{t+1} = \theta_t - \eta_t(r_t + e_t)$. Smoothness and $\eta_t L \leq 1$ imply pathwise

$$\Phi_t(\theta_{t+1}) \leq \Phi_t(\theta_t) - \frac{\eta_t}{2}\|r_t\|^2 + \frac{\eta_t}{2}\|e_t\|^2.$$

For $t \geq 1$, let $\delta_t = [\beta_t H_t(\theta_t) - \beta_{t-1} H_{t-1}(\theta_t)]_+$. Then $\Phi_t(\theta_t) \leq \Phi_{t-1}(\theta_t) + \delta_t$. Telescoping, using $\Phi_{T-1}(\theta_T) \geq f^\star$, and taking expectations yields

$$\frac{1}{S_T} \sum_{t<T} \eta_t \mathbb{E}\|r_t\|^2 \leq \frac{2(\Delta_0 + \mathcal{V}_T)}{S_T} + \bar{v}_T.$$

Let $\mathcal{A}_t(v)$ be the update map with current memory and q_t fixed, and $C_t(v) = \mathcal{A}_t(v) - v$. For $h_t \neq 0$,

$$C_t(v) = (\bar{q}_t[-\langle v, u_t\rangle]_+ + \min\{\alpha_{\mathrm{P}}\|h_t\|, \rho_{\mathrm{P}}\|v\|\})\, u_t.$$

Thus C_t is $(1+\rho_{\mathrm{P}})$-Lipschitz, also when $h_t = 0$. Since $r_t = \nabla f(\theta_t) + C_t(\widehat{g}_t)$ and $g_t^{\mathrm{PAIN}} = \nabla f(\theta_t) + C_t(\nabla f(\theta_t))$,

$$\|g_t^{\mathrm{PAIN}}\|^2 \leq 2\|r_t\|^2 + 2(1+\rho_{\mathrm{P}})^2 \|e_t\|^2.$$

Substitution proves Eq. (5); taking limits proves Corollary 1. □

Remark 2. *The bound controls the PAIN update residual, not $\|\nabla J\|$, and does not imply global optimality. Vanishing residual requires $\mathcal{V}_T = o(S_T)$ and $\bar{v}_T \to 0$; periodic refresh alone does not ensure the variation condition.*

C.2 Proof of the Shared-Parameter Separation

We first instantiate the protocol used in Proposition 2, specifying its weights and stochastic update. In the remainder, gradients are ascent directions. Write $z = y - x$, $p_1 = \ell(4x)$, $p_2 = \ell(z)$, and $f_1 = (4,0)^\top$, $f_2 = (-1,1)^\top$. An episode has gradient

$$H(\theta; s_i, a, E) = f_i(a - p_i)(a - p_i + E), \qquad \widehat{g}_t = B^{-1} \sum_{k=1}^{B} H(\theta_t; s_{t,k}, a_{t,k}, E_{t,k}).$$

This is the PPO-Clip gradient at the behavior policy with the exact baseline $V^\pi(s_i) = p_i$. Define

$$a(x) = 2\ell'(4x), \quad b(z) = \tfrac{1}{2}\ell'(z), \quad g(x,y) = (a(x) - b(z), b(z)), \quad e_t = \widehat{g}_t - g(\theta_t).$$

Conditioning on the pre-batch history $\mathcal{F}_t$ gives $\mathbb{E}[e_t \mid \mathcal{F}_t] = 0$, $\mathbb{E}[\|e_t\|^2 \mid \mathcal{F}_t] \leq G^2/B$, and $\|\widehat{g}_t\| \leq G = 4(1+\sigma)$, since $\mathbb{E}[(a - p_i)(a - p_i + E)] = p_i(1 - p_i)$ and $\|f_i\| \leq 4$.

A supporting record is exactly $(s_1, a = 0, E \leq 0)$. Paired replay gives gains $\{0, 1\}$ and excess gains $\{0, 1/2\}$; action 1 instead gives zero excess gains. With zero direction assigned to the factual control, each supporting record has strength

$$P_e = (1 - e^{-1/(2s_P)}) \frac{1/2}{1/2 + \epsilon_{\mathrm{num}}}, \qquad P = P_e(1 - e^{-2/n_{\mathrm{scale}}}) > 0,$$

where $s_P, n_{\mathrm{scale}} > 0$. Exact state matching and two distinct records authorize

$$\psi(x,y) = -\log \frac{\pi(1 \mid s_1)}{\pi(0 \mid s_1)} = -4x \leq 0. \tag{7}$$

If authorization follows batch jU, then $\tau = jU + 1$. Re-verification preserves both records, strength P, and support count two. For fixed $0 < \gamma_{\min} \leq \gamma_{\max} < 1$, set $\gamma = \min\{\gamma_{\max}, \gamma_{\min} + (1 - \gamma_{\min})P\}$. The age $d_t \in [0, U]$ at each actor update gives

$$w_t^{\mathrm{auth}} = w_t^{\mathrm{persist}} = w_t = P\gamma^{d_t}, \qquad L_{\mathrm{P},t}(x) = \frac{w_t}{w_t + \epsilon_{\mathrm{num}}}[-4x]_+^2. \tag{8}$$

Weights are frozen in differentiation. With $\rho_{\mathrm{P}} = 1$, arbitrary memory-dependent $q_t \in [0,1]$, $\kappa_t = 32\alpha_{\mathrm{P}} w_t/(w_t + \epsilon_{\mathrm{num}})$, and $\bar{q}_t = q_t\|g_{\mathrm{P},t}\|^2/(\|g_{\mathrm{P},t}\|^2 + \epsilon_{\mathrm{num}})$, PAIN uses

$$\theta_{t+1} = \theta_t + \eta_t(\widehat{g}_t + \Gamma_t e_x), \qquad e_x = (1,0)^\top,$$
$$\Gamma_t = \begin{cases} \bar{q}_t[-\widehat{g}_{x,t}]_+ + \min\{\kappa_t[-x_t], \|\widehat{g}_t\|\}, & t \geq \tau,\ x_t < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

PPO uses $\Gamma_t = 0$ throughout. In particular, $\Gamma_t \geq 0$ and $\|\theta_{t+1} - \theta_t\| \leq 2G\eta_t \to 0$ for PAIN. The correction may depend on the current batch; all arguments involving it below are pathwise.

The proof uses the following four lemmas.

Lemma 1 (Accumulated sampling noise). *For either algorithm, $M_n = \sum_{t<n} \eta_t e_t$ converges almost surely and in L^2, and*

$$\mathbb{E}\|M_\infty\|^2 < \frac{2G^2\eta_0^2}{B}, \qquad \Pr\left(\sup_n \|M_n\| > r\right) \le \frac{2G^2\eta_0^2}{Br^2} \quad (r > 0). \tag{10}$$

Proof. Martingale increments are orthogonal in L^2, so $\sup_n \mathbb{E}\|M_n\|^2 \le (G^2/B)\sum_t \eta_t^2 < 2G^2\eta_0^2/B$. Martingale convergence and the maximal inequality for $\|M_n\|^2$ give the claims. □

Lemma 2 (Correction formation). *Under the protocol in Appendix C.2 and $\eta_0 \le 1/(8G)$, $\tau < \infty$ almost surely.*

Proof. Couple the process before formation to PPO with the same samples. Since $|\widehat{g}_{x,t}| \le G$,

$$x_t \le -2 + \eta_0 G(1 + \log(t+1)), \qquad r_t := \Pr(\text{support} \mid \mathcal{F}_t) = \frac{1}{4(1+e^{4x_t})} \ge r_*(t+1)^{-\zeta},$$

where $\zeta = 4\eta_0 G \le 1/2$ and $r_* = [4(1+e^{-8+\zeta})]^{-1}$. Let K_m count supports in the first m inspection batches of this auxiliary PPO process, and $S_m = \sum_{j<m}(jU+1)^{-\zeta}$. Conditional binomial sampling and iterated expectation give

$$\Pr(K_m \le 1) \le e\mathbb{E}[e^{-K_m}] \le \exp\{1 - (1-e^{-1})Br_*S_m\} \longrightarrow 0.$$

Retaining the first record makes nonformation through these inspections exactly $K_m \le 1$. Since $S_m \to \infty$, formation is almost sure. □

Lemma 3 (Post-formation protection). *For all $t \ge \tau$, $\kappa_t \ge \kappa_* > 0$ and*

$$\kappa_* = 32\alpha_{\mathrm{P}} \frac{P\gamma_{\min}^U}{P\gamma_{\min}^U + \epsilon_{\mathrm{num}}}, \tag{11}$$

$$x_t \ge L_\tau := \min\{x_\tau, -G/\kappa_* - \eta_0 G\}, \qquad \pi_t(1 \mid s_1) \ge \ell(4L_\tau) > 0. \tag{12}$$

Proof. Re-verification gives $w_t \ge P\gamma_{\min}^U$. If $x_t \le -G/\kappa_*$, the correction saturates at its norm cap, hence $\widehat{g}_{x,t} + \Gamma_t \ge \widehat{g}_{x,t} + \|\widehat{g}_t\| \ge 0$. Otherwise a negative step decreases x by at most $\eta_0 G$. Induction gives the bound. □

Lemma 4 (Pathwise drift). *For $r_{t+1} = r_t + \eta_t h_t + N_{t+1} - N_t$, assume $\eta_t > 0$, $\sum_t \eta_t = \infty$, N_t converges, and $|r_{t+1} - r_t| \to 0$. If eventually $r_t \le 0$ implies $h_t \ge 0$, then r_t has a finite lower bound. Given a lower bound L, if the drift is eventually uniformly positive on every $[L, A+1]$, then $r_t \to \infty$.*

Proof. Choose a tail on which increments and noise oscillations are below ε. Each excursion below zero starts above $-\varepsilon$ (or at the tail's initial value) and loses at most another ε. For the second claim, remaining below $A+1$ contradicts the divergent sum of positive drift. A subsequent downcrossing from $A+1$ to A is impossible when $2\varepsilon < 1$, since its drift is nonnegative and its initial overshoot and noise loss total at most 2ε.

□

Proof of Proposition 2. PAIN branch. Work on the probability-one event of formation and noise convergence. With $a_t = a(x_t)$ and $b_t = b(z_t)$,

$$\begin{aligned} x_{t+1} &= x_t + \eta_t(a_t - b_t + \Gamma_t) + \Delta M_t^x, \\ y_{t+1} &= y_t + \eta_t b_t + \Delta M_t^y, \end{aligned} \qquad x_t \ge L_\tau, \quad \Gamma_t \ge 0. \tag{13}$$

Here $\Delta M_t = M_{t+1} - M_t$. The y recursion implies either $y_t \to \infty$ or y_t has a finite limit. In the latter case $\sum \eta_t b_t < \infty$, and the x recursion implies that x_t has a finite limit or tends to infinity. Finite limits of both coordinates give $b_t \to b_\infty > 0$, a contradiction. If $x_t \to \infty$, then eventually $\Gamma_t = 0$, $z_t \to -\infty$, and

$$\frac{a_t}{b_t} \le 16e^{-4x_t - z_t} = 16e^{-3x_t - y_t} \to 0.$$

Consequently $\sum \eta_t a_t < \infty$, again forcing a finite limit for x_t. Thus $y_t \to \infty$.

For each finite $A \geq L_\tau$, set $a_A = \min_{x \in [L_\tau, A+1]} a(x) > 0$. Since $b(y_t - x) \to 0$ uniformly on this interval, eventually

$$x_t \in [L_\tau, A+1] \quad \Longrightarrow \quad a_t - b_t + \Gamma_t \geq a_A/2.$$

Lemma 4 yields $x_t \to \infty$, so the correction is eventually inactive. Thereafter

$$z_{t+1} = z_t + \eta_t(2b_t - a_t) + \Delta N_t, \qquad N_t = M_t^y - M_t^x.$$

For $z_t \leq 0$ and eventually $x_t > 0$, $a_t/b_t \leq 16e^{-4x_t - z_t} = 16e^{-4y_t + 3z_t} \leq 16e^{-4y_t} \to 0$. Lemma 4 first gives a finite lower bound L_z. On any $[L_z, A+1]$, b has a positive minimum while $a_t \to 0$, so the same lemma gives $z_t \to \infty$. Hence $p_{1,t}, p_{2,t} \to 1$ and $J(\theta_t) \to 1$.

PPO branch. Set $W = 3x + y = 4x + z$ and

$$\mathcal{R} = \{x \leq -1,\ z \geq 1,\ W \leq -\log 32\}.$$

On $\mathcal{R}$, $a/b \leq 16e^{4x+z} = 16e^W \leq 1/2$, giving

$$g_x \leq -\tfrac{1}{2}b, \qquad g_z \geq \tfrac{3}{2}b, \qquad g_W \leq -\tfrac{1}{2}b.$$

Let $\mathcal{E} = \{\sup_n \|M_n\| \leq 1/8\}$. Lemma 1 gives $\Pr(\mathcal{E}^c) \leq 128G^2\eta_0^2/B \leq \delta/2$. On $\mathcal{E}$, a first exit n from $\mathcal{R}$ would satisfy, by summing the preceding drifts from $(x_0, z_0, W_0) = (-2, 2, -6)$,

$$x_n \leq -2 + \tfrac{1}{8} < -1, \quad z_n \geq 2 - \tfrac{1}{4} > 1, \quad W_n \leq -6 + \tfrac{1}{2} < -\log 32,$$

contradicting exit. On this event intersected with almost-sure noise convergence, let $V_n = \sum_{t<n} \eta_t b_t$. If $V_\infty < \infty$, then $a_t \leq b_t/2$ implies finite limits for both coordinates, contradicting $b(z_\infty) > 0$ and $\sum \eta_t = \infty$. Therefore $V_n \to \infty$ and

$$x_n \leq -2 - \tfrac{1}{2}V_n + M_n^x \to -\infty, \qquad z_n \geq 2 + \tfrac{3}{2}V_n + M_n^y - M_n^x \to \infty.$$

Thus $J(\theta_n) \to 1/2$ with probability at least $1 - \delta/2 \geq 1 - \delta$. □

Further consequences. The same construction also gives explicit bounds on correction formation, revisitation, and counterfactual query cost.

At initialization, a supporting episode has probability $r_0 = \frac{1}{4}(1 - \ell(-8))$. Thus

$$\Pr(\tau = 1) = 1 - (1 - r_0)^B - Br_0(1 - r_0)^{B-1}. \tag{14}$$

Since $r_0 > 1/8$, $\Pr(\tau > 1) \leq e^{1-(1-e^{-1})Br_0} \leq e^{1-B/16}$. Hence $B \geq \lceil 16 \log(e/\delta_f) \rceil$ suffices for $\Pr(\tau = 1) \geq 1 - \delta_f$. The proof of Lemma 2 also gives, with its r_*, ζ, S_m,

$$\Pr(\tau > (m-1)U + 1) \leq \min\{1, \exp[1 - (1 - e^{-1})Br_* S_m]\}. \tag{15}$$

We next bound the return to a verified action under a changing policy. Let $q_\tau = \ell(4L_\tau) > 0$. For $t \geq \tau$, let T_t count batches starting at t until the first batch containing $(s_1, a = 1)$, including that successful batch. Conditional on $\mathcal{F}_t$, the bound q_τ is known and remains valid along every future path. Each future episode has conditional probability at least $q_\tau/2$ of that state-action pair. Conditional independence within a batch and repeated conditional expectation across batches yield

$$\Pr(T_t > k \mid \mathcal{F}_t) \leq (1 - q_\tau/2)^{Bk}, \qquad \mathbb{E}[T_t \mid \mathcal{F}_t] \leq \frac{1}{1 - (1 - q_\tau/2)^B} \leq 1 + \frac{2}{Bq_\tau}. \tag{16}$$

The expectation follows by summing the tail; the last inequality uses $(1 - v)^{-B} \geq 1 + Bv$ for $v = q_\tau/2$. This is a geometric tail bound, not an assertion that waiting times under a changing policy are geometrically distributed. The return of a direct sample also does not guarantee that its noisy gradient has the correct sign.

On $\{\tau = 1\}$, the first actor update is PPO and $x_1 \geq -2 - \eta_0 G$. Thus one may replace L_τ by the deterministic $L_* := \min\{-2 - \eta_0 G, -G/\kappa_* - \eta_0 G\}$ in the probability bound. Equation (14) gives the probability of this event.

For discovery and maintenance, the first-batch result uses B fresh episodes and at most $2B$ replay calls, counting both factual and alternative returns for every inspected candidate. Thus, taking $B \geq \lceil 16 \log(e/\delta_f) \rceil$ gives $O(\log(1/\delta_f))$ on-policy and counterfactual formation budgets. It does not bound the total samples needed for near-optimal return.

For fixed small B, Eq. (15) applies across inspections. The sum satisfies $S_m \geq U^{-\zeta}((m+1)^{1-\zeta} - 1)/(1-\zeta)$. Any m for which $(1-e^{-1})Br_*S_m \geq 1 + \log(1/\delta_f)$ guarantees formation by update $(m-1)U+1$ with probability at least $1-\delta_f$. This uses at most $B((m-1)U+1)$ fresh episodes and $2Bm$ discovery replay calls.

After $T \geq 1$ actor updates, $N_{\text{on}}(T) = BT$. Let $n_{\text{disc}}(T)$ count pre-formation and formation inspections, and $n_{\text{maint}}(T)$ count later maintenance inspections. Then

$$N_{\text{cf}}(T) \leq 2B\, n_{\text{disc}}(T) + 4\, n_{\text{maint}}(T), \qquad n_{\text{disc}}(T) + n_{\text{maint}}(T) \leq 1 + \left\lfloor \frac{T-1}{U} \right\rfloor. \tag{17}$$

The maintenance cost is four calls because each of two records has two returns re-evaluated. On $\{\tau = 1\}$, the bound simplifies to $N_{\text{cf}}(T) \leq 2B + 4\lfloor (T-1)/U \rfloor$. Reusing a tape refreshes its evidence but creates no new independent support. A larger U reduces query frequency while weakening the lower bound in Eq. (11).

Numerical check. For the shared-parameter construction, the numerical check (Table 1) evaluates the post-verification core dynamics with the population-gradient version of the construction (identical initialization $(-2, 0)$, step size 0.1, 10,000 rounds, teaching cap retained; this does not simulate the new discovery protocol; script and full trajectories released as supplementary material).

Method	$\pi(1 \mid s_1)$	$\pi(1 \mid s_2)$	J
PPO	$1.86 \cdot 10^{-8}$	0.998998	0.499499
PAIN-PPO	0.999220	0.997394	0.998307

Table 1: Numerical check of the shared-parameter construction (population gradients, constant step size; sanity check of the dynamics, not an estimate of the stochastic theorem's failure probability).

D Environment, Invariants, and the Three-Layer Principle

The execution engine replays recorded order and trade events, reconstructing per-price queues under price-time priority; the agent's shadow orders join queues behind recorded volume and are filled only by recorded executable flow after the ahead-queue depletes. The engine is frozen (versioned, golden-tested) before all audits; accounting invariants (inventory conservation, cash consistency, fill-source attribution) hold to machine precision on every episode. Observation and execution datasets are precomputed for the full market (losslessly compressed, verified column-equal against the reference builder). The reconstruction defect noted in Appendix G (marketable orders recorded as resting, creating ghost book levels) was fixed by a double-sided marketable invariant in the rebuild; earlier results that it affects are marked as pre-fix where they appear (Appendix G).

Three-layer separation: the environment exposes mechanics and guarantees feasibility; the Router, which is the POV schedule in the runs of Section 5.5, maps state to baseline actions deterministically; the agent contributes bounded residuals. Three earlier departures from this separation were removed, a redirect of rejected limit orders to market orders, environment-side cancellation of resting orders after a fixed lifetime, and a 50% passive target for action axes that a run does not control in place of the Router's explicit reference, which coupled the stale-rescue lever x to passive reconciliation. All three behaviors are now explicit, controllable, and logged.

E Statistical Protocol

Uncertainty is reported at the level at which observations are independent. In the simulator arenas the unit is the training seed (40 confirmatory seeds on the corridor, three blocks of 40 on the maze, 96 seed clusters in the sepsis simulator), and intervals are bootstrap intervals over seeds. On the execution data the unit is the trading day: stock-days within a day share the market's move, so every interval is a trading-day-clustered bootstrap with 1,500 resamples over paired same-tape differences, with the five seeds averaged inside each resample; each month has 8 to 10 such clusters. Feasibility is judged per seed-month on the point estimate of the monthly mean against the pre-registered budget of -0.2 bp. The point estimate is used because these data cannot resolve the budget at the interval level:

with 8 to 10 trading days per month, the day-clustered interval of a monthly mean is 13 to 21 bp wide, against a budget of 0.2 bp.

Every comparison is paired on the same seeds, tapes, or trading days, with the same admission and feasibility machinery for every arm, so that the market's own move cancels in the difference. Tails are reported as paired differences against the reference policy on the same stock-days.

Thresholds and configurations are fit on training data. The confirmatory seeds of the corridor and of the maze training runs were not used to fit any training parameter, no network weight or threshold of the execution system was fit on its three test months (Appendix F.4). This training-only selection also applies to the b cap and the number of policy updates. Deployable frozen policies are always distinguished from ex-post regime-optimal comparators, which appear only as diagnostics. Validity screening separates data or simulator degeneracies (excluded and counted) from economically hard episodes (kept and reported).

F Experimental Details

Numbers for the corridor, the Guarded Maze, and the sepsis simulator (Appendices F.1–F.3) come from result files in the repository released anonymously at `https://anonymous.4open.science/r/PAIN-PPO-iclr2027/`, and each of these arenas ships scripts that regenerate its main tables. The exogenous-share sweep of the corridor was run after the other results were frozen. The execution results are computed from per-stock-day gain files derived from exchange market data. The release includes these files with anonymized stock identifiers and trading-day indices, the script that recomputes every execution number from them, and the code of the reported system, but not the market data.

PAIN has two parts that can be tested separately. Admission decides what is remembered, and retention decides how the stored corrections act on later updates. Table 2 gathers the controls that change one part at a time, to show which component carries the gain. Replacing counterfactual admission by severity or by observed association removes most of the gain wherever it was tested. The tail condition adds nothing with an exact verifier, but it matters once the verifier is noisy and in the Guarded Maze. Keeping the pipeline and the states while changing what is stored, with shuffled keys, rehearsed earlier actions, or an anchor whose targets are the earlier actions, also removes the gain, so the content of the stored corrections matters. Losses that use the same verified corrections are compared in Appendix F.3. In the corridor, shuffling the keys also lets hindsight events through the direction gate (Appendix F.1). The channel acts through teaching and through the veto, and each can work alone. In the sepsis simulator, when the training data shift before the corrections are learned, teaching alone and the veto alone each carry the gain across the shift, and removing the same gradient mass in random directions does not. Once the corrections are learned, holding them needs teaching only, and adding the veto brings no gain (Appendix F.3). Cross-realization consolidation chooses which entries to keep. Among selections of the same size it picks entries that reproduce, and it raises the tail where most candidates do not reproduce ($+32$ against $+12$ and $+14$, Appendix F.1.3). Against keeping every verified entry it does not raise the tail under a loose budget, and with a noisy verifier the variant without its two gates keeps more of the tail gain. Under a binding budget it decides feasibility, with 16 of 20 seeds within budget with it and none when both gates are off and reliability is fixed at 1, and what protects the budget there is the smaller number of entries, since a random selection of the same number is feasible at least as often (Appendix F.1.3).

F.1 Clean Corridor: scoring the admission rule against ground truth

The corridor has $T = 20$ steps. The action is $a = (v, p)$: v is a speed, which is pure gain in ordinary steps and therefore tempting, and p is a benign control whose optimum is 0.5 and which only costs when moved. Each tape draws one of four observable regimes and a shock time t_{shock}, uniform on $\{7, \dots, 13\}$ and not observed by the agent; the disaster settles once, at that step, on the mean speed w up to that step, with a magnitude $M = K \cdot \text{relu}(|\zeta| - m_0)$, $\zeta \sim \mathcal{N}(0, 1)$, drawn independently of the regime:

Table 2: **Component controls and joint ablations**, gathered from the four arenas. Corridor: median $\Delta\mathrm{CVaR}_{5\%}$ against the same-seed backbone and seeds within budget. Guarded Maze: seeds meeting the pre-registered criterion. Sepsis: mean return gain, with 95% bootstrap intervals, and seeds favoring PAIN. Execution: $\Delta\mathrm{CVaR}_{5\%}$ against POV in bp (July / August / June). The Guarded Maze column for PAIN-PPO is the development configuration with teaching and the veto both on (seeds 0–19) under the repaired verifier, not the confirmatory runs.

Control	Arena (seeds)	Control result	PAIN-PPO
What is admitted			
Severity only, no counterfactual check	Corridor (200–239)	$+12.8$, 30/40	$+66.6$, 40/40
Failure memory, severity admission	Corridor (200–239)	$+14.0$, 31/40	$+66.6$, 40/40
Observed association at the same failure moments	Sepsis (13000+, 14000+)	$+0.0018$, $+0.0057$ over the backbone	$+0.0307$, $+0.0456$
Nothing avoidable ($\lambda = 1$)	Sepsis	no lesson admitted, gain 0 exactly	—
Without the tail condition	Corridor, noise $\sigma = 0$ / 5 (300–319)	$+64.5$ / $+2.7$	$+64.6$ / $+53.3$
Without the tail condition	Maze (0–19)	1/20	19/20
What is stored (same pipeline and states)			
Shuffled retrieval keys	Corridor (200–239)	$+1.8$, 37/40	$+66.6$, 40/40
Rehearsal of earlier actions, same storage	Sepsis shift (39000+)	PAIN ahead by $+0.0268$ $[+0.0119, +0.0430]$, 10/12	—
Rehearsal of earlier actions, all entries	Sepsis shift (39000+)	PAIN ahead by $+0.0384$ $[+0.0200, +0.0589]$, 11/12	—
Anchor with earlier actions as targets	Sepsis shift (39000+)	PAIN ahead by $+0.0403$ $[+0.0209, +0.0612]$, 9/12	—
How it is applied			
Supervised regression on the same verified evidence	Corridor (200–239)	$+42.4$, 13/40	$+66.6$, 40/40
Supervised selector on the same counterfactual menu	Execution	23.6 / 8.9 / 14.1	66.2 / 27.7 / 36.8
Veto against the same gradient removed in random directions	Sepsis shift (38000+)	veto ahead by $+0.0369$ $[+0.0198, +0.0544]$, 11/12	—
Support-count gate off ($n_{\min} = 1$)	Maze (0–19)	16/20	19/20
Cross-tape gates off (support count and direction), loose budget	Corridor, noise $\sigma = 5$ (300–319)	$+62.2$	$+53.3$
Both cross-tape gates off and reliability fixed at 1, binding budget	Corridor (400–419)	$+41.72$, 0/20	$+3.25$, 16/20
Whole channel			
Same snapshot selection, no teaching	Corridor (200–239)	$+0.4$, 36/40	$+66.6$, 40/40
Pain channel off, same recipe retrained	Execution	54.2 / 18.3 / 29.1	66.2 / 27.7 / 36.8

Regime	Disaster	Class
NORMAL	none	—
L	$-M\,w$, w the average speed so far	avoidable
U	$-M\,\tilde{w}$, $\tilde{w}$ an independent draw with the law of w	zero leverage
H	$-M\,w$ or $-M\,(1-w)$, chosen by an unobserved $h = \pm 1$	hindsight

Three properties hold by construction and are checked numerically on 200,000 tapes by a script that ships with the arena: the expected severity is equal across L, U, and H, so the posterior probability of L given the severity quintile equals the prior in every quintile (0.6613 to 0.6735 against 0.6669); the U disaster does not depend on the action at all (the largest difference between the shock at $w = 0$ and at $w = 1$ is exactly zero); and the corrective direction is the same on every L tape (0 of 3,000 flips) while on H it flips with h (2,358 positive against 2,388 negative). Figure 5 draws one episode and the disaster law of each regime.

With a fixed reference policy ($v = p = 0.5$) and 20,000 tapes, each admission rule is scored by the fraction of admitted evidence that comes from L tapes.

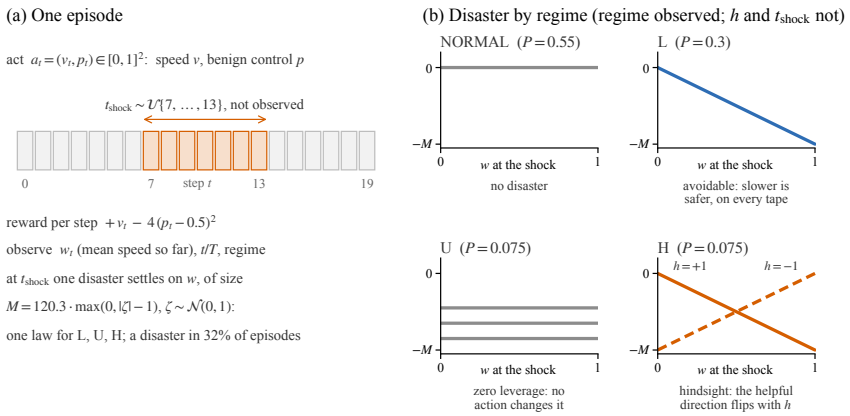


Figure 5: **The Clean Corridor.** (a) One episode of 20 steps. The disaster settles once, at a hidden step in $\{7, \ldots, 13\}$, on the mean speed w up to that step, and its size M has the same law in every regime. (b) The disaster as a function of w in each regime, in units of M, with the regime probabilities of the confirmatory runs. Three independent draws $\tilde{w}$ are shown for U.

Admission rule	Events	Precision	Base rate	Admitted from
A: severity only	39,992	0.664	0.663	L 26,539, U 6,600, H 6,853
B: A + same-tape counterfactual verification	15,993	0.788	0.663	L 12,600, H 3,393
C: B + cross-tape direction agreement (Pain)	12,599	**1.000**	0.663	L 12,599
C without the tail condition	13,064	1.000	0.663	L 13,060, H 4
C with shuffled retrieval keys	15,504	0.790	0.663	L 12,242, H 3,262

Rule B removes every U tape and rule C removes every H tape while losing one L tape; which gate removes which class is decided by the three constructive properties above, not by tuning. Harvested at the training batch size (256 tapes per update, about 25 in the tail, 60 updates, same reference policy), rule C admits 9,243 L and 338 H events (precision 0.965) and rule B 9,491 L and 2,467 H (0.794): with only a few tapes per key, three H tapes with the same sign occasionally pass the direction gate, so the 1.000 above is a large-sample property of the rule and not a guarantee per update (`admission/gt_admission_batch256.py` in the released repository).

F.1.1 End-to-end results

The end-to-end runs use forty fresh seeds (200 to 239), 60 updates, 6,000 evaluation tapes, and a mean budget of -0.2 shared by every arm through the same controller and the same best-feasible snapshot selection. For every seed we first train the backbone PPO with the Pain channel disabled under the identical protocol and evaluate its $\text{CVaR}_{5\%}$ on the same 6,000 tapes. This per-seed value is the reference against which every Δ in Table 3 is computed within the seed. Its median over the 40 seeds is -83.23 and it ranges from -101.4 to -19.9; it is a property of the trained backbone on that seed, not a bound of the environment and not a constant shared across seeds, which is why every comparison is paired. The backbone's speed on L is bimodal across seeds (21 seeds near 0.94, 16 between 0.3 and 0.8, 3 near 0.06), because the mean return on L is flat in the speed by calibration and PPO has no gradient there; the L medians of the two backbone rows therefore sit on opposite sides of the gap although 34 of 40 seeds agree within 0.05.

Paired sign tests of PAIN-PPO against each arm on $\Delta\text{CVaR}_{5\%}$ all give $p < 10^{-6}$ (wins 40, 39, 39, 40, 40, 40, and 38 of 40 in the order of the table). The two strongest opponents fail in complementary ways. The supervised regression uses the same counterfactual evidence and reaches $+42.4$, but it extrapolates "slow down" to the zero leverage and hindsight regimes (speeds 0.163 and 0.093) and is feasible on 13 seeds; CVaR-PPO leaves those regimes alone but loses speed on the flat regime (0.882) and is feasible on 19. The difference is not the evidence but the constraint to act only where verified evidence exists. Figure 6 shows the same comparison seed by seed. The spread of PAIN-PPO there follows the backbone. The 21 seeds with a gain above 50 are exactly the 21 on which the backbone runs fast in L (≥ 0.93), and on the others the backbone has already slowed down, leaving less to gain.

Table 3: **Clean Corridor, end to end** (40 seeds; mean ± SD over seeds, median in parentheses; Δ against the per-seed backbone reference; speeds are medians of the per-regime mean speed; feasible counts the seeds whose selected snapshot met the budget on the validation tapes). The failure-memory arm stores and retrieves failures in the manner of FEMA (Miao, 2026) but admits them by severity, since the corridor has no early termination to trigger FEMA's admission.

Arm	$\Delta\mathrm{CVaR}_{5\%}$	ΔMean	FLAT	L	U	H	Feasible
Backbone (reference; $\Delta \equiv 0$)	0	0	0.936	0.935	0.936	0.934	—
Backbone + same snapshot selection	$+3.8 \pm 11.2$ ($+0.4$)	$+0.04 \pm 0.29$	0.934	0.727	0.931	0.927	36/40
Shuffled retrieval keys	$+4.8 \pm 10.6$ ($+1.8$)	$+0.01 \pm 0.26$	0.929	0.680	0.927	0.922	37/40
Severity only (no counterfactual check)	$+10.7 \pm 13.5$ ($+12.8$)	-0.06 ± 0.27	0.926	0.682	0.876	0.889	30/40
Failure memory, severity admission	$+11.4 \pm 14.0$ ($+14.0$)	-0.09 ± 0.25	0.927	0.667	0.874	0.884	31/40
CVaR-PPO (optimizes CVaR directly)	$+18.3 \pm 22.6$ ($+8.9$)	-1.02 ± 1.78	0.882	0.435	0.935	0.935	19/40
Supervised regression on the same evidence	$+35.2 \pm 25.9$ ($+42.4$)	-1.28 ± 2.07	0.932	0.062	0.163	0.093	13/40
PAIN-PPO	$\mathbf{+44.7 \pm 26.4}$ ($\mathbf{+66.6}$)	$\mathbf{+1.34 \pm 1.07}$	0.937	0.379	0.938	0.931	**40/40**

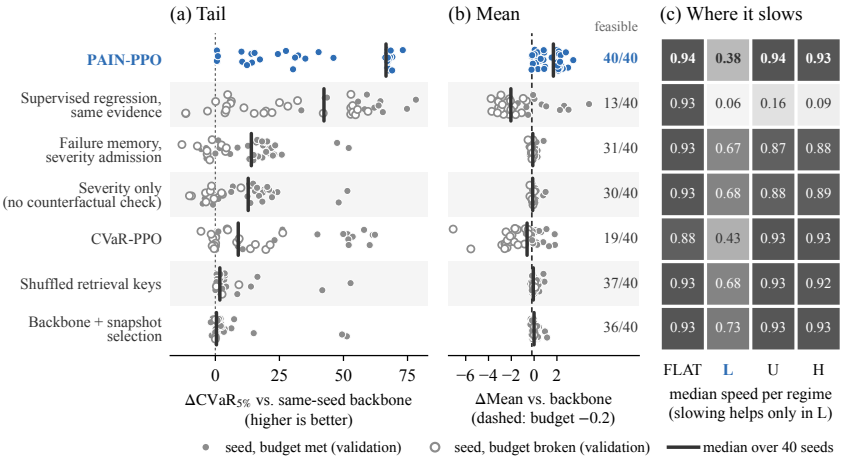


Figure 6: **Clean Corridor, every arm of Table 3, per seed** (seeds 200–239). (a) $\Delta\mathrm{CVaR}_{5\%}$ and (b) ΔMean on the evaluation tapes against the same-seed backbone. Bars are medians. Filled dots met the budget on the validation tapes, as in the Feasible column. On the evaluation tapes, which the dashed budget line uses, 36 filled dots fall left of it, among them 12 of the 31 filled failure-memory seeds and one PAIN-PPO seed (-0.227). (c) Median speed per regime (FLAT is NORMAL in Figure 5).

Table 3 is a loose-budget setting: PAIN-PPO's mean is $+1.34$, far from the -0.2 line. A second setting (20 fresh seeds, $K_L = 90$, t_{shock} fixed at 19) pins the mean to the budget. There PAIN-PPO gives $+3.25$ with a median mean of -0.021 and is feasible on 16 of 20 seeds, whereas the same system with its three consolidation protections removed (support count, direction agreement, and reliability) gives $+41.72$ at a mean of -0.980 and is feasible on 0 of 20 (two-sided Fisher $p = 1.5 \times 10^{-7}$); severity-only admission and shuffled keys give $+1.15$ and $+0.56$. Under a binding budget the protections decide deployability, and they do so through the number of entries they let through rather than through their choice of entries: on a fresh block of 20 seeds in the same setting, PAIN is feasible on 11, a random selection of the same number on 16, and keeping every single-tape verified entry on 1 (Appendix F.1.3). The tail reduction obtainable within the budget therefore depends on how tight the budget is: $+66.6$ with 40 of 40 feasible when it is loose, $+3.25$ with 16 of 20 when it binds. In the loose setting removing the protections costs nothing (on seeds 200 to 219, 20 of 20 feasible and $+36.36$ against $+36.31$ for PAIN-PPO on the same seeds).

F.1.2 Exogenous share and verifier noise

This check, shown in Figure 7, varies the mixture of the three classes while the total disaster rate stays at 0.45, following the pre-registered sweep (`prereg/PREREG_EXO.md`, seeds 500–509, 60 updates). As the exogenous and hindsight share rises from 0.1 to 0.9, severity admission stays at chance and cross-tape admission stays at precision 1.000, with a recall just under one half at every

point. End to end, PAIN-PPO has the largest tail gain at every point, keeps all 10 seeds within budget, and keeps backbone speed in U. Supervised regression on the same evidence comes closest in tail gain up to an exogenous share of 0.7, but it slows down in U at shares up to 0.5 and breaks the budget on up to six seeds of ten, and at 0.9 every arm's gain is close to zero. Failure memory stays within budget on 9 or 10 seeds of ten. The absolute gap narrows at high exogenous share because the avoidable part itself shrinks from 0.405 to 0.045 of episodes. The paired comparison with failure memory favors PAIN-PPO on 10 of 10 seeds at the first four points and on 9 of 10 at the last.

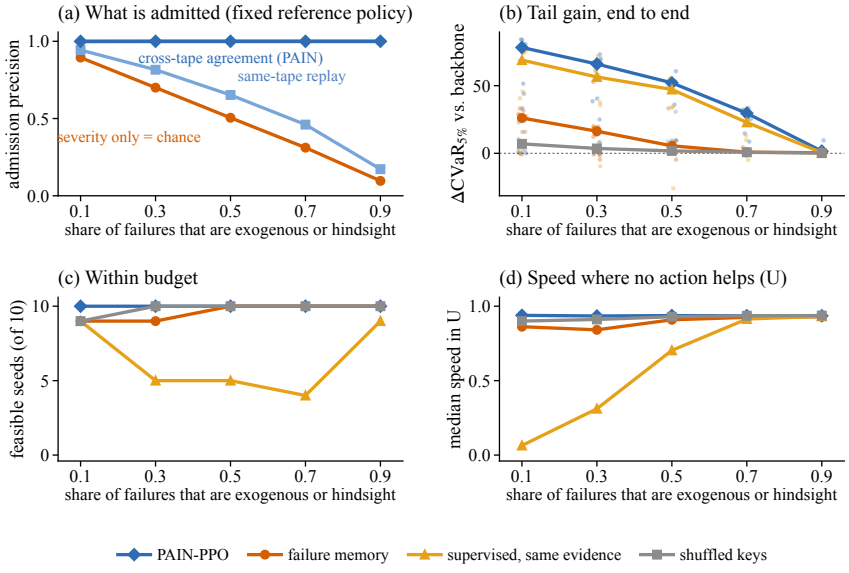


Figure 7: **Clean Corridor, varying the exogenous share** (total disaster rate 0.45, 10 seeds per point). (a) Admission precision against the class labels, with a fixed reference policy on 20,000 tapes per point. The dashed chance line coincides with severity admission. (b) Median $\Delta\mathrm{CVaR}_{5\%}$ against the same-seed backbone, one dot per seed. (c) Seeds within the mean budget. (d) Median speed in the zero-leverage regime U.

The corridor verifier is exact. To ask what happens when it is not (Figure 3a), Gaussian noise of standard deviation σ is added to every counterfactual estimate before admission; the unit of σ is one step of free progress ($\epsilon_v\,\delta_{\mathrm{probe}} = 0.2$), and the nominal net gain of one probe in L is

$$K_L\,\delta_{\mathrm{probe}}/T - \epsilon_v\,\delta_{\mathrm{probe}} = 1.0,$$

so the noise is about the size of the signal at $\sigma = 5$ (under the reference policy $v = 0.5$, the median gain of one slow-down probe on tail L tapes is 1.19). Seeds 300 to 319, loose budget, medians of $\Delta\mathrm{CVaR}_{5\%}$ against the same-seed backbone. All ten levels come from one run of the frozen corridor code, and at the four pre-registered levels it reproduces the earlier sweep seed by seed (`recompute_noise_ext.py` in the released repository):

σ	0	0.5	1.5	3	5	8	12	20	35	60
PAIN-PPO	+64.6	+62.5	+58.1	+57.4	+53.3	+41.2	+26.7	+18.7	+12.4	+7.4
without the tail condition	+64.5	+2.4	+0.7	+4.4	+2.7	+2.8	+2.2	+1.9	+1.4	+1.4
without cross-tape agreement	+64.6	+64.2	+64.4	+63.2	+62.2	+60.7	+59.8	+57.2	+52.2	+44.8

With an exact verifier the tail condition is worth nothing; with any noise, removing it collapses the gain to a few points, while the full method keeps at least 82% of it up to $\sigma = 5$ and falls below ten points only at $\sigma = 60$. The difference between the two arms is positive on 17 to 19 of the 20 seeds at every noisy level and its sign test gives $p \le 0.003$ there, against $p = 0.26$ with an exact verifier. The tail condition buys robustness to verifier error, not precision, and that is the regime a learned or model-based verifier lives in. Without cross-tape agreement the tail gain is at least as high at every level under this loose budget; that gate's contribution is admission precision (Figure 2a), and together

with the other consolidation protections it decides feasibility once the budget binds (above), not the tail.

F.1.3 SELECTION WITH THE NUMBER OF ENTRIES FIXED

A stricter admission rule can help in two ways, by choosing more reliable corrections or by intervening on fewer states. To separate them we hold the number of stored entries fixed in each batch. Every rule forms its candidates the same way, from the tail events of each batch with their same-tape probes, and the rules differ only in which keys they keep. PAIN keeps a key once three distinct tapes agree in direction, a random rule keeps the same number of keys drawn from the single-tape verified pool, and a strongest-evidence rule keeps the keys with the largest single verified gain. As a reference that is not count-matched, a keep-all rule keeps every single-tape verified key. Teaching amplitudes of the matched rules are drawn from PAIN's own amplitudes in the same batch. Four settings are run with 20 seeds each (600 to 619): the default corridor (E1), the binding budget of the previous paragraph (E2), a regime mix $(0.55, 0.15, 0.075, 0.225)$ in which most candidates do not reproduce (E3), and both changes together (E4). Every number below is regenerated by `recompute_screening.py` in the released repository. Each matched rule keeps, in each batch, as many keys as the PAIN rule would keep on that rule's own candidate pool. Since each pool follows its own policy, the matched rules accumulate 1.03 to 1.26 times PAIN's entries over training.

Offline, under a fixed reference policy and with the number kept equal, PAIN's entries reproduce on held-out tapes more often than any matched rule in all four settings, 0.824 against at most 0.777 in E1, 0.918 against 0.797 in E2, 0.859 against 0.673 in E3, and 0.856 against 0.726 in E4. End to end at $\eta = 0.5$ (Table 4), the difference is large where most candidates do not reproduce and small where few do. In E3 PAIN gains $+32.4$ against $+11.9$ for the random rule and $+13.7$ for the strongest-evidence rule, ahead on 19 of 20 seeds against each, and its mean is higher as well. In E1 the gaps are $+6.5$ and $+2.0$ (20 of 20 seeds), and the mean is significantly higher only against the random rule. A 2×2 decomposition that thins PAIN's keys to the random rule's coverage and densifies the random rule to PAIN's coverage attributes $+12.5$ $[+4.2, +20.9]$ of the E3 gap to which keys are chosen and $+0.2$ to how many, about nine tenths by the ratio of means, whereas in E1 the two parts are $+2.8$ $[+0.6, +4.3]$ and $+3.3$ $[+1.2, +6.6]$. Against keeping every single-tape verified entry, PAIN's tail is not higher under a loose budget (-1.8 $[-3.1, -0.2]$ in E3), and what it buys is a higher mean in E1 ($+0.14$, 16 of 20) and feasibility once the budget binds. Under the binding budget PAIN and the count-matched rules do not differ detectably on the deployable scale (smallest $p = 0.45$ in E2 and 0.12 in E4), while keeping every single-tape entry is feasible on 1 and 10 of 20 seeds. Selection therefore pays in the tail when many candidates are wrong, and restraint pays in the budget when it binds.

Table 4: **Clean Corridor, PAIN, per-batch count-matched controls, and a keep-all reference** (seeds 600–619, $\eta = 0.5$; medians of $\Delta\mathrm{CVaR}_{5\%}$ and ΔMean against the same-seed backbone, and seeds within budget). E1 is the default corridor, E2 the binding budget, E3 the regime mix in which most candidates do not reproduce, and E4 both. The last two columns are paired medians on the deployable scale, on which an infeasible run counts as zero, with the seed counts for and against PAIN.

	PAIN	keep all (reference)	random, count-matched	strongest evidence, count-matched	PAIN − random	PAIN − strongest
E1	+59.4 / +2.00 / 20	+60.1 / +1.71 / 20	+51.0 / +0.94 / 18	+56.6 / +1.69 / 19	+6.45 (20:0)	+2.02 (20:0)
E2	+3.3 / −0.07 / 11	+38.0 / −0.84 / 1	+1.4 / −0.08 / 16	+1.6 / −0.07 / 15	0.00 (9:8)	0.00 (10:6)
E3	+32.4 / +0.70 / 19	+32.8 / +0.75 / 18	+11.9 / +0.04 / 17	+13.7 / +0.14 / 18	+10.88 (19:0)	+7.18 (19:0)
E4	+1.4 / −0.04 / 19	+7.6 / −0.12 / 10	+1.1 / −0.06 / 20	+1.1 / −0.07 / 20	+0.20 (13:7)	+0.30 (14:6)

F.1.4 PAIRED AGAINST INDEPENDENT VERIFICATION

The verifier of Section 2.2 replays both actions on the same recorded realization. To ask whether pairing is part of the method, we replace it in the corridor by independent draws. The alternative action is evaluated on n fresh realizations drawn from the same conditional law (same regime, same observed past, same tail membership) and the factual action on n others, with admission, teaching, and budget unchanged (seeds 700 to 719, 60 updates, $\eta = 0.5$, loose budget). At $n = 1$, which costs the same two simulator calls per candidate as pairing, the tail gain is $+3.2$ against $+34.9$, and pairing is ahead on 20 of 20 seeds. The independent estimate has about 850 times the variance of the paired

one, and the direction gate then rejects most candidates. At $n = 1024$ the independent verifier reaches $+34.0$ and pairing is still ahead by $+0.33$ $[+0.10, +0.50]$ (16 of 20, $p = 0.012$); at $n = 4096$ it reaches $+34.8$ and is ahead of pairing by 0.28 $[0.13, 0.42]$ (16 of 20). Pairing is therefore an efficiency device in this environment, worth three to four orders of magnitude in simulator calls, and not a requirement (`recompute_pairing.py` in the released repository). In the sepsis simulator the independent verifier remains significantly positive at an equal sample size ($+0.033$ $[+0.015, +0.051]$ over the backbone, 13 of 16 seeds), with pairing ahead of it by $+0.013$ $[+0.001, +0.026]$.

F.2 Guarded Maze: details

The environment is `guarded_maze_cesor` of the official Return Capping repository (mode 1: 8×8, continuous coordinates, action noise 0.2, guard probability 0.05, guard cost $4 \times 16 = 64$, goal $+16$, step cost 1 for the first 32 steps). The papers of Greenberg et al. (2022) and Mead et al. (2025) describe the guard as present with probability 20% and with an exponential cost of mean 32, and we use the configuration in the code. Return Capping uses `cap_alpha`$= 0.05$ and `cap_tau`$= 0.2$. PAIN-PPO runs the same capped-return PPO, takes the worst 5% of episodes in each training batch as failure events, teaches each verified action up to probability $\eta = 0.5$ with the one-sided constraint of Appendix H, and keeps the veto on. The two replays of a comparison share both the recorded environment noise and the random numbers that the continuation policy draws, so the two actions face the same future. An earlier implementation shared only the environment noise, and none of its runs on the continuous maze are reported here. Both arms log $\mathrm{CVaR}_{5\%}$ over the episodes of each 10,000-step training batch, and there is no separate evaluation run. Figure 8 draws the maze and what each route earns.

Table 5 gives the counts on the three blocks of seeds. Averaged over the last five batches, the paired difference in $\mathrm{CVaR}_{5\%}$ (PAIN-PPO minus Return Capping) has median $+0.41$ over the 120 seeds and is positive on 74 and negative on 46 (sign test $p = 0.013$), and it is significant on the first block alone ($+0.98$, 30 against 10, $p = 0.002$). The paired difference in mean return has median -0.136. PAIN-PPO's mean return is 0.96 to 1.18 by block against 1.59 to 2.07 for Return Capping, because the mean of a policy that sometimes takes the short route includes the episodes in which the guard was absent. PAIN-PPO's replays add a median of 962,711 steps per run (942,413 to 1,225,261) to the two million environment steps. On the 76 seeds that converge under both and end at or above -10, we count the interactions up to the first batch from which $\mathrm{CVaR}_{5\%}$ stays at or above -10, with PAIN-PPO's replay steps included. The median paired difference is $+514{,}924$ steps, PAIN-PPO is cheaper on 24 seeds and costlier on 52, and the sign test gives $p = 0.002$. Figure 9 shows the training curves.

Table 5: **Guarded Maze, seeds meeting the pre-registered criterion**, out of 40 per block, for Return Capping (RC) and PAIN-PPO, with Fisher two-sided p. Steps are environment steps. At 1.5 million interactions PAIN-PPO is read after one million steps, when its replays have added about half a million more. Strict asks each of the last five batches, not only their mean, to reach -10.

	1M steps			1.5M interactions			2M steps			Strict, 2M steps	
Seeds	RC	PAIN	p	RC	PAIN	p	RC	PAIN	p	RC	PAIN
100–139	18	36	2.9×10^{-5}	23	36	0.0018	24	37	0.0012	22	36
220–259	14	34	8.9×10^{-6}	26	34	0.069	28	36	0.048	26	29
260–299	21	39	3.0×10^{-6}	27	39	6.7×10^{-4}	30	40	0.001	24	37
All 120	53	109		76	109		82	113		72	102

The repository also ships a discrete variant of the maze, introduced by Luo et al. (2023), in the form of Mead et al. (2025), in which stepping on the guarded cell costs $30\,\zeta$ with $\zeta \sim \mathcal{N}(0, 1)$, a zero-mean draw. A pre-registered pilot (20 seeds per arm, criteria fixed before the runs) was run after a defect in the discrete replay path was found and fixed (the replay drew fresh guard noise instead of the episode's own draw). As in the continuous runs, PAIN-PPO was built on Return Capping. The fixed replay admitted corrections on all 20 seeds, but the first criterion, that no PAIN-PPO seed averages below -70 over the last five batches, failed on 2 of 20 seeds. Return Capping itself reached the convergence criterion and then lost it on 11 of 20 seeds under the same configuration. By the pre-registered rule the pilot stopped before its 60-seed confirmation, and only the continuous variant is reported.

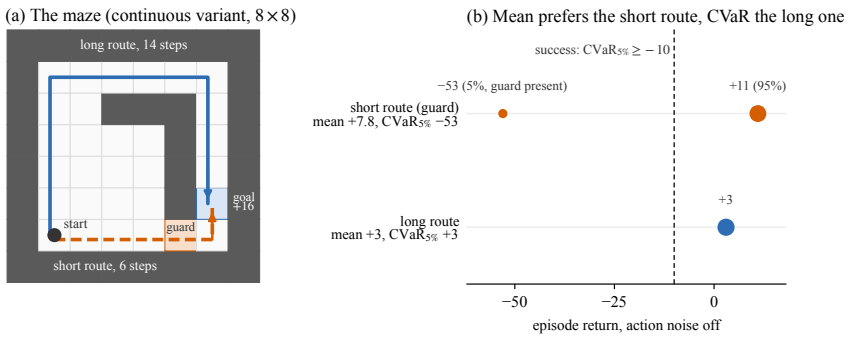


Figure 8: **The Guarded Maze as run here**, the continuous variant `guarded_maze_cesor` of the Return Capping repository (mode 1), not the discrete variant excluded below. (a) The map built by the environment's code. Moves carry Gaussian noise of standard deviation 0.2, and the guard, present with probability 0.05 and unobserved, costs 64 once. (b) Return of each route in (a), replayed with the noise off, at -1 per step for the first 32 steps, except that the step reaching the goal pays $+16$ instead. Dashed, the pre-registered criterion.

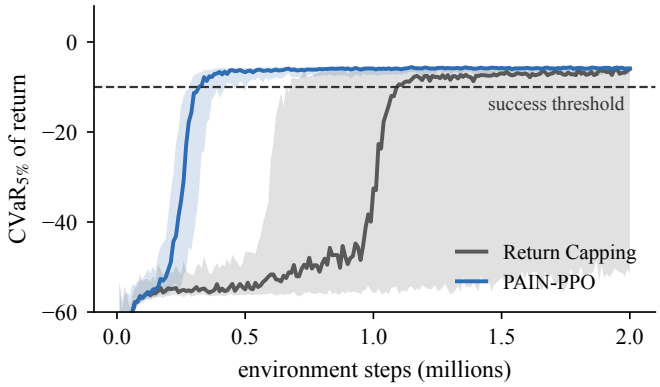


Figure 9: **Guarded Maze, training curves.** $\mathrm{CVaR}_{5\%}$ of each 10,000-step training batch, median and interquartile band over the 120 seeds of the three blocks per arm, Return Capping from the official code and PAIN-PPO from the same code with the Pain channel. The dashed line is the pre-registered success threshold. Seeds that converge reach the same plateau under both arms.

F.2.1 Retention after a fork

The convergence counts above show that the corrections are learned. Whether they are kept is tested by a fork on the two blocks on which Return Capping was also forked. Each seed that has converged by update 100 (one million steps; 36 of the 40 seeds 100 to 139 and 39 of 260 to 299 for PAIN-PPO, 18 and 21 for Return Capping) continues for 100 more updates under plain PPO with the capping off, whose mean objective favors the shortcut, and a seed relapses when its last five batches no longer meet the criterion. The first 100 updates of every branch are byte-identical to the parent run. Table 6 gives the counts, and `recompute_maze_onesided.py` in the released repository regenerates every number in this subsection. Every converged Return Capping seed relapses, and so do 69 of the 75 PAIN-PPO seeds once the Pain channel is switched off, so both parent policies lose the risk-averse route after switching to the mean-return objective. Keeping the channel lowers relapse to 33 of 75 (McNemar 19:0 and 17:0 against off on the two blocks). After the fork each stored correction is held at the probability it had reached at the fork, and a correction formed after the fork is taught to 0.9, close to the probability of the long route that the criterion needs at the forks of the maze. The retention cap $\rho_{\mathrm{P}} = 1$ was chosen from $\{0.05, 0.1, 0.2, 0.5, 1\}$ on the seed block 220 to 259, which is not used here. Under it the median probability of the verified actions stays above 0.98 throughout the continuation. The price is mean return, whose median over the last five batches falls from 6.6 and 6.7 with the channel off to 1.1 and 1.0, because the long route is worse in the mean. Revision matters as

well. The same arbitration with the entries and the reliability frozen at the fork relapses on 45 of 75 (McNemar 18:6 against the live memory, $p = 0.023$). Over the continuation the live memory grows from a median of 94 to 137 stored states, as corrections formed after the fork join those held from before.

Table 6: **Guarded Maze, relapse after the fork.** Seeds converged at update 100 that no longer meet the criterion at update 200, per block. The PAIN-PPO rows share the parent runs, the fork, and the continuation.

Continuation after the fork	Seeds 100–139	Seeds 260–299
Return Capping parent, plain PPO	18/18	21/21
PAIN-PPO parent, Pain channel off	32/36	37/39
Pain channel kept, $\rho_P = 1$	13/36	20/39
entries and reliability frozen at the fork	19/36	26/39

F.3 Sepsis simulator: details

The simulator of Oberst & Sontag (2019) has 720 observed states in the authors' code (1,440 with the diabetes indicator), eight actions, episodes of at most 20 steps, and an outcome of $+1$ at discharge, -1 at death, and 0 at truncation. We hide the diabetes indicator from the policy. The dial mixes each action's transition law toward the average over the eight actions,

$$p_\lambda(\cdot \mid s, a) = (1 - \lambda)\, p(\cdot \mid s, a) + \frac{\lambda}{8} \sum_{a'} p(\cdot \mid s, a').$$

At $\lambda = 0$ the simulator is the original; at $\lambda = 1$ all eight actions share one transition law and no death is avoidable. The grid was chosen in terms of the share of deaths that are avoidable, calibrated once on two dedicated seeds (1,500 episodes each) and frozen before any confirmatory run: $\lambda = 0.25$, 0.57, 0.81, 0.94, and 0.975 give avoidable shares of 0.80, 0.61, 0.38, 0.18, and 0.11, with mortality under the calibration policy rising from 0.41 to 0.80. The rise in mortality is a known confound of the dial, which is why the gain is read against the avoidable share and mortality is reported alongside. The main result pools $\lambda \in \{0.25, 0.57, 0.81\}$ on one segment and $\lambda = 0.25$, the level with the most avoidable deaths and the largest signal, on the others; the per-level gains ($+0.0347$, $+0.0223$, $+0.0192$) fall with the avoidable share.

The backbone is 11 features $\to$ 64 $\to$ 64 (tanh) $\to$ 8, trained with Adam at learning rate 3×10^{-3} on 256 episodes per update, for 400 warm-up updates and 1,600 further updates. Each run has an evidence budget of 12,000 proposal tapes and 100,000 verification tapes. Bootstrap intervals use 20,000 resamples over seed clusters; the three λ levels of the first segment share seeds and are averaged within seed before pooling.

F.3.1 Results and controls

Pooled over the 96 seed clusters, PAIN minus backbone is $+0.0383$ (95% CI $[+0.0301, +0.0469]$; 82 of 96 clusters positive; Wilcoxon $p = 2.4 \times 10^{-13}$). Segment S4 (12 seeds) also runs oracle targets, which add $+0.0048$ $[-0.0031, +0.0126]$ over PAIN there, an interval that includes zero, so on that segment PAIN reaches 88% of the oracle's gain. At $\lambda = 1$ the pipeline admits no lesson in 12 of 12 seeds and the gain is exactly zero.

Per segment, Pain minus backbone gives the following n, mean, 95% CI, and positive seeds (Figure 10): S1 λ=0.25: 16, $+0.0347$ $[+0.0098, +0.0626]$, 12; S1 λ=0.57: 16, $+0.0223$ $[+0.0104, +0.0351]$, 13; S1 λ=0.81: 16, $+0.0192$ $[+0.0069, +0.0328]$, 12; S2: 16, $+0.0307$ $[+0.0096, +0.0550]$, 13; S3: 16, $+0.0456$ $[+0.0266, +0.0662]$, 15; S4: 12, $+0.0350$ $[+0.0096, +0.0628]$, 8; S5: 12, $+0.0566$ $[+0.0343, +0.0816]$, 11; S6: 12, $+0.0313$ $[+0.0148, +0.0494]$, 11; S7: 12, $+0.0478$ $[+0.0196, +0.0792]$, 9.

On segments S1 to S3 at $\lambda = 0.25$ (48 seeds), mortality falls from 43.0% to 41.3% (-1.72 percentage points, $[-2.66, -0.90]$, Wilcoxon $p = 0.0005$); it falls on 30 of the 48 seeds, and the decreases are larger than the increases.

At the same failure moments, replacing the counterfactual target by the observationally associated one gives $+0.0018$ and $+0.0057$ against the backbone in two segments (intervals include zero),

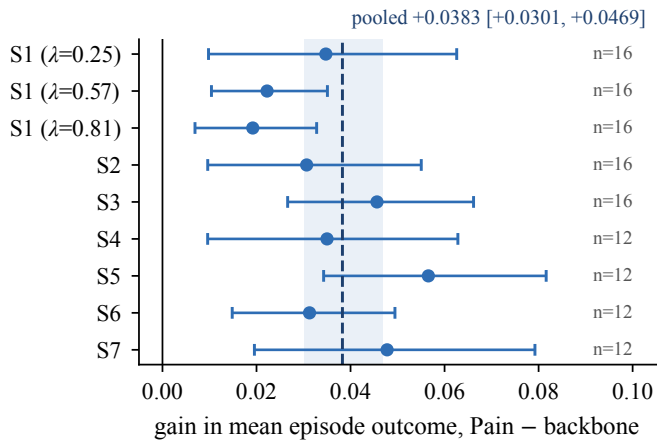


Figure 10: **Sepsis simulator, seven independent seed segments.** Gain of Pain over the same backbone in each segment (95% bootstrap CI), with segment S1 run at three levels of λ on the same seeds, and the pooled estimate over 96 seed clusters (dashed line and band).

against $+0.0307$ and $+0.0456$ for the counterfactual target. The counterfactual target is ahead of the associated one by $+0.0288$ and $+0.0399$. A severity-only memory that admits every terminal failure lowers the outcome in this simulator (it writes disease-driven deaths into memory). The offline analysis of memory content across the λ dial is regenerated by `report_v2.py` in the released repository.

Whether a learned correction survives later training is tested by a fork (24 seeds, 51000 to 51023). Every arm first warms up on low-oxygen admissions (family A), forms its corrections there, and learns them jointly for 400 updates, until the verified action has median probability 0.97 at the stored states. The policy is then saved and trained on normal-oxygen admissions (family B) only, for 1,600 updates, with the stored states' samples kept in the PPO batch at a share $f \in \{1, 0.1, 0.02, 0\}$ of their natural frequency. Without memory the probability drifts to 0.83, 0.76, 0.76, and 0.68 at the four shares, and the stored action stops being the most likely one at 17% to 31% of the stored states, while the return on family A does not move detectably, because the stored states are 7% to 10% of the steps evaluated there and the seed-to-seed interval of the change (± 0.024) is two to three times the effect. Restoring the verified side of the stored states alone in the final policy raises the family-A return by $+0.0072$ $[+0.0022, +0.0131]$ at $f = 0$ (16 seeds up, 5 down), a placebo that restores visit-matched other states gives -0.0009, and restoring the whole rows gives $+0.0160$; at $f > 0$ the restoration changes nothing detectable, so the return cost of forgetting appears only once the stored states leave the training data. The channel's one-sided retention ($\rho_{\mathrm{P}} = 0.05$, veto off) keeps the probability at 0.996 to 0.999 at every share, and its family-B return is indistinguishable from the no-memory arm ($+0.0001$ $[-0.0071, +0.0057]$ at $f = 1$). Replaying the same corrections with a log-likelihood loss pushes the probability to 1.000 but lowers the family-B return by 0.024 $[0.010, 0.041]$ and the family-A return by 0.018 $[0.005, 0.033]$ against the no-memory arm at $f = 1$. A two-sided anchor at the same target is indistinguishable from the one-sided constraint at this target, and adding the veto brings no gain. What separates PAIN from replay is that it holds each stored choice at the level already learned instead of pushing it to saturation. Figure 3b shows the $f = 1$ end points, and every number here is regenerated by `recompute_forget_fork.py` in the released repository.

An earlier experiment trains on admissions with low oxygen (family A), then on admissions with normal oxygen (family B), and evaluates on A. At the shift the stored corrections were not yet learned, with a median probability of 0.13 for the verified action at the stored states, so this experiment compares what each memory teaches across the shift rather than what it retains, and the fork above is the retention measurement. In a run of 12 seeds, the veto alone, with teaching off, is ahead of the no-memory backbone by $+0.0406$ $[+0.0214, +0.0589]$ (9 of 12 seeds). A control that removes the same gradient mass in random directions stays near the backbone ($+0.0037$ $[-0.0087, +0.0182]$), and the veto is ahead of it by $+0.0369$ $[+0.0198, +0.0544]$ (11 of 12, Wilcoxon $p = 0.0034$). On a second block of 12 seeds, counterfactual replay is ahead of experience rehearsal of the same storage size by $+0.0268$ $[+0.0119, +0.0430]$ (10 of 12). Rehearsal preserves the old behavior, and on one

probe seed its agreement with the stored actions rises from 0.920 to 0.974, but the old behavior is not worth preserving. With the same keys, axes, counts, and pipeline and only the target content swapped, replay is ahead by $+0.0403$ $[+0.0209, +0.0612]$ (9 of 12). The swapped anchor does keep its own targets, with a median agreement of 0.985 against 0.682 for the backbone, yet it is only $+0.0008$ ahead of the backbone. Without a shift the veto is nearly inert. In two diagnostic runs at a smaller evidence budget it removes 0.11% and 0.13% of the gradient.

Two other backbones were run on 12 seeds each. A 136,969-parameter ReLU backbone gives $+0.0470$, and a policy with a 144-dimensional one-hot input, which shares no state features and is trained three times as long, gives $+0.0478$. The main runs use the two Pain amplitude settings $(\rho_{\mathrm{P}}, \eta)$ of Appendix H. A separate sensitivity analysis over a 3×2 grid of these settings finds no configuration detectably better than the one used. The candidate chosen on a six-seed calibration segment is ahead by only $+0.0038$ $[-0.0054, +0.0141]$ on a fresh 12-seed segment.

F.3.2 What bounds the gain

Four follow-up analyses, on the evidence budget, the policy class, the observation, and the anchor, locate the gain's ceiling (Table 7; every number in the table is regenerated by `recompute_v14_v16.py` in the released repository). The evidence-budget, observation, and anchor experiments were pre-registered with predictions, and the policy-class search was not. The first two ran on fresh 16-seed segments, and the anchor and revision experiments reuse those segments and add a third.

Quadrupling the verification replays per candidate (1,500 to 6,000), the proposal tapes (12,000 to 48,000), and the verification pool (100,000 to 400,000) raises the median number of admitted lessons from 16 to 24.5 per run and the gain by $+0.0055$ $[-0.0010, +0.0127]$ on the same seeds, an interval that includes zero.

The policy observes the simulator's 144-state projection (the upstream `proj_obs`), which drops the glucose level although glucose is one of the four vitals that decide death and discharge. A greedy key-wise search over deterministic 144-key policies, started from each seed's backbone and scored on held-out tapes, reaches -0.120 to -0.147 with mortality between 0.406 and 0.425 on eight seeds; PAIN-PPO's policies on the same seeds land within 0.03 of that search (-0.005 on average) at the same mortality. The backbone's seed-to-seed level ranges from -0.118 to -0.257 and the gain correlates -0.92 with it, so PAIN-PPO brings every seed close to the level that the search reaches. This search was not pre-registered.

Giving the policy the full 720-state observation (glucose visible, diabetes still latent) lifts the backbone itself to -0.032 (mortality 0.307), and PAIN's additional gain is smaller and uncertain at $+0.0084$ $[-0.0075, +0.0239]$. This run uses 200,000 verification tapes instead of 100,000. The upstream projection is the protocol of the main result and is kept.

The anchor experiment starts from the arms that collapse. On one 16-seed segment, 2 Pain arms and 3 oracle arms end more than 0.03 below the backbone (3 of 48 Pain arms over three segments). Four of the five collapsed arms were rerun with their final policies and lesson keys saved, and each collapse decomposes the same way (`recompute_spillover.py`). The taught policy changes its action on 135 to 290 raw states, while the lessons cover 50 to 95. Using the taught policy only at the lesson states is never worse than the backbone, and using it only at the other states reproduces the collapse. On Pain seed 41001, for example, these two policies score -0.161 and -0.348 against -0.233 for the backbone, and on oracle seed 41014 the second one lowers discharge from 23% to 3.4%. In each diagnosed collapse the harm is spillover through the shared network into states that hold no lesson, not the lessons themselves. Re-verifying each lesson every 200 updates against its opposite direction and demoting the losers (a version of the demotion rule in Appendix H) drops 14 of 253 verified lessons and leaves the pain arm where it was, but drops 10 of 253 oracle lessons, which were never gated, and lifts that arm from $+0.0147$ to $+0.0416$ $[+0.0257, +0.0566]$.

The sepsis pipeline of the main result anchors the states that hold no lesson to the previous update's snapshot, which bounds one update's movement and not the drift; the corridor anchors them to the same-update policy of a shadow backbone. Anchoring the sepsis arms the same way, outside the arbitration cap, removes every collapse: on 48 seeds in three segments, 3 collapses become 0 and the worst seed moves from -0.073 to -0.019. It is not free. On the segment with two of the three collapses the mean is unchanged ($+0.0236$ against $+0.0250$), but on the other two

segments the anchored arm gains about 0.02 less, and over the 48 seeds the paired difference is -0.0155 $[-0.0261, -0.0044]$. Spillover through the shared network is thus on average beneficial and occasionally catastrophic, and the snapshot anchor of the main result keeps both. We also test a guard, under which the shadow anchor is switched on only while the arm's own training return, averaged over the last 50 updates, trails the backbone's by more than 0.02. On the same 48 seeds it fires on at most 3% of the updates and never on 33 seeds, whose runs are therefore bit-identical to the baseline; the three collapses recover (-0.043, -0.073, and -0.033 to $+0.030$, -0.020, and $+0.016$), the worst seed moves from -0.073 to -0.020, and the mean gain is $+0.0390$ $[+0.0288, +0.0496]$ against $+0.0356$, a paired difference of $+0.0034$ $[-0.0008, +0.0084]$. The main result above was produced with the snapshot anchor and is reported as pre-registered; the guard is the candidate revision of the pipeline.

Table 7: **Sepsis follow-ups** (16 seeds per row unless 48 are marked, with the seed block of each experiment in the first column; mean and 95% bootstrap interval over seeds; "collapses" counts seeds where the arm ends more than 0.03 below the backbone). Baseline rows use the snapshot anchor of the main result.

Experiment	Arm	Gain over backbone	Positive	Collapses
Evidence budget (seeds 40000+)	Pain, standard budget	$+0.0476$ $[+0.0249, +0.0713]$	14/16	0
	Pain, $4\times$ verification and proposals	$+0.0530$ $[+0.0310, +0.0763]$	15/16	0
Observation (seeds 41000+)	Pain, 144-state projection	$+0.0250$ $[+0.0045, +0.0442]$	13/16	2
	Pain, full 720-state observation	$+0.0084$ $[-0.0075, +0.0239]$	11/16	1
Anchor (seeds 41000+)	Pain, shadow anchor	$+0.0236$ $[+0.0129, +0.0356]$	14/16	0
	Oracle targets, snapshot anchor	$+0.0147$ $[-0.0188, +0.0450]$	10/16	3
	Oracle targets, shadow anchor	$+0.0197$ $[+0.0074, +0.0324]$	11/16	0
Anchor, three segments (48 seeds)	Pain, snapshot anchor	$+0.0356$ $[+0.0234, +0.0479]$	40/48	3
	Pain, shadow anchor	$+0.0201$ $[+0.0129, +0.0275]$	35/48	0
	Pain, return-guarded shadow anchor	$+0.0390$ $[+0.0288, +0.0496]$	43/48	0
Revision (seeds 41000+)	Pain, with revision	$+0.0263$ $[+0.0043, +0.0467]$	13/16	1
	Oracle targets, with revision	$+0.0416$ $[+0.0257, +0.0566]$	14/16	0

F.4 A-SHARE EXECUTION: DETAILS

The reported system has three layers. (i) Admission: gradient-boosted trees over 33 pre-open features; a stock-day with score > 0 is handed to the policy, otherwise it is executed by POV from step 0. Labels come from the training months, where a fixed menu of four plans is replayed on the same tape. The trees predict each plan's gain over POV and the probability that the stock-day falls in POV's worst 5%, and the score is the largest over the four plans of that probability times the plan's predicted gain on such days plus a quarter of its predicted average gain. The release rate is 54 to 71%. (ii) Amplitude: PAIN-PPO as in Section 3 with a Lagrangian mean budget and $b \leq 0.2$, with this upper bound selected on the January–May training months and fixed across all seeds and test months. The fixed-b training-month diagnostic in Table 8 has the largest tail gain at $b = 0.2$ among the four sizes tested; larger fixed sizes reduce mean performance and $b = 0.5$ also lowers the tail. (iii) Regime layer: five cross-sectional market-context features and the first-minute market return m_1 enter the observation, and the budget constraint is applied separately to predicted sharp-drop days ($m_1 < -2.8$) and ordinary days, binding only on ordinary days.

Table 8: **Fixed-b diagnostic on the training months** (January to May, a panel of 71,474 stock-days sampled from the full market). The admitted stock-days hold b from 09:31 and the others follow POV. This diagnostic compares fixed sizes; the reported policy instead chooses $b \leq 0.2$ at each step. ΔMean and $\Delta\mathrm{CVaR}_{5\%}$ against POV in bp.

b	0.1	0.2	0.3	0.5
ΔMean	-0.61	-4.95	-12.87	-35.22
$\Delta\mathrm{CVaR}_{5\%}$	$+15.8$	$\mathbf{+24.3}$	$+19.5$	-23.2

Table 9 reports the 120-update comparison, selected on the January–May training months, alongside the 60-update runs evaluated on the test months. Table 10 lists every seed at 120 updates, and all 15 seed-months are feasible. These test-month results were not used to choose the b cap or the number of updates.

Table 9: **A-share execution, three test months.** $\Delta\mathrm{CVaR}_{5\%}$ (ΔMean) against POV in bp, mean $\pm$ SD over 5 seeds; a seed-month is feasible when $\Delta\mathrm{Mean} \geq -0.2$. Upper block: the main comparison at 120 updates; the Pain-off arm is the same recipe retrained with the Pain channel disabled, same seeds; the supervised selector is a single model. Lower block: runs at 60 updates, namely the pre-registered 60-update run of the reported recipe, update 60 of the Pain-off retrain, and development arms whose cap on b is 0.3, the last three being ablations of that development recipe. POV's own worst 5% against the arrival price is -375, -269, and -334 bp.

Arm	July	August	June	Feasible
POV (reference)	0 (0)	0 (0)	0 (0)	—
Supervised plan selector	23.6 (0.83)	8.9 (1.10)	14.1 (2.21)	3/3
Pain channel off (same recipe, 120 updates)	54.2 ± 6.3 (4.34 ± 1.98)	18.3 ± 3.3 (1.13 ± 1.13)	29.1 ± 2.4 (3.25 ± 1.31)	15/15
PAIN-PPO (120 updates)	$\mathbf{66.2 \pm 6.0}$ (4.49 ± 1.14)	$\mathbf{27.7 \pm 4.8}$ (3.54 ± 1.22)	$\mathbf{36.8 \pm 2.6}$ (4.51 ± 0.73)	**15/15**
PAIN-PPO, pre-registered 60-update run	61.7 ± 6.0 (4.57 ± 1.49)	24.0 ± 3.6 (2.47 ± 1.00)	34.9 ± 2.4 (4.21 ± 0.81)	15/15
Pain channel off, update 60 of the retrain	47.6 ± 10.0 (3.91 ± 2.10)	14.7 ± 7.2 (0.86 ± 1.43)	25.4 ± 4.6 (2.93 ± 1.03)	15/15
PAIN-PPO with $b \leq 0.3$ (development recipe)	62.7 ± 7.0 (2.29 ± 2.41)	25.9 ± 4.5 (1.67 ± 1.57)	36.9 ± 2.7 (3.80 ± 1.16)	13/15
no m_1 observation	39.7 ± 5.6 (2.65 ± 0.61)	13.2 ± 2.9 (0.12 ± 0.52)	27.0 ± 2.7 (2.89 ± 0.42)	14/15
no regime layer	35.1 ± 8.7 (1.79 ± 0.83)	15.0 ± 3.3 (1.52 ± 1.28)	19.4 ± 5.0 (0.95 ± 0.99)	13/15
no regime layer, Pain channel off	22.1 ± 4.9 (-0.56 ± 2.01)	9.3 ± 3.3 (-0.69 ± 0.83)	10.1 ± 2.2 (-1.57 ± 1.43)	4/15

Table 10: **Per-seed results at 120 updates.** ΔMean / $\Delta\mathrm{CVaR}_{5\%}$ against POV, bp.

Seed	July	August	June
0	+4.67 / +73.5	+5.09 / +34.5	+4.28 / +40.7
1	+2.87 / +62.9	+2.60 / +25.5	+3.50 / +35.0
2	+6.07 / +71.6	+4.51 / +30.6	+5.13 / +38.0
3	+4.46 / +60.6	+3.30 / +24.7	+5.32 / +34.6
4	+4.37 / +62.2	+2.23 / +22.9	+4.33 / +35.8

F.4.1 SHARP-DROP DAYS AND TRADING-DAY INTERVALS

The regime decomposition below is for seed 0 at 60 updates, and predicted sharp-drop days are those with m_1 below a threshold fixed on the training months.

Month	Tail, sharp-drop days (no regime layer)	Mean, sharp-drop	Mean, ordinary	Load ratio sharp/ordinary (p, b)	Acting stock-days sharp / ordinary
July	+79.2 (+29.7)	+10.54	−1.05	9.2×, 3.9×	93% / 28%
August	+32.5 (+16.2)	+7.81	−0.90	2.0×, 1.8×	100% / 68%
June	+56.6 (+20.0)	+10.65	−1.81	1.3×, 1.1×	88% / 84%

All five seeds of this 60-update run show the same structure: the sharp-drop tail difference against the same-seed baseline is positive in 15 of 15 seed-months, and ordinary-day ΔMean lies between -0.9 and -2.4 bp.

The main text compares systems seed by seed. Table 11 gives the same comparisons at the level of trading days, the unit at which observations are independent (Appendix E). Against the supervised selector, the tail margin of PAIN-PPO excludes zero in every month (Figure 11 shows it seed by seed), while that of the Pain-off retrain excludes zero only in June. The tail difference between PAIN-PPO and the Pain-off retrain excludes zero in every month. No mean difference is resolved at this level except that between PAIN-PPO and the Pain-off retrain in August.

Table 11: **Paired comparisons at the trading-day level** (120 updates, five-seed mean, same stock-days; PAIN-PPO and the Pain-off retrain share the admission layer, while the supervised selector chooses its own stock-days; bp; 95% trading-day-clustered bootstrap intervals, 1,500 resamples).

	Comparison	July (10 days)	August (8 days)	June (8 days)
$\Delta\mathrm{CVaR}_{5\%}$	PAIN-PPO − supervised	$+42.6$ $[+2.1, +69.5]$	$+18.8$ $[+3.6, +33.6]$	$+22.7$ $[+9.2, +37.7]$
	Pain-off − supervised	$+30.6$ $[-7.5, +55.5]$	$+9.4$ $[-2.8, +21.6]$	$+14.9$ $[+1.2, +30.3]$
	PAIN-PPO − Pain-off	$+12.0$ $[+4.1, +20.8]$	$+9.3$ $[+5.4, +12.6]$	$+7.8$ $[+5.6, +11.0]$
ΔMean	PAIN-PPO − supervised	$+3.66$ $[-3.06, +11.16]$	$+2.45$ $[-1.45, +8.00]$	$+2.30$ $[-4.72, +11.04]$
	Pain-off − supervised	$+3.51$ $[-2.60, +10.49]$	$+0.03$ $[-2.71, +4.08]$	$+1.05$ $[-3.96, +7.71]$
	PAIN-PPO − Pain-off	$+0.15$ $[-1.18, +1.64]$	$+2.42$ $[+0.67, +4.52]$	$+1.26$ $[-1.70, +3.88]$

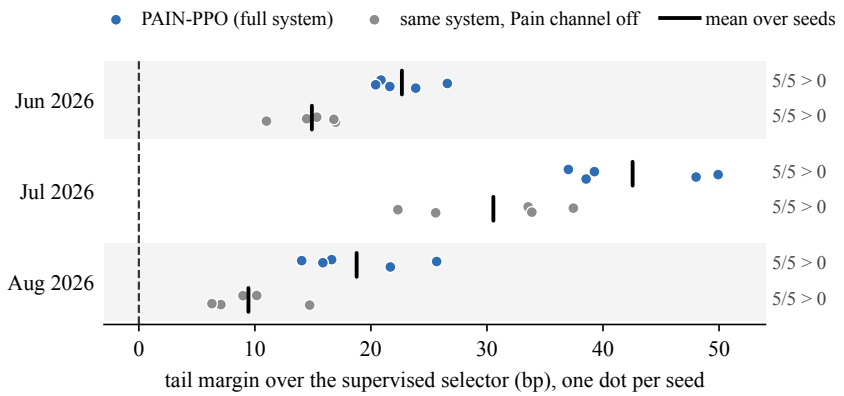


Figure 11: **Tail margin over the supervised selector, per seed-month** (120 updates). Each dot is one seed's $\Delta\mathrm{CVaR}_{5\%}$ minus the selector's in the same month, for PAIN-PPO (blue) and the Pain-off retrain (gray), and bars are means over seeds. Trading-day-level intervals are in Table 11.

At 120 updates the trading-day-clustered interval of the five-seed mean is $+4.49$ $[-1.32, +13.28]$ bp in July, $+3.54$ $[-2.23, +10.85]$ in August, and $+4.51$ $[-4.50, +16.93]$ in June. All fifteen seed-month point estimates meet the budget, between $+2.2$ and $+6.1$ bp, while these intervals cross -0.2 in all three months. The evidence for the tail improvement is therefore stronger than the evidence that the mean stays within budget. Two things make the intervals wide. The first is the structure of the data. Stock-days on the same trading day share the market's move, so a month holds only 8 to 10 independent observations. The second is the structure of the gain, because one or two sharp-drop days carry each month's mean. Removing the single best day in mean leaves July at $+0.52$ and August at $+0.18$, inside the budget, and puts June at -0.99; removing the two best days puts every month below the budget (-0.84, -1.47, and -2.64). A month's mean is therefore a statement about its sharp-drop days, which is the insurance structure of the regime decomposition above.

The days that carry a month's result differ greatly from one another. On ordinary days PAIN-PPO's daily ΔMean stays close to its average of -0.97, -0.88, and -1.59 bp in July, August, and June, with day-to-day standard deviations of 1.7, 2.9, and 0.9 bp. Its gain is concentrated on predicted sharp-drop days, those whose first-minute market return falls below a threshold fixed on the training months. There are 5, 3, and 4 such days in the three months, and on them the daily ΔMean ranges from -14.1 to $+42.7$ bp. Across all days the standard deviation of the daily ΔMean is 13.5, 10.7, and 16.8 bp, more than fifty times the budget of 0.2 bp. A month is therefore a sample of three to five draws of a very volatile quantity, and its interval is wide. The seed-level comparisons in the main text hold these days fixed. They show how consistent the result is across training runs, not how it would change on other days.

F.4.2 Ablations and the mean–tail trade-off

Table 9 collects the arms. Disabling the Pain channel in the full system (same recipe, same five seeds, retrained; no Pain telemetry in its training logs) leaves feasibility at 15 of 15 and lowers the tail to 54.2 ± 6.3, 18.3 ± 3.3, and 29.1 ± 2.4 bp at 120 updates (July, August, June). The mean is lower by 0.15 to 2.42 bp without the channel, most in August. Removing admission from the reported system makes 3 of 5 August seeds infeasible and none in July or June. The remaining ablations are test-month evaluations of development arms at 60 updates on a recipe whose cap on b is 0.3 instead of 0.2. That recipe reaches a similar tail (62.7 ± 7.0, 25.9 ± 4.5, and 36.9 ± 2.7) but has one seed over budget in July and August (13 of 15 feasible). On it, removing admission makes 3 of 5 July seeds and 5 of 5 August seeds infeasible. Removing the m_1 observation while keeping the bucketed constraint leaves 14 of 15 feasible and lowers the tail to 39.7 ± 5.6, 13.2 ± 2.9, and 27.0 ± 2.7. Removing the regime layer leaves 13 of 15 feasible, and removing it together with the Pain channel leaves 4 of 15. Switching externally between a drop-day policy and an ordinary-day policy (four such pairs) keeps all 12 seed-months feasible; three of the four pairs reach the single policy's tail and the fourth stays far below it (18.2, 8.5, and 10.1 bp). The single policy is reported as the main method because it

needs neither selection nor a switch. Two alternative targets for the regime-layer constraint were rejected on their pre-registered criteria.

The replay assumes that the policy's own orders do not move the market beyond what the recorded book already charged. To bound what a market impact would change, we add an open-loop penalty to every fill of the frozen policies, the Pain-off retrain, the selector's plans, and POV, and recompute the three months (Table 12). Two forms are used, a permanent linear impact and a transient impact decaying with time constant $\tau \in \{60, 300, 900\}$ s, each at κ from 0.25 to 10 bp per 1% of window volume. At $\kappa = 1$ the permanent form charges POV 5.8 bp, which matches the closed form for an all-active, evenly spread sale of 12% of the window. The ordering PAIN-PPO, Pain-off, selector holds in every model, at every κ, and in every seed-month, all fifteen seed-months stay feasible, and no first failing κ exists up to 10. The permanent form favors PAIN-PPO, which fills 14% to 32% of its volume passively against none for POV. The transient form at $\tau = 300$ s is the most adverse and costs PAIN-PPO 1.6, 1.5, and 1.5 bp of tail at $\kappa = 10$. The penalty is open loop, the policies do not react to it, and its form and scale are assumptions rather than estimates on these data. The evaluation output is in the released repository, and the tick-level replays it was computed from are not.

Table 12: **A-share execution under an added market-impact penalty** ($\Delta\mathrm{CVaR}_{5\%}$ against POV in bp, five-seed means, July / August / June, 120 updates). The last column is the smallest seed-month margin of PAIN-PPO over the Pain-off retrain.

Impact model	κ	PAIN-PPO	Pain-off	selector	smallest margin
none	0	66.2 / 27.7 / 36.8	54.2 / 18.3 / 29.1	23.6 / 8.9 / 14.1	4.7 / 5.1 / 5.9
permanent linear	1	66.9 / 28.4 / 37.5	54.7 / 18.8 / 29.5	24.0 / 9.2 / 14.4	4.8 / 5.2 / 6.0
permanent linear	10	72.9 / 34.7 / 43.3	59.1 / 23.2 / 33.8	27.6 / 12.1 / 17.0	5.8 / 6.7 / 6.6
transient, $\tau = 60$ s	10	65.5 / 27.1 / 36.3	53.7 / 17.9 / 28.6	23.4 / 8.7 / 13.9	4.6 / 5.0 / 5.9
transient, $\tau = 300$ s	10	64.6 / 26.1 / 35.3	52.8 / 16.9 / 27.5	22.9 / 8.0 / 13.3	4.6 / 5.1 / 6.1
transient, $\tau = 900$ s	10	66.2 / 28.0 / 36.8	53.9 / 17.9 / 28.4	23.6 / 8.6 / 13.8	5.0 / 5.7 / 6.4

F.4.3 Memory, training checks, and scope

The correction memory grows during training while its influence on the PPO update shrinks. In the reported execution system (120 updates, 5 seeds, medians over seeds), the memory holds 416 entries after the first refresh, 1,644 at 60 updates, and 1,942 at 120 updates. Each refresh verifies about 500 new events (271 to 877), and about two thirds of them merge into existing entries rather than creating new ones. The share of entries with positive authority rises to 42% by 120 updates, and about half of the authorized constraints are violated at a given reading. The veto strength q falls from 0.13 at 15 updates to 0.016 at 60 and 0.004 at 120. Over all readings with authorized memory, the PPO gradient keeps a median of 99.5% of its norm after the veto, and never less than 89%. In the sepsis simulator the memory stays small, with a median of 17 entries per seed (3 to 21, the 12 confirmatory seeds at $\lambda = 0.25$). There the teaching term acts on every update at about 5% of the PPO gradient norm, and the veto fires on a median of 6.8% of updates (0.2% to 19.9%). In neither arena does the memory freeze the policy.

The pre-registered training length was 60 updates. On the January–May training months, mean gain continued to increase from 60 to 120 updates and performance became more stable across seeds. The 120-update length for both arms was selected on these training months and fixed before the June–August evaluation. The 60-update run is also reported in Table 9. All runs of the reported recipe use 56 rollout processes.

No network weight, threshold, b cap, or training length was fit or selected on the three test months. The b cap of 0.2, the 120-update training length, the m_1 threshold, the bucket edges, and the admission-score weight 0.25 were selected using only the January–May training months; the admission-score weight was chosen by leave-one-month-out prediction under the same budget rule. The 120-update length was motivated by rising mean gain and greater across-seed stability from 60 to 120 updates on those training months. Sharp-drop days that the first-minute return does not predict, such as 21 August, lose 4.95 to 6.91 bp under all five seeds, and the training months contain one continuation and one reversal, which cannot be told apart. Each month has 8 to 10 trading-day clusters, 3 to 5 predicted sharp-drop days, and 2 to 4 realized ones (POV below -20 bp), so tail intervals are wide

by construction. The tape is exogenous: replay assumes that the agent's own orders do not change the behavior of other participants, with a parent order of 12% of window volume. The per-regime dual constraint used here was developed on this execution line. The corridor enforces a single mean budget, and the Guarded Maze and the sepsis simulator have none, so the other arenas do not exercise it.

G The Problem Class, Measured: Audit Summary and Negative-Results Register

This appendix holds the measurements behind features P1 and P2 of Section 1, all taken on the execution data before the method was built.

An episode executes a parent sell order over a fixed intraday horizon on one stock-day *tape* ω, the recorded event stream replayed under a small-agent assumption with order-level queue reconstruction and price-time priority (Appendix D). Implementation shortfall against the arrival price P_A,

$$\mathrm{IS} = 10^4 \sum_i q_i(P_i - P_A)/(Q_0 P_A),$$

where fill i sells q_i shares at price P_i out of a parent order of Q_0 shares, is signed so that larger is better and is the objective. A deterministic *Router* provides the execution prior and policies act as bounded residuals around it. The audit uses the January to May 2026 full market ($\sim 1.9 \times 10^5$ stock-days) on a data build later corrected for a reconstruction defect (Appendix D), except the per-axis value map below, which uses April subsamples. Every PAIN-PPO result in Section 5.5 uses the corrected build.

G.1 Features P1 and P2

For P1, bad outcomes are mostly the world. An additive mechanism decomposition of episode IS, exact to machine precision, attributes about 91% of the shortfall in POV's worst 5% of episodes to drift and timing exposure and about 9% to book-walk impact. This share was re-measured on the corrected build over seven months and stays between 88 and 92% for tail definitions from the worst 0.5% to the worst 25%. At action level the same action pays $+8.8$ bp on a favorable day and -9.3 bp on an adverse day, while the across-day average surface is flat at sub-bp scale. Naive pain, punish what preceded a bad outcome, would be trained almost entirely on exogenous events.

For P2, hindsight value is abundant and causal value is scarce. A pre-specified per-axis audit (hindsight one-deviation ceilings, adaptive-versus-constant decomposition D_k, causal learnability probes C_k, and 1D residual RL; Appendix G.2) found hindsight ceilings on essentially every execution axis (D_k up to 2.3 bp, clustered intervals excluding zero) while causal probes were almost everywhere null or negative, and a six-axis residual learner with paired credit ties the best fixed policy out of sample (difference $+0.000$ bp). The one adaptive checkpoint with a tail gain over its own matched constant ($+0.44$ bp $[+0.26, +0.65]$, concentrated in the constant's worst 1% at $+1.21$) breaks the mean budget under the canonical evaluation (below), and no static configuration is feasible in every month (the pooled-best static survives only by cross-month subsidy, per-month means $-0.98/-0.06/+0.66$). Counterfactual possibility is not a reproducible causal lesson, and the true lessons live in rare states.

Table 13 summarizes the per-axis hindsight ceilings D_k against the causal probe values C_k behind P2 above. The ceilings come from 100 April stock-days with a 600-second horizon, the probe values from probe runs on 45 and 150 April stock-days, and all numbers are on the pre-fix data build.

The audit's backbone, a bounded residual policy around the Router with same-tape paired credit, an episode-CVaR surrogate, and a pooled-dual mean constraint (Section 2.1), produced one checkpoint that first looked deployable. On an April replay of 39,691 paired episodes it was within the mean budget against POV (-0.177 bp) and beat the train-selected best static risk policy by $+1.06$ bp in $\mathrm{CVaR}_{5\%}$. On a separate set of 19,854 April episodes, its CVaR gain against its own matched constant, the fixed policy equal to its average action replayed on the same tapes, was $+0.44$ bp $[+0.26, +0.65]$ in a paired bootstrap with 5,000 resamples, concentrated in the constant's worst episodes ($+1.21$ on the worst 1%). That replay drew its episode windows differently from the canonical protocol and dropped stock-days with crossed books, which made the case set easier. Under the canonical protocol

Table 13: Per-axis hindsight ceilings D_k versus causal probe values C_k (bp; stock-day-clustered CIs; pre-fix data build). Hindsight value is abundant; causal survivors are scarce.

Axis	Group	D_k [CI]	Causal C_k
c release cadence	sched.	2.32 [1.88, 2.84]	$+0.07 - +0.11$ (only survivor)
p passive fraction	mode	1.42 [1.26, 1.59]	-0.21 (regime-dependent)
u urgency	sched.	1.33 [1.14, 1.53]	≈ 0 (timing beta)
x stale rescue	resting	1.11 [0.85, 1.42]	≈ 0 (tail lever; see text)
r queue refresh	resting	0.79 [0.62, 0.98]	≈ 0
ℓ, g, e, b, L (five more)	—	0.09–0.65	null / low priority

with four windows per stock-day (159,304 episodes, stock-day-clustered intervals), the tail gain over the static policy remains ($+0.73$ bp [$+0.61, +0.85$]), but the mean against POV is -0.334 bp [$-0.437, -0.232$], outside the budget, and the static policy is itself infeasible (-0.318). The result is single-seed on the pre-fix data build. It shows the shape of adaptive value in this domain, rare, conditional, and concentrated in disasters, and it shows that this backbone paid for its tail gain with more mean than the budget allows.

G.2 Audit Chain and Negative Results

The audit chain has four layers for each action axis k with Router π_R and deviation grid $\Delta_k \ni 0$: **Layer I**, a sequential one-deviation hindsight oracle (re-plan at every boundary by forward counterfactual rollout on the same tape) gives an attainable clairvoyant ceiling; **Layer II**, D_k subtracts the best recalibrated constant—small D_k with a large ceiling means re-tuning, not learning; **Layer III**, a fixed three-learner probe ladder maps causal state to counterfactual action values, evaluated by closed-loop rollout (C_k), as a screen rather than a final verdict; **Layer IV**, bounded 1D residual RL with pure-PPO, residual-only, and unpaired ablations. Same-tape pairing removes shared-path noise but not action–future interaction, which is what the privileged-future controls disambiguate. Instrument validation: with a planted near-ceiling signal, a supervised mapping captures about half the oracle ceiling while vanilla PPO captures zero until credit assignment is repaired (dense telescoping rewards plus paired credit), after which PPO reliably learns the planted signal (3/4 seeds, no anti-learning); the tested learners recover the planted signal but do not show a reliable gain over the best-fixed baseline on the market data.

The negative-results register for the mean objective has six entries. (i) *Urgency/pacing*: oracle headroom correlates -0.61 to -0.86 with future returns, so it measures price-timing exposure, and the optimal macro-pacing direction flips sign from training to test data. (ii) *Passive fraction*: it passed leverage, observability, frozen-threshold deployment ($+0.72$ bp, $t = 4.6$), and fill-model robustness, but did not beat per-regime re-fit constants ($+0.65$ in-month against $+0.09$, not significant, out-of-month), because the regime it depends on is not predictable in advance. (iii) *Dynamic cancellation*: keeping resting orders is the strongest deployable static rule, timing signals reverse out of sample (direction AUC 0.35–0.41), and the attainable amplitude is below 0.06 bp. (iv) *Fill-assumption sensitivity*: apparent passive alpha scales with fill-model optimism, so claims are stated on increments. (v) *Signal mining*: direction AUC decays with horizon ($0.59 \to 0.54$), and the AUCs of volatility features track volatility rather than direction. (vi) *Cross-month fragility of conditioning*: a later conditioning policy that passed two months had every checkpoint infeasible on full-universe replay of a third, so feasibility is required in every month separately.

For the risk actuators, a $\pm$-branch scoring calibration overturned earlier signed-aggregation false negatives; the stale-rescue actuator x acted only through the passive default removed in Appendix D, and then only as extra liquidation speed; the mechanism decomposition constrains deployable risk levers to exposure/speed. The audited risk action space is $a = (b, p, \Delta\rho)$—inventory-risk buffer, passive exposure, bounded pace residual—each verified live and non-degenerate, with $z = 0$ reproducing the pure-pace anchor exactly.

H Full Algorithm and Implementation Variants

Algorithm 1 gives one full update iteration of the canonical PAIN-PPO instantiation.

Algorithm 1 Canonical PAIN-PPO policy-update iteration

Require: Current policy parameters θ_u; committed correction memory $\mathcal{M}_{u-1}$
Ensure: Updated policy parameters θ_{u+1}; refreshed correction memory $\mathcal{M}_u$

Collect an on-policy rollout $\mathcal{B}_u$ under π_{θ_u} and form L_{PPO}
Stage 1: persistent constraint-based policy update
Evaluate L_{P} on $\mathcal{M}_{u-1}$ using Eq. (3)
$g_{\mathrm{A}} \leftarrow \nabla_\theta L_{\mathrm{PPO}}$; $g_{\mathrm{P}} \leftarrow \nabla_\theta L_{\mathrm{P}}$
$g_{\mathrm{P}}^{\mathrm{corr}} \leftarrow$ bounded correction, Eq. (4); $q \leftarrow R_{\mathrm{P}} F_{\mathrm{P}}$ ▷ Appendix H
$g'_{\mathrm{A}} \leftarrow g_{\mathrm{A}}$, replaced by Eq. (4) when $\langle g_{\mathrm{A}}, g_{\mathrm{P}} \rangle < 0$
$\theta_{u+1} \leftarrow \textsc{OptimizerStep}\big(\theta_u,\ g_{\mathrm{final}} = g'_{\mathrm{A}} + g_{\mathrm{P}}^{\mathrm{corr}}\big)$
Stage 2: counterfactual verification, on the snapshot θ_u
$\mathcal{M}_u \leftarrow \mathcal{M}_{u-1}$
if memory refresh is scheduled **then**
  for each candidate decision (t, a_0, ω) selected from the severe episodes of $\mathcal{B}_u$ **do**
    $G_i \leftarrow \Delta_t(a_i; a_0, \omega)$ for the local probes $a_1, \ldots, a_K$ ▷ Section 2.2
    $h_i \leftarrow$ directional excess gains, Eq. (2); form the candidate correction ▷ Section 3.1
  end for
  Stage 3: cross-realization consolidation
  Match each candidate against stored evidence at similar states
  Consolidate support across distinct realizations ▷ Section 3.1
  Authorize an entry once $n_j \geq n_{\min}$, and commit $\mathcal{M}_u$ ▷ Section 3.1
end if

H.1 Backbone and Action Parameterization

For the backbone (channel A), we use PPO (Schulman et al., 2017) with GAE (Schulman et al., 2016) ($\gamma = 0.99$, $\lambda_{\mathrm{GAE}} = 0.95$) and Adam (Kingma & Ba, 2015). In risk-sensitive tasks the advantage is the Lagrangian advantage $\widehat{A}_t^{\mathrm{risk}}$ of Section 2.1, which combines a tail term and a mean term, normalized by their RMS for step-size stability; the critic is a value regressor and the tail term uses the stop-gradient batch quantile as plug-in threshold. The advantage is *not* clipped at ± 3RMS: clipping truncates exactly the deepest tail, which is the part of the signal a CVaR objective needs. For the plug-in tail weight $A = \alpha^{-1}\mathbf{1}\{J \leq \nu\}(J - \nu)$ with $\nu = \mathrm{VaR}_\alpha(J)$, the Rockafellar–Uryasev representation (Rockafellar & Uryasev, 2000; 2002) gives

$$\mathbb{E}|A| = \tfrac{1}{\alpha}\mathbb{E}[(\nu - J)_+] = \nu - \mathrm{CVaR}_\alpha(J), \qquad \{|A| > c\} = \{J < \nu - \alpha c\} \quad (c > 0).$$

Thus truncating A to $\max(A, -c)$ clips exactly the episodes more than αc below the VaR; the implementation disables this clipping for the tail term.

In execution, the policy is a bounded residual around the reference policy: with latent Gaussian heads $z = (z_b, z_p)$ and scaling vector s_{act}, the executed action is

$$a = \mathrm{clip}\big(a^{\mathrm{ref}} + s_{\mathrm{act}} \odot \tanh z\big),$$

so $z = 0$ reproduces the reference. The retention constraint of Eq. (3) is defined on the latent z, whose gradient survives at physical clip boundaries.

For physical-to-latent canonicalization, let

$$D_s(z) = \mathrm{clip}\left(a^{\mathrm{ref}}(s) + s_{\mathrm{act}} \odot \tanh z\right)$$

be the actor's physical decoder, including its physical action bounds. Its numerical inverse I_s normalizes the action residual componentwise, clips it to the finite inverse range $[-1 + \delta_z, 1 - \delta_z]$ with endpoint tolerance $0 < \delta_z < 1$, and applies artanh:

$$I_s(a) = \mathrm{artanh}\big(\mathrm{clip}\big((a - a^{\mathrm{ref}}(s)) \oslash s_{\mathrm{act}}, -1 + \delta_z, 1 - \delta_z\big)\big),$$

where $\oslash$ denotes element-wise division. The endpoint tolerance is separate from the loss stabilizer ϵ_{num}. Canonicalization is the round trip $\mathcal{C}_s(a) = D_s(I_s(a))$, applied to physically feasible probe targets and factual actions. The stored latent representative is $I_s(\mathcal{C}_s(a))$; the implementation checks the idempotence $\mathcal{C}_s(\mathcal{C}_s(a)) = \mathcal{C}_s(a)$. This selects a finite representative even when physical clipping

makes the decoder non-injective. Targets outside the reachable residual range are clipped to the representable boundary, and corrective displacements are computed only after this canonicalization.

For discrete action spaces, as in the sepsis simulator, the displacement of a lesson is measured by the probability mass that the policy puts on the actions agreeing with the verified correction on the treatment axes it concerns, $p_\theta(s_j) = \sum_{a \in S_j} \pi_\theta(a \mid s_j)$. Moving all of the mass onto S_j is a displacement of unit length along this coordinate, so the constraint of Eq. (3) becomes

$$\psi_j(\theta) = m_j - p_\theta(s_j) \leq 0,$$

with $m_j = \eta$ while a correction is being learned. It is one-sided, since a probability above m_j incurs no penalty. The sepsis simulator and the Guarded Maze use $\eta = 0.5$. The retention forks of Section 5.4 and Appendix F.2.1 set m_j to the probability reached at the fork, so that a learned correction is held at its level, and the maze forks set $m_j = 0.9$ for corrections formed after the fork. The shared three-axis action space of replay baselines stores all three axes, which is why the single-axis rehearsal arm (`rehearsal_1ax`) was added as a matched control.

In the execution task, the actor is an MLP over per-step book features plus the regime observation; two Gaussian heads decode (b, p) through the bounded residual transform. The scaled dual coefficient λ for the mean budget is updated from the constraint slack (one λ per regime bucket). The regime-conditioned budget enforces the mean constraint per m_1-bucket on non-crash buckets only. The admission layer is a gradient-boosted tree over 33 pre-open features with labels from counterfactual menu replays; rejected stock-days execute pure POV from step 0, and the evaluation uses exactly this protocol.

H.2 Evidence Formation and Authority

A fixed loss floor selects consequential episodes for the verification of Section 3.1 (severity decides *what gets inspected*, never *how strongly* a lesson acts). Within a selected episode, a proposal critic nominates the top-K_{cand} decision steps by improvability

$$\text{imp}(t) = \max_i \left| Q_\lambda(s_t, a_t + \delta_i) - Q_\lambda(s_t, a_t) \right|$$

over a small probe grid $\{\delta_i\}$; the critic only prioritizes expensive counterfactual queries and never decides whether a correction is accepted. The K probe actions $\{a_i\}$ are local perturbations around the executed action a_0 (per-axis $\pm\delta$ in the continuous-action environments; action-side probes along the corrected axis in the discrete Sepsis environment). If $J(a;\omega) = E(\omega) + f(a;\omega)$ with action-independent $E(\omega)$, same-realization differencing cancels $E(\omega)$ exactly:

$$J(a';\omega) - J(a;\omega) = f(a';\omega) - f(a;\omega).$$

From the same-realization gains $G_i = \Delta_t(a_i; a_0, \omega)$ (Eq. (1)), the thresholded gains h_i of Eq. (2) yield the *verified headroom* $H = \max_i h_i$, the *directional clarity*, and the single-realization evidence strength,

$$C_{\text{dir}} = \Big\| \sum_i h_i \hat{v}_i \Big/ \Big(\sum_i h_i + \epsilon_{\text{num}} \Big) \Big\|, \qquad P = \big(1 - e^{-H/s_P}\big)\, C_{\text{dir}} \in [0, 1),$$

where $\hat{v}_i$ is the unit direction of probe i. P is an ordinal score with scale knob s_P, not a calibrated probability. Events with negligible strength are discarded. For entry j, let $\bar{P}_j$ be the mean evidence strength over its n_j distinct supporting realizations. Its consolidated strength is

$$P_j = \bar{P}_j \big(1 - e^{-n_j / n_{\text{scale}}}\big),$$

where $n_{\text{scale}} > 0$ controls saturation. Consolidation follows three anti-ratchet rules: re-observing the same realization *replaces* its evidence rather than stacking it; only *distinct* realizations increase n_j in the consolidated strength; and opposing evidence on a similar state *demotes* the lesson, so a direction that flips across realizations never consolidates. The hard admission floor enforces $w_j^{\text{auth}} = 0$ for $n_j < n_{\min}$; such pending evidence contributes no direct teaching or veto term, although its persistence weight remains in the normalization denominators. The floor $n_{\min}$ and saturation n_{scale} are design parameters inspired by the sign-consistency calculation below, not calibrated significance levels.

Idealized recurrence calculation. Fix a correctly matched state prototype. Let $X_e \in \{0,1\}$ indicate agreement of the verified correction on distinct realization e with a candidate direction, conditionally independent across realizations. For a stable causal lesson suppose $\Pr(X_e = 1) \geq \frac{1}{2} + \kappa$; for a hindsight-only lesson suppose directions are sign-symmetric (so $\Pr(X_e = 1) \leq \frac{1}{2}$). If a direction consolidates only when $m^{-1} \sum_e X_e \geq \frac{1}{2} + \frac{\kappa}{2}$, then

$$\Pr(\text{false consolidation}) \leq e^{-m\kappa^2/2}, \qquad \Pr(\text{miss a stable lesson}) \leq e^{-m\kappa^2/2}.$$

Both statements are Hoeffding bounds on the empirical mean around its expectation. For the stable lesson and for the sign-symmetric lesson,

$$\Pr\left(m^{-1}\sum X_e < \tfrac{1}{2} + \tfrac{\kappa}{2}\right) \leq \Pr\left(m^{-1}\sum (X_e - \mathbb{E}X_e) < -\tfrac{\kappa}{2}\right) \leq e^{-2m(\kappa/2)^2} = e^{-m\kappa^2/2},$$
$$\Pr\left(m^{-1}\sum X_e \geq \tfrac{1}{2} + \tfrac{\kappa}{2}\right) \leq e^{-2m(\kappa/2)^2}.$$

The implemented soft gate (Section 3.1) and admission floor $n_{\min}$ are design parameters *inspired* by this analysis, not calibrated significance levels: the candidate direction is selected from the same realization's best probe, so the sign-symmetric null above does not literally hold, and multiple testing across candidate steps is not corrected. Where the text reports precision, it reports empirical precision against environment ground truth, which needs no null model.

Evidence is grouped into entries by a state key that each arena defines. The corridor bins its observation into the segment type, two-step time buckets, and average-speed buckets of width 0.25. The sepsis simulator matches exactly on the 144-state projection that the policy observes, and the Guarded Maze matches on the agent's grid cell. The execution runs project the standardized observation onto a one-dimensional score. The score starts from a linear gate score fitted earlier on the same observations and is refit by ridge regression, separating stored events from background states, once memory holds 50 events. An event joins an entry when the Gaussian kernel of their score distance, with bandwidth 0.25 in standardized units, is at least 0.7.

The arenas instantiate the formation steps differently. The table lists what each arena's frozen code does; a dash marks a mechanism the arena does not implement.

Step	Exec.	Corridor	Sepsis	Maze
Episodes inspected	replayed IS below -60 bp on a fresh panel of 2,000 training stock-days, every 10 updates	worst 10% of the batch	every death among 12,000 proposal tapes	worst 5% of the batch
Steps probed	top 3 by critic improvability within steps 4–14	every step	4 at random	4 at random
Probes	± 0.1 per axis on (b, p); at a bound only the feasible side	± 0.2 on the speed v	all 7 alternative actions; the flagged axis re-verified on up to 1,500 fresh tapes	all 3 alternative actions
Continuation	probe action held to the end of the episode; baseline follows the frozen mean policy	exact; later actions as executed	frozen warm-start policy, taped	taped state, common random numbers, current policy
Admission	≥ 2 tapes, $n_{\text{scale}} = 2$, conflict rate ≤ 0.3	≥ 3 tapes, direction agreement ≥ 0.7, confidence $\min(1, n/6)$	≥ 40 tapes, lower bound > 0 at $z = 1.645$, reliability $1 - e^{-n/40}$	≥ 2 tapes, purity ≥ 0.6, weight purity$\cdot(1 - e^{-n/2})$
Teaching	one-sided retention on the latent z, $\eta = 0.5$	step of $0.5\times$ the verified displacement, at most 0.2	one-sided margin 0.5 on the probability of the verified side	one-sided margin 0.5 on the probability of the verified action; 0.9 for corrections formed after a retention fork
Update	teaching uncapped, reliability-weighted learning rate	penalty added to the PPO loss, weight set by the budget throttle	Eq. (4), cap $0.05 R_{\text{P}} \|g_{\text{A}}\|$	Eq. (4), cap $0.05 R_{\text{P}} \|g_{\text{A}}\|$; $R_{\text{P}} \|g_{\text{A}}\|$ in the retention forks
Veto	on, $q = R_{\text{P}} F_{\text{P}}$	— (the corridor tests admission only)	on, $q = R_{\text{P}} F_{\text{P}}$; off in the retention fork, whose corrections are already learned	on, $q = R_{\text{P}} F_{\text{P}}$
Persistence	$\gamma_j \in [0.9, 0.99]$; capacity 8,000 (at most 2,347 used)	— (lessons re-formed from each batch)	fixed after one build	no decay; at most 64 states per (cell, action)
Mean budget	dual per regime bucket, $\tau_\lambda = 1$	multiplicative throttle	—	— (capped-return objective)
What it tests	the whole system	admission; forgetting is weak here, a median 88% of the taught shift remaining 60 updates after teaching stops	admission against exact counterfactuals; retention under a shift	persistence and revision of a live memory

Two implementation notes matter for reading the results. The first concerns probes at action bounds. By default an axis contributes to the corrective target and the clarity score only when both $+\delta$ and $-\delta$ probes are feasible, which prevents mechanically one-sided "corrections" at action boundaries. The execution runs relax this rule: at a bound only the feasible side is probed, and that one-sided probe still contributes. With $\delta = 0.1$ and $b \in [0, 0.2]$, the b axis is then almost always probed from one side.

The second concerns probe-set centering. Subtracting the positive median removes probe-set-wide improvement that does not identify a specific corrective direction. This conservative centering can attenuate events for which a majority of probes improve, by design; it is distinct from cancellation of additive exogenous variation in the paired replay. For a probe set containing exactly two probes, the median is their mean; when both gains are positive and one is larger, only the better probe can retain positive excess gain. This two-probe statement does not apply to the median of a larger, multi-axis probe set.

For the revisable authority of Section 3.1, each lesson decays with retention factor γ_j and has storage weight $w_j^{\text{persist}}(u)$,

$$\gamma_j = \min\big(\gamma_{\max},\, \gamma_{\min} + (1-\gamma_{\min})P_j\big), \quad \gamma_{\max} < 1, \qquad w_j^{\text{persist}}(u) = P_j \gamma_j^{\,u-u_j},$$

where u_j is the last re-verification update. Authority discounts each evidence entry by the log-distance between the scaled dual coefficient at formation and now,

$$r_\lambda = \exp\Big(-\tfrac{|\log(1+\lambda_{\text{now}}) - \log(1+\lambda_{\text{form}})|}{\tau_\lambda}\Big), \qquad n_j^{\text{eff}} = \sum_{e \in \mathcal{E}_j} r_{\lambda,e},$$

$$w_j^{\text{auth}} = \mathbf{1}\{n_j \ge n_{\min}\}\, \gamma_j^{\,u-u_j} \Big(\tfrac{1}{n_j} \sum_e P_e\, r_{\lambda,e}\Big)\big(1 - e^{-n_j^{\text{eff}}/n_{\text{scale}}}\big).$$

Discounting the effective count n^{eff} prevents many stale entries from buying confidence by quantity; storage weights are context-free, so stale lessons lose authority but are not deleted. Normalizing L_{P} by persistence rather than authority lets context-stale entries lose influence without being renormalized to full strength.

H.3 Update, Arbitration, and Replay

The capped update in Section 3.3 is a particular choice of gradient scaling. More generally, retain the same conflict-aware projection and use

$$g_u = \xi_u g'_{\text{A}} + \tau_u g_{\text{P}}, \qquad \xi_u \in [0,1], \quad \tau_u \ge 0. \tag{18}$$

The coefficients are held fixed within each update. Equation (4) corresponds to $\xi_u = 1$ and $\tau_u = \min\{\alpha_{\text{P}}, \rho_{\text{P}}\|g_{\text{A}}\|/\|g_{\text{P}}\|\}$ when $g_{\text{P}} \neq 0$, with $\tau_u = 0$ otherwise. Optimizer-side step-size control, including KL-based control, is separate from these gradient weights. The same algebra gives

$$g_u = \nabla_\theta\big(\xi_u L_{\text{PPO}} + \widetilde{\beta}_u L_{\text{P}}\big)\big|_{\theta_u}, \qquad \widetilde{\beta}_u = \tau_u + \frac{\xi_u q_u [-\langle g_{\text{A}}, g_{\text{P}}\rangle]_+}{\|g_{\text{P}}\|^2 + \epsilon_{\text{num}}} \ge 0,$$

with all coefficients held fixed. Proposition 1 states the capped case and its norm bound; the separation results also use that instantiation.

For the arbitration parameters, let

$$v_j = [\psi_j(\theta)]_+, \qquad f_j = \text{clip}\,\big(v_j/(\eta\|d_j\|), 0, 1\big).$$

The veto strength is $q = R_{\text{P}} F_{\text{P}}$, with authority-weighted violation fraction F_{P} and memory reliability R_{P} defined by

$$F_{\text{P}} = \frac{\sum_j w_j^{\text{auth}} f_j}{\sum_j w_j^{\text{auth}}},$$

$$R_{\text{P}} = \frac{\sum_j w_j^{\text{auth}} (1 - e^{-n_j^{\text{eff}}/n_{\text{scale}}})}{\sum_j w_j^{\text{persist}} + \epsilon_{\text{num}}}.$$

If no entry has positive authority, set $F_{\mathrm{P}} = q = 0$. Since $0 < r_{\lambda,e} \le 1$, we have

$$n_j^{\mathrm{eff}} \le n_j, \qquad \frac{1}{n_j} \sum_e P_e \, r_{\lambda,e} \le \bar{P}_j,$$

hence $0 \le w_j^{\mathrm{auth}} \le w_j^{\mathrm{persist}}$. Therefore $0 \le R_{\mathrm{P}} < 1$ for $\epsilon_{\mathrm{num}} > 0$; together with $0 \le F_{\mathrm{P}} \le 1$, this gives $q = R_{\mathrm{P}} F_{\mathrm{P}} \in [0, 1)$. In the execution runs the teaching term enters the actor step without the $\rho_{\mathrm{P}} \|g_{\mathrm{A}}\|$ cap of Eq. (4); its step size is set instead by a reliability-weighted learning rate under the same KL control. The Guarded Maze and the sepsis simulator use Eq. (4) with $\xi_u = 1$ and $\alpha_{\mathrm{P}} = 1000$, and they multiply its cap by the memory reliability, so that $\|g_{\mathrm{P}}^{\mathrm{corr}}\| \le \rho_{\mathrm{P}} R_{\mathrm{P}} \|g_{\mathrm{A}}\|$. The cap is $\rho_{\mathrm{P}} = 0.05$ in the sepsis simulator and in Guarded Maze training, and $\rho_{\mathrm{P}} = 1$ in the Guarded Maze retention forks. Where the veto is on, its strength is $q = R_{\mathrm{P}} F_{\mathrm{P}}$ with the dimensionless forms defined above. The corridor has no gradient arbitration and no veto. It adds a squared-error loss toward the bounded action targets of the table above to the PPO loss, with a weight that its mean-budget throttle scales. The aggregated gradient g_u is handed to the optimizer; the deployment objective (CVaR subject to the mean budget) is never replaced by $L_{\mathrm{PPO}} + \beta L_{\mathrm{P}}$.

Memory refresh happens every U updates, strictly *after* the actor step (Algorithm 1). The verifier freezes the actor snapshot from the event's episode and reuses all recorded randomness. Execution continuations use deterministic mean actions, except that each probe branch holds the probe action from step t to the end of the episode, so the measured gain is the value of holding the alternative rather than of a one-step change (the other arenas change a single step and continue with the rule in the table above); stochastic continuations reuse base sampling variables from the tape at their regenerated states, as specified in Section 2.2. With $\mathcal{M}_0 = \varnothing$, the early updates with $\xi_u = 1$ are gradient-identical to the backbone; a self-test asserts this to machine precision (max $|\Delta g_{\mathrm{A}}| = 0.0$).

Both $J(a; \omega, t)$ and $J(a_0; \omega, t)$ use the same recorded randomness. The replayed baseline need not equal a logged return generated by a different continuation rule. Post-intervention states are regenerated; stochastic continuations reuse the recorded base sampling variables at those states. Replay is exact for the specified continuation when the dynamics and relevant randomness are fully captured; an approximate replay environment instead yields a replay-model counterfactual. Historical replay additionally requires the small-agent assumption in Appendix F.4.

Appendix C.2 specifies a binary comparator with a factual control, a zero logit margin, one frozen constraint, and bounded-gap re-verification of both supporting records. These choices define the instance in Proposition 2; they are not additional claims about the experimental configurations above.