FROM ADVANTAGE PERTURBATIONS TO COUNTERFACTUAL PPO GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Proximal Policy Optimization (PPO) relies on advantage estimates to reinforce useful actions, but in sparse-reward tasks these estimates can be delayed and noisy, making it difficult to identify which earlier actions led to success. An external oracle, such as a planner, expert, safety checker, or simulator, can provide action-level information not available from the environment reward. We study advantage perturbations as an interface for adding oracle information directly to PPO's policy update, without changing the environment reward or executed actions. We show that state-only perturbations do not shift the expected local policy gradient at the rollout policy, whereas action-relative perturbations can change preferences between actions at the same state. Controlled CartPole experiments confirm this mechanism and show that reversing the perturbation reverses the learned preference for a corrective action. We then use task-informed manual oracles to guide PPO on four sparse-reward MiniGrid tasks. Since manually specifying preferred actions requires task-specific knowledge, we next introduce a more general oracle based on local counterfactual action evaluation, which compares the outcomes of alternative actions from selected rollout states. Beyond adding this counterfactual signal at the advantage level, we also test it in the TD update and directly in the policy-loss objective. These interventions show task-dependent effects relative to standard PPO, with improvements in several task–intervention combinations.

1 INTRODUCTION

PPO improves a policy by reinforcing sampled actions with a positive estimated advantage. In sparse-reward, long-horizon tasks, however, an agent may make many decisions before a terminal outcome reveals whether the trajectory succeeded. PPO must then infer which earlier actions contributed to that outcome using sampled advantage estimates and a learned critic. As a result, the feedback can be weak, delayed, and noisy even after the agent reaches a rewarding outcome. (Williams, 1992; Schulman et al., 2015; 2017; Arjona-Medina et al., 2019).

An external oracle can provide action-level information that is unavailable from the environment reward. A planner, safety checker, human expert, or simulator can evaluate the actions available at a state: for example, a shortest-path planner can rank navigation actions by progress toward a goal, while a safety oracle can label state–action pairs as safe or unsafe (Saunders et al., 2017; Bennett et al., 2023). In sparse-reward tasks, a simulator or task planner can compare likely action outcomes before a terminal reward is observed (Sutton, 1991; Kocsis & Szepesvári, 2006). Prior work usually uses such guidance for imitation learning from planner-generated trajectories (Levine & Koltun, 2013), to modify executed actions or restrict exploration (Johannink et al., 2019; Bennett et al., 2023; Saunders et al., 2017), or as an additional shaped reward (Ng et al., 1999; Xiao et al., 2019). We instead ask whether PPO can use oracle information directly in its policy update without changing the environment reward or executed actions.

We incorporate oracle information by perturbing PPO's advantage estimate, which weights the policy update: positive advantages reinforce sampled actions, while negative advantages discourage them (Schulman et al., 2017). Adding an oracle score to the sampled advantage strengthens or weakens this update according to its sign, changing the actor update without modifying the environment reward, collected trajectory, or critic target. To test whether state-only or action-dependent information is needed for steering, we conduct controlled CartPole experiments with both types of

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.

perturbation. State-only perturbations do not systematically steer PPO because they act as baselines, whereas action-dependent perturbations alter learning. This differs from baselines and control variates, which reduce policy-gradient variance while preserving the environment-return objective (Thomas & Brunskill, 2017; Tucker et al., 2018; Gu et al., 2016). Our action-relative signal instead deliberately changes the actor update; reversing its direction reverses the learned preference for a locally correct action, showing that both action dependence and direction determine how oracle guidance steers PPO.

We next test task-informed oracle perturbations beyond the controlled setting. On four sparse-reward MiniGrid tasks, RedBlueDoors 6x6, KeyCorridor S3R3, DoorKey 6x6, and UnlockPickup (Chevalier-Boisvert et al., 2023), we construct manual oracles using privileged knowledge of the environment structure. Each oracle ranks available actions by their expected progress toward task completion. Our results show that these task-aligned, action-dependent signals can help PPO learn faster before sparse terminal rewards provide direct feedback.

Manual oracles require task-specific knowledge of which actions to prefer at each state. We therefore construct a more general oracle based on local counterfactual (CF) action evaluation, related to methods for multi-agent credit assignment, hindsight credit assignment, and counterfactually guided policy search (Foerster et al., 2018; Harutyunyan et al., 2019; Mesnard et al., 2020; Buesing et al., 2018). The CF oracle compares the predicted outcomes of alternative actions at the same rollout state and uses their relative values as an action-relative signal for PPO. Using simulated action outcomes is also common in model-based methods based on imagined trajectories (Sutton, 1991; Chua et al., 2018; Feinberg et al., 2018; Janner et al., 2019). Here, however, these outcomes construct a guidance signal rather than replace the environment's experience or train the critic on imagined returns.

We also test CF perturbations at two additional levels in PPO. TD-level guidance adds the signal to the TD error, changing the advantages, return targets, and critic learning, whereas loss-level guidance adds it directly to the policy objective across available actions. Comparing these settings tests how CF guidance behaves when introduced at different points in PPO. Across the sparse-reward MiniGrid tasks, CF-based guidance can enable faster learning than standard PPO, with outcomes varying across tasks and intervention types.

We make the following contributions:

1. We introduce actor-level advantage perturbations as an interface for incorporating external guidance into PPO without changing the environment reward or critic targets, and subsequently study related TD- and loss-level interventions.

2. We show that only the action-dependent component of a perturbation can systematically shift the expected policy gradient. Controlled experiments confirm this result and show that the perturbation direction determines the direction of policy steering.

3. We study task-informed guidance for sparse reward settings using manually constructed oracles and local counterfactual action evaluation, which compares the outcomes of alternative actions at the same visited state.

2 PRELIMINARIES

Consider a finite-action, finite-horizon discounted episodic Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \mu_0, T_{\max})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the finite action space, P denotes transition kernel, r is the environment reward function, $\gamma \in [0, 1)$ is the discount factor, μ_0 is the initial-state distribution, and $T_{\max}$ is the maximum episode length. At time t, the agent observes state s_t, samples $a_t \sim \pi_\theta(\cdot \mid s_t)$, receives reward r_t, and transitions to s_{t+1}. We use a differentiable stochastic policy $\pi_\theta(a_t \mid s_t)$, parameterized by θ. Let $Q^{\pi_\theta}(s_t, a_t)$ and $V^{\pi_\theta}(s_t)$ denote the action-value and state-value functions at time step t, respectively. For notational simplicity, we suppress the time index when referring to quantities associated with a single state–action pair, writing s and a in place of s_t and a_t. We retain time indices when needed to distinguish quantities across different steps of a trajectory. Unless otherwise specified, quantities such as Q^{π_θ} and V^{π_θ} denote their true values, while their estimated counterparts are denoted with a hat, e.g., $\widehat{Q}^{\pi_\theta}$ and $\widehat{V}^{\pi_\theta}$.

3 ACTION-RELATIVE ADVANTAGE PERTURBATIONS IN PPO

We analyze PPO at the fixed rollout policy $\pi_0 \equiv \pi_{\theta_0}$, with normalized discounted state-visitation distribution d^{π_0}. Its advantage function is $A^{\pi_0}(s,a) = Q^{\pi_0}(s,a) - V^{\pi_0}(s)$. We define the policy score at the rollout parameters as $z_{\theta_0}(s,a) = \nabla_\theta \log \pi_\theta(a \mid s)|_{\theta=\theta_0}$. The policy gradient for maximizing the expected discounted return is

$$g_0 \coloneqq \mathbb{E}_{\substack{s\sim d^{\pi_0} \\ a\sim\pi_0(\cdot|s)}} \left[A^{\pi_0}(s,a)\, z_{\theta_0}(s,a)\right]. \tag{1}$$

3.1 ACTION-RELATIVE EXTERNAL PERTURBATIONS

Let $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be an external perturbation function added to the advantage:

$$A^{\mathrm{pert}}(s,a) = A^{\pi_0}(s,a) + \beta f(s,a), \tag{2}$$

where $\beta \geq 0$ controls the perturbation strength. At each state, f can be decomposed into a state-only component $c : \mathcal{S} \to \mathbb{R}$ and an action-relative component $\tilde{f} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$:

$$f(s,a) = \underbrace{\tilde{f}(s,a)}_{\text{action-relative part}} + \underbrace{c(s)}_{\text{state-only part}}, \quad \text{where} \quad c(s) = \mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[f(s,a')\right], \tag{3}$$

Proposition 1 (Policy-gradient shift induced by action-relative perturbations)**.** *Let g_β denote the local actor gradient obtained by replacing A^{π_0} with A^{pert}; then*

$$g_\beta = g_0 + \beta \mathbb{E}_{\substack{s\sim d^{\pi_0} \\ a\sim\pi_0(\cdot|s)}} \left[\tilde{f}(s,a) z_{\theta_0}(s,a)\right]. \tag{4}$$

Proposition 1 shows that the policy-gradient shift depends only on the action-relative component $\tilde{f}(s,a)$. The state-only component acts as a baseline and is canceled from the expected gradient. In contrast, the action-relative perturbation deliberately shifts the expected actor gradient to steer PPO. A detailed proof is provided in Appendix B.

3.2 GRADIENT ALIGNMENT

Let $\Delta g_f \coloneqq g_\beta - g_0$ denote the gradient shift induced by the action-relative perturbation. We use the inner product $\langle g_0, \Delta g_f \rangle$ to measure its alignment with the original policy gradient. For $\beta > 0$, a positive inner product means that the perturbation reinforces the original update, whereas a negative value means that it opposes the update and can redirect learning away from maximizing the expected discounted return. If $\langle g_0, \Delta g_f \rangle = 0$, the perturbation shifts the actor gradient in the direction orthogonal to g_0. Thus, the effect of an action-relative perturbation depends on its direction, not only its magnitude. It can reinforce, oppose, or redirect the PPO update.

3.3 CONTROLLED CARTPOLE EXPERIMENTS

Proposition 1 shows that only action-relative preferences can shift the local policy gradient. We test this prediction in controlled CartPole experiments, where the two discrete actions make perturbation-induced changes in action preference easy to specify and measure. We test whether state-only perturbations produce no systematic change in PPO learning, whether the effect increases with the action-relative component, and whether reversing an action-relative perturbation reverses the learned preference for a locally correct action. All conditions use the same PPO implementation, environment configuration, training budget, and evaluation procedure; the perturbation is the only intended difference. Appendix C provides the full experimental protocol.

To isolate the effect of action-relative information, we perturb the sampled PPO advantage using:

$$\begin{aligned} f(s,a) &= h(s)\left[(1-\chi) + \chi\, u(a \mid s)\right], \\ A^{\mathrm{pert}}(s,a) &= \widehat{A}(s,a) + \beta f(s,a). \end{aligned} \tag{5}$$

Here, $\widehat{A}(s,a)$ is the sampled PPO advantage, $\beta \geq 0$ controls the perturbation strength, $h(s)$ is a fixed state-dependent function, and $u(a \mid s)$ is an action-dependent function. The parameter $\chi \in$

$[0, 1]$ controls the action-relative fraction of the perturbation: $\chi = 0$ gives a state-only perturbation, whereas $\chi = 1$ gives a fully action-dependent perturbation. Within each PPO minibatch, we rescale $f(s, a)$ so that its standard deviation matches that of the sampled advantage.

We use a random action-relative perturbation whose arbitrary preferences can conflict with reward-supported behavior and slow PPO learning. We measure its effect by the PPO updates required to reach an episodic-return threshold of 300. Let τ_{pert} and τ_{PPO} denote the mean threshold-crossing times across seeds for a perturbation condition and standard PPO, respectively. We report the standardized threshold-time difference $\Delta_\tau = (\tau_{\text{pert}} - \tau_{\text{PPO}})/\,\text{sd}(\tau_{\text{PPO}})$. Here, $\text{sd}(\tau_{\text{PPO}})$ is the standard deviation of PPO threshold-crossing times across seeds. A value of $\Delta_\tau = 0$ matches PPO; positive and negative values indicate slower and faster learning, respectively.We use five seeds per condition; runs that do not reach the threshold within the training budget are assigned the first update beyond the training horizon. We first compare $\chi = 0$ and $\chi = 1$ across increasing β.

Figure 1A shows that the state-dependent perturbation ($\chi = 0$) remains close to standard PPO across all tested strengths. In contrast, the action-dependent perturbation ($\chi = 1$) produces larger learning delays as β increases, exceeding five PPO seed-to-seed standard deviations at the largest strength. Thus, disruption arises from arbitrary relative action preferences, not perturbation magnitude alone. We next fix $\beta \in \{0.05, 0.10\}$ and vary the action-relative fraction $\chi \in \{0, 0.25, 0.5, 1\}$. Figure 1B shows that relative threshold time increases with χ for both strengths, with a larger effect at $\beta = 0.10$. Thus, arbitrary action-relative preferences, rather than state-only offsets, systematically steer PPO and can disrupt learning.

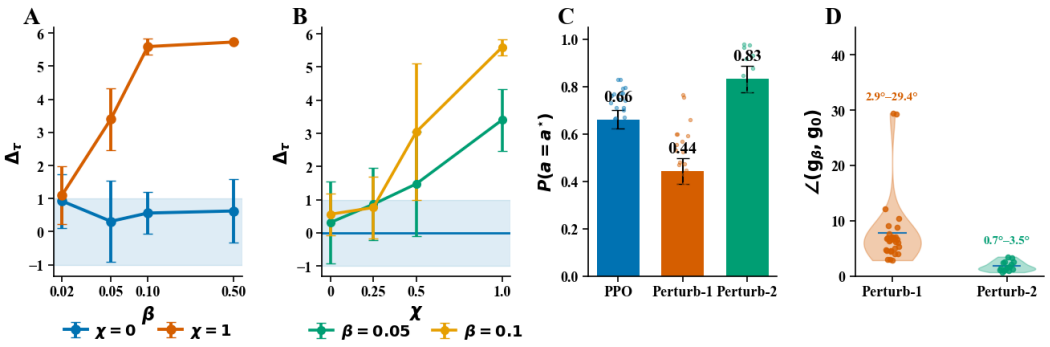

Figure 1: Controlled CartPole experiments. **(A)** Standardized threshold-time difference (Δ_τ) as perturbation strength β increases. **(B)** Δ_τ as the action-relative fraction χ increases. **(C)** Probability of selecting the targeted corrective action a^* in the trigger region after training. Perturb-1 suppresses this action, whereas Perturb-2 promotes it. **(D)**The gradient angle between the ordinary PPO policy gradient and the perturbation-modified policy gradient for Perturb-1 and Perturb-2.

We next test whether the direction of a targeted action-relative perturbation controls the learned preference for a selected action. We define the trigger region as $\mathcal{T} = \{s : 0.0743 \leq \theta_{\text{pole}}(s) \leq 0.2095\}$, where the pole is tilted right and $a^\star = 1$, which pushes the cart right, is the targeted corrective action. Within $\mathcal{T}$, we use $u(a \mid s) = \pm[\mathbb{I}(a = a^\star) - \pi_0(a^\star \mid s)]$. Perturb-1 uses the negative sign to reduce the relative advantage of $a^\star$, whereas Perturb-2 uses the positive sign to increase it. This signal is centered under π_0, so the conditions have the same scale and differ only in action preference direction. We apply it only within the trigger region, $f(s, a) = \mathbb{I}(s \in \mathcal{T})u(a \mid s)$, with $\beta = 0.2$ and $\chi = 1$ for both conditions. We train standard PPO, Perturb-1, and Perturb-2 with 15 random seeds and evaluate the probability of selecting $a^\star$ on held-out trigger-region states. Figure 1C shows that standard PPO selects the corrective action with probability 0.66; Perturb-1 reduces this to 0.44, whereas Perturb-2 increases it to 0.83. Figure 1D shows the corresponding later-training actor-gradient angles: Perturb-1 deviates by $2.9°$–$29.4°$, while Perturb-2 remains closer to the ordinary PPO update at $0.7°$–$3.5°$. Together, these results show that reversing the direction of an action-relative perturbation reverses its effect on learned action selection.

4 MANUAL ORACLE GUIDANCE FOR STEERING PPO IN SPARSE-REWARD MINIGRID

The CartPole results establish that action-relative perturbations can steer PPO. We next test whether task-informed guidance is useful in sparse-reward MiniGrid tasks: RedBlueDoors 6×6, KeyCorridor S3R3, DoorKey 6×6, and UnlockPickup. Each task requires several decisions before a sparse terminal reward reveals success, making it difficult for the standard PPO to identify which earlier actions should be reinforced. For each task, we construct a manual oracle $f^{\text{manual}}(s, a)$ that ranks available actions by their predicted progress toward task completion. The oracle reads the MiniGrid state and gives higher scores to actions that advance the current task step, and lower scores to actions that move away from completion or cause failure. For example, in DoorKey, it successively favors picking up the key, opening the door, and moving toward the goal. The task-specific design of f^{manual} for all the MiniGrid tasks is summarized in Table 1. Although the completion rules differ across tasks, each oracle provides the same type of signal: a relative ranking of the currently available actions.

Table 1: Task-specific manual oracle design.

Environment	Task structure	High-scoring actions
RedBlueDoors	Open red before blue	Move toward and open the red door first
KeyCorridor	Key $\rightarrow$ door $\rightarrow$ drop key $\rightarrow$ ball	Complete the next required subtask
DoorKey	Key $\rightarrow$ door $\rightarrow$ goal	Pick up key, unlock door, then reach goal
UnlockPickup	Key $\rightarrow$ door $\rightarrow$ drop key $\rightarrow$ box	Complete prerequisites and pick up the box

We center the manual-oracle scores under the rollout policy to retain only their relative ranking across actions, $\tilde{f}^{\text{manual}}(s, a) = f^{\text{manual}}(s, a) - \sum_{a' \in \mathcal{A}} \pi_0(a' \mid s) f^{\text{manual}}(s, a')$, and is used to perturb the advantage as follows:

$$\widehat{A}^{\text{manual}}(s, a) = \widehat{A}^{\text{Norm}}(s, a) + \frac{\beta}{\text{sd}(\widehat{A})} \tilde{f}^{\text{manual}}(s, a). \tag{6}$$

Here, $\widehat{A}(s, a)$ is the PPO advantage estimate computed using GAE (Generalized Advantage estimate) (Schulman et al., 2015), $\widehat{A}^{\text{Norm}}(s, a)$ is its normalized form, and $\text{sd}(\widehat{A})$ is the standard deviation of sampled GAE advantages in the current rollout batch. Actions ranked above the rollout-policy average receive additional reinforcement, whereas actions ranked below it receive reduced reinforcement.

Figure 2 compares standard PPO with manual-oracle PPO (PPO-manual) across the four sparse-reward MiniGrid tasks, using five seeds per method. Both methods use the same environment configuration and PPO hyperparameters; the manual advantage perturbation is the only intended difference. PPO-manual improves learning, showing that task-aligned, action-relative guidance can steer PPO toward successful behavior before sparse terminal rewards provide direct feedback.

5 COUNTERFACTUAL GUIDANCE FOR PPO

In this section, we replace the task-specific manual oracle in Section 4 with a more general counterfactual (CF) construction that does not require manually specified action preferences. Rather than using hand-designed rules, it evaluates alternative actions at a rollout state and uses their relative CF returns as guidance. In an idealized local population analysis with exact counterfactual action values, this signal preserves the policy-gradient direction while increasing its magnitude; we then characterize how counterfactual estimation error can affect this alignment. We test CF advantage perturbations in sparse-reward MiniGrid tasks and compare two alternative injection points: the TD residual and the policy-loss objective.

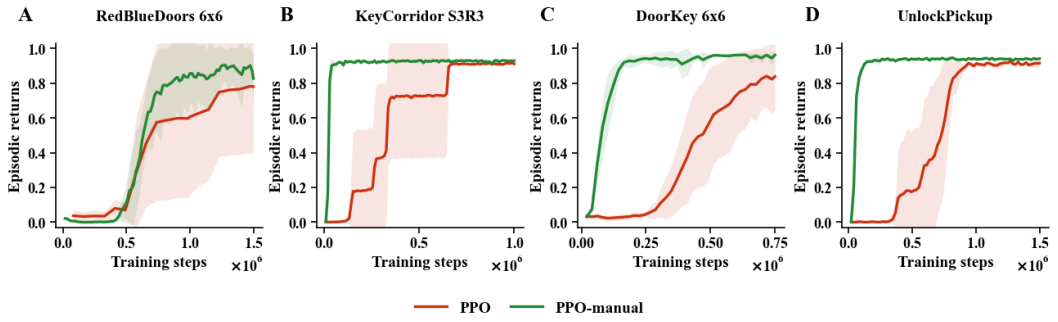

Figure 2: PPO-manual versus PPO on four sparse-reward MiniGrid tasks. PPO-manual adds a manual-oracle score to the actor advantage. Lines show mean episodic returns over five seeds; shaded regions show standard errors.

5.1 Counterfactual Advantage Perturbations and Local Alignment

We construct the CF perturbation by evaluating alternative actions at rollout states. For a state s and candidate action a, we restore the environment to s, force a, and then follow the rollout policy π_0. The resulting action-value estimate is $\widehat{Q}^{\pi_0,\mathrm{CF}}(s,a) := \mathbb{E}\left[\sum_{k=0}^{T_{\max}-t-1} \gamma^k r_{t+k} \,\middle|\, s_t = s,\ a_t = a\right]$.

Under exact CF evaluation, $\widehat{Q}^{\pi_0,\mathrm{CF}}(s,a) = Q^{\pi_0}(s,a)$. In practice, finite rollouts, finite-horizon truncation, and critic bootstrapping introduce counterfactual estimation error, including sampling variability and truncation or bootstrap error. We model this error as.

$$\widehat{Q}^{\pi_0,\mathrm{CF}}(s,a) = Q^{\pi_0}(s,a) + e(s,a), \qquad \|e\|_\infty \le \epsilon_{\mathrm{CF}}, \tag{7}$$

where $e(s,a)$ denotes CF estimation error.

We use the CF action-value estimate as the raw perturbation, $f^{\mathrm{CF}}(s,a) := \widehat{Q}^{\pi_0,\mathrm{CF}}(s,a)$. To retain only relative action preferences, we subtract its rollout-policy average. Let $\tilde{e}(s,a) := e(s,a) - \mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[e(s,a')\right]$ denote the corresponding action-relative CF error. Then,

$$\begin{aligned} \tilde{f}^{\mathrm{CF}}(s,a) &:= f^{\mathrm{CF}}(s,a) - \mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[f^{\mathrm{CF}}(s,a')\right] \\ &= A^{\pi_0}(s,a) + \tilde{e}(s,a). \end{aligned} \tag{8}$$

We perturb the estimated advantage according to

$$\widehat{A}^{\mathrm{pert}}(s,a) = \widehat{A}(s,a) + \beta \tilde{f}^{\mathrm{CF}}(s,a). \tag{9}$$

The following proposition characterizes the resulting local alignment.

Proposition 2 (CF local alignment)**.** *For $\beta \ge 0$, let $g_0 \ne 0$ and g_β denote the ordinary and CF-perturbed PPO actor gradients, respectively. Let ϵ_{CF} denote the maximum CF action-value estimation error across state-action pairs. The gradient distortion induced by the action-relative CF error is $g_e := \mathbb{E}_{\substack{s\sim d^{\pi_0}\\ a\sim\pi_0(\cdot|s)}}\left[\tilde{e}(s,a) z_{\theta_0}(s,a)\right]$. Then,*

$$g_\beta = (1+\beta)g_0 + \beta g_e, \qquad \|g_e\| \le 2\epsilon_{\mathrm{CF}} \mathbb{E}_{\substack{s\sim d^{\pi_0}\\ a\sim\pi_0(\cdot|s)}}\left[\|z_{\theta_0}(s,a)\|\right]. \tag{10}$$

Consequently,

$$2\beta\epsilon_{\mathrm{CF}} \mathbb{E}_{\substack{s\sim d^{\pi_0}\\ a\sim\pi_0(\cdot|s)}}\left[\|z_{\theta_0}(s,a)\|\right] < (1+\beta)\|g_0\| \quad \Longrightarrow \quad \langle g_0, g_\beta\rangle > 0. \tag{11}$$

The proposition gives a sufficient condition under which the CF-perturbed local population gradient remains positively aligned with the ordinary local population gradient at the rollout policy. In particular, when the CF estimation error is sufficiently small, $\langle g_0, g_\beta\rangle > 0$. Under exact CF evaluation, $g_e = 0$ and $g_\beta = (1+\beta)g_0$, so in this idealized local analysis the perturbation preserves the gradient direction and changes only its magnitude. A complete proof of Proposition 2 is provided in Appendix B.

5.2 COUNTERFACTUAL ADVANTAGE PERTURBATION FRAMEWORK IN SPARSE-REWARD MINIGRID

We apply CF advantage perturbations to the sparse-reward MiniGrid tasks. To construct the CF perturbation, we evaluate alternative actions at selected rollout states s. For each available action a, we restore the environment to s, force a, and then follow the frozen rollout policy π_0 for the remaining $H-1$ steps. Thus, H denotes the total counterfactual horizon including the forced action. We estimate the resulting H-step discounted return, bootstrapped with the critic value at the final simulated state when the rollout is nonterminal. Because individual CF trajectories can be noisy, we roll out multiple trajectories for each state-action pair and average their discounted returns to obtain the CF action-value score $f^{\mathrm{CF}}(s,a)$.

We obtain the action-relative CF signal by subtracting the rollout-policy average, $\tilde{f}^{\mathrm{CF}}(s,a) := f^{\mathrm{CF}}(s,a) - \mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[f^{\mathrm{CF}}(s,a')\right]$. A positive signal indicates that an action has a higher estimated CF return than the rollout-policy average at s, whereas a negative signal indicates a lower estimated return.

Because CF evaluation is more expensive than an ordinary PPO rollout, we apply it only at selected rollout states. Let $m(s) \in \{0,1\}$ indicate whether state s is selected for CF evaluation. When $m(s)=1$, we evaluate all available actions at s and use the resulting signal to perturb the advantage of the sampled action; when $m(s)=0$, PPO uses the unmodified advantage. The Advantage-CF estimate is

$$\widehat{A}^{\mathrm{CF}}(s,a) = \widehat{A}^{\mathrm{Norm}}(s,a) + \beta \frac{m(s)\tilde{f}^{\mathrm{CF}}(s,a)}{\mathrm{sd}(\widehat{A})}. \tag{12}$$

Here, β controls the CF perturbation strength, and $\mathrm{sd}(\widehat{A})$ is the standard deviation of the sampled GAE advantages in the current rollout batch. Thus, Advantage-CF changes only the actor advantage, leaving the environment reward, critic targets, and value-function objective unchanged. For uniform state subsampling, the expected counterfactual gradient contribution is scaled by the query fraction q; see Appendix B.

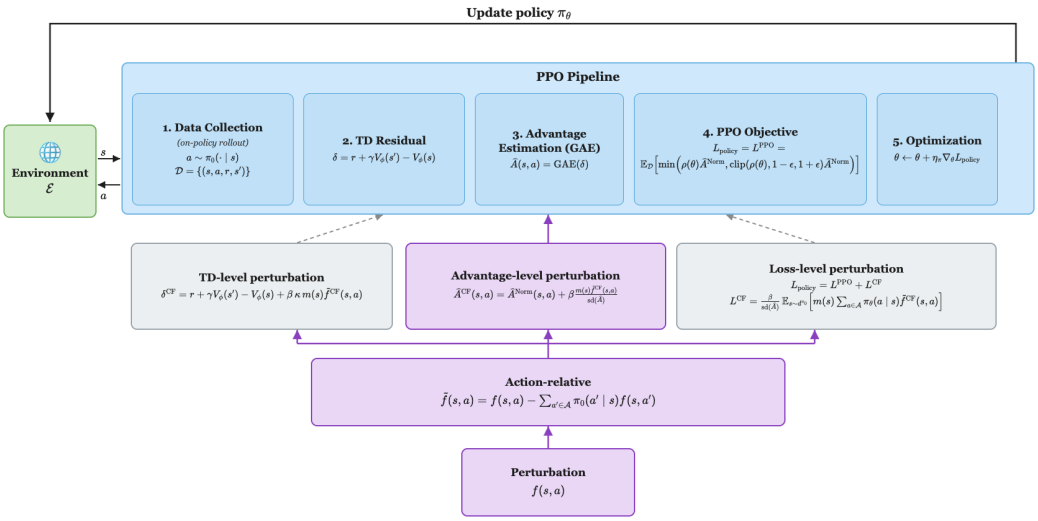

Figure 3: Overview of the PPO perturbation framework. An external signal is policy-centered to obtain an action-relative perturbation, which can be introduced at the TD, advantage, or policy-loss level of the PPO pipeline.

5.3 ALTERNATIVE COUNTERFACTUAL PERTURBATION LOCATIONS

We further study whether the same CF action preferences have different effects when introduced at other points in the PPO update. Advantage-CF adds the signal to the actor advantage, whereas TD-

CF and Loss-CF introduce the same signal at the TD residual and policy-loss objective, respectively. All methods use the same CF-based perturbation $\tilde{f}^{\mathrm{CF}}(s, a)$.

TD-CF: TD-CF introduces the perturbation before GAE by adding it to the reward used to construct the TD residual:

$$\delta^{\mathrm{CF}}(s, a, s') = r + \gamma V_\phi(s') - V_\phi(s) + \beta\, \kappa m(s) \tilde{f}^{\mathrm{CF}}(s, a) \tag{13}$$

Here, κ rescales the local CF signal to account for its propagation through the GAE trace. PPO then uses δ^{CF} to construct GAE estimates and critic targets. Thus, TD-CF propagates a local CF action comparison through the temporal trace, changing both the actor advantage and critic target.

Loss-CF: Loss-CF applies the same perturbation directly in the PPO policy objective. Let $\rho(\theta) := \pi_\theta(a \mid s)/\pi_0(a \mid s)$ denote the PPO likelihood ratio, and let $S_\epsilon(\rho(\theta), \widehat{A}^{\mathrm{Norm}})$ denote the clipped PPO surrogate contribution.

$$L^{\mathrm{CF}}(\theta) = \mathbb{E}_{\substack{s\sim d^{\pi_0} \\ a\sim\pi_0(\cdot|s)}} \left[S_\epsilon \left(\rho(\theta), \widehat{A}^{\mathrm{Norm}} \right) \right] \quad + \beta \frac{1}{\mathrm{sd}\left(\widehat{A}\right)} \mathbb{E}_{s\sim d^{\pi_0}} \left[m(s) \sum_{a' \in \mathcal{A}} \pi_\theta(a' \mid s) \tilde{f}^{\mathrm{CF}}(s, a') \right]. \tag{14}$$

Both expectations are estimated from the current PPO minibatch collected under π_0. The first term is the standard clipped PPO surrogate, while the second evaluates the centered CF perturbation over all actions at selected states, increasing the probability of positively scored actions and decreasing that of negatively scored actions. Unlike TD-CF, Loss-CF leaves the environment reward, GAE estimates, return targets, and critic objective unchanged. Unlike Advantage-CF, it evaluates all actions rather than only the sampled action.

Across TD-CF, Advantage-CF, and Loss-CF, the CF construction and state-sampling procedure are held fixed while the same action-relative guidance is introduced at different points in the PPO update. Guidance strength is calibrated separately across intervention types to account for differences in how the signal enters the update. Complete pseudocode for Advantage-CF, TD-CF, and Loss-CF is provided in Appendix A.

5.4 Sparse-Reward MiniGrid Results

We compare standard PPO with TD-CF, Advantage-CF, and Loss-CF on four sparse-reward MiniGrid tasks. Full training and algorithm-specific hyperparameters are provided in Appendix D.

Figure 4 shows that counterfactual perturbations can improve learning relative to PPO, but the effect depends on both the environment and the perturbation location. On RedBlueDoors 6x6, the counterfactual variants attain higher returns than PPO, although the magnitude of the improvement varies across interventions. On DoorKey 6x6, TD-CF and Advantage-CF both improve learning speed and final return. On UnlockPickup, the methods reach similar final returns, but TD-CF enters the high-return regime earlier. On KeyCorridor S3R3, the methods reach similar final returns, and the main difference is the time required to reach this regime.

Because the three interventions inject the same counterfactual signal at different points in the PPO update, the same numerical value of β does not in general induce the same effective actor-update magnitude. We use $\beta = 0.75$ for TD-CF and Advantage-CF and $\beta = 0.5$ for Loss-CF, while keeping the PPO optimizer, environment, counterfactual evaluation procedure, query fraction, and training budget fixed. The resulting comparisons are intended to assess whether counterfactual guidance is effective at each insertion point rather than to rank the interventions by their raw performance.

To compare learning speed, we measure the first training step at which each seed reaches a rolling success rate of 0.7. We report the standardized threshold-time difference $\Delta_\tau = (\tau_{\mathrm{pert}} - \tau_{\mathrm{PPO}})/\,\mathrm{sd}(\tau_{\mathrm{PPO}})$, where positive values indicate slower and negative values faster threshold attainment than PPO. Runs not reaching the threshold are assigned the final training step.

Table 2 summarizes learning speed relative to standard PPO for each CF intervention. TD-CF reaches the threshold faster than PPO on RedBlueDoors 6x6, DoorKey 6x6, and UnlockPickup,

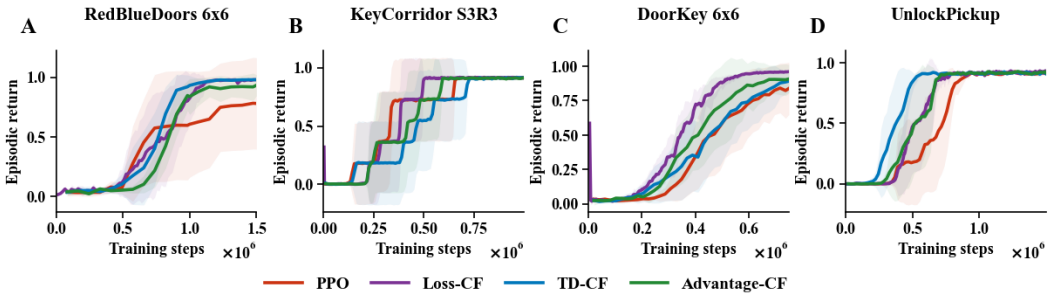

Figure 4: Episodic-return learning curves for PPO and counterfactual perturbation variants across four sparse-reward environments. Lines show mean episodic returns over five seeds, and shaded regions denote standard error.

but is slower on KeyCorridor S3R3. Advantage-CF is faster on DoorKey 6x6 and UnlockPickup, shows a negligible change on RedBlueDoors 6x6, and is slower on KeyCorridor S3R3. Loss-CF reaches the threshold earlier than PPO on all four tasks. Because Loss-CF uses a separately calibrated guidance strength ($\beta = 0.5$, compared with $\beta = 0.75$ for TD-CF and Advantage-CF), these results are intended to assess each intervention relative to standard PPO rather than to rank the three CF interventions against one another.

Table 2: Standardized time to reach a 0.7 success rate, relative to PPO. Negative values indicate faster threshold attainment. Runs not reaching the threshold are assigned the final training step; the best value in each row is bolded.

Environment	Loss-CF	TD-CF	Advantage-CF
RedBlueDoors 6x6	-0.29	$\mathbf{-0.30}$	-0.01
KeyCorridor S3R3	$\mathbf{-0.27}$	$+0.57$	$+0.25$
DoorKey 6x6	$\mathbf{-1.82}$	-0.22	-0.53
UnlockPickup	-0.81	$\mathbf{-1.65}$	-0.80

6 Discussion

In this paper we show that oracle-based advantage perturbations can steer PPO and improve learning in sparse-reward environments. Our analysis establishes that only the action-relative component of an oracle signal changes the expected actor gradient, and the controlled experiments show that reversing this component reverses the learned action preference. In the MiniGrid tasks, manual guidance improves learning, while counterfactual (CF) guidance produces task-dependent effects, including improvements in several settings. We also find that the effect depends on where the perturbation enters PPO: it can be applied to the actor advantage, propagated through the TD residual, or added directly to the policy-loss objective.

In practice, an oracle could be provided by a human, a planner, or a simulator. For example, a human safety expert or simulator could provide action preferences based on obstacle clearance, proximity to people or buildings, restricted-airspace constraints, or other auxiliary criteria. Such guidance could potentially influence policy learning before the primary task reward provides direct feedback. When a restorable simulator is available, a CF oracle can compare the predicted outcomes of alternative actions. These examples illustrate potential uses of the framework rather than capabilities demonstrated by our experiments.

However, a limitation of CF guidance is the cost of evaluating multiple actions and rollouts. Our implementation assumes restorable simulator states and a discrete action space; real systems may instead require a learned world model or another external evaluator. Evaluating only selected actions at informative states may reduce this cost in larger or continuous action spaces.

AI USE STATEMENT

In this work, we used generative AI tools, including ChatGPT, Gemini, and Claude, extensively during code development, plotting and visualization, analysis, ideation, proof development and verification, and manuscript preparation. These tools were used to assist with implementing and debugging experimental code, developing analysis and plotting scripts, exploring and refining research ideas, deriving and checking mathematical arguments and proofs, and improving the presentation and wording of the manuscript. All AI-assisted code, analyses, proofs, experimental results, and scientific claims were independently reviewed and verified by two authors. The authors made the final decisions regarding the methodology, mathematical arguments, experiments, interpretation of results, and content of the manuscript. We take full responsibility for the final content of this work, including text, proofs, claims, code, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work is methodological and evaluates reinforcement-learning algorithms exclusively in simulated environments. It does not involve human subjects, personal data, or real-world deployment. We do not identify additional ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide the complete implementation as supplementary material in a ZIP archive, together with a `README.md` containing instructions for running the experiments. Complete proofs of the theoretical results are provided in the appendix. The appendix also documents the implementation details and hyperparameters used in our experiments.

REFERENCES

Jose A Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems*, 32, 2019.

Andrew Bennett, Dipendra Misra, and Nathan Kallus. Provable safe reinforcement learning with binary feedback. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 10871–10900. PMLR, 25–27 Apr 2023. URL `https://proceedings.mlr.press/v206/bennett23a.html`.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. *arXiv preprint arXiv:1811.06272*, 2018.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Advances in Neural Information Processing Systems 36, New Orleans, LA, USA*, December 2023.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E Turner, and Sergey Levine. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, et al. Hindsight credit assignment. *Advances in neural information processing systems*, 32, 2019.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 international conference on robotics and automation (ICRA)*, pp. 6023–6029. IEEE, 2019.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pp. 282–293. Springer, 2006.

Sergey Levine and Vladlen Koltun. Guided policy search. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1–9, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/levine13.html.

Thomas Mesnard, Théophane Weber, Fabio Viola, Shantanu Thakoor, Alaa Saade, Anna Harutyunyan, Will Dabney, Tom Stepleton, Nicolas Heess, Arthur Guez, et al. Counterfactual credit assignment in model-free reinforcement learning. *arXiv preprint arXiv:2011.09464*, 2020.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.

William Saunders, Girish Sastry, Andreas Stuhlmueller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. *arXiv preprint arXiv:1707.05173*, 2017.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Philip S Thomas and Emma Brunskill. Policy gradient methods for reinforcement learning with function approximation and action-dependent baselines. *arXiv preprint arXiv:1706.06643*, 2017.

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. In *International conference on machine learning*, pp. 5015–5024. PMLR, 2018.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Baicen Xiao, Bhaskar Ramasubramanian, Andrew Clark, Hannaneh Hajishirzi, Linda Bushnell, and Radha Poovendran. Potential-based advice for stochastic policy learning. In *2019 IEEE 58th conference on decision and control (CDC)*, pp. 1842–1849. IEEE, 2019.

A ALGORITHMS

Algorithm 1 Advantage-CF: Counterfactual Advantage Perturbation for PPO

Input: PPO iterations I, query fraction q, counterfactual horizon H, counterfactual rollouts N, perturbation strength β
Input: Discount factor γ, GAE parameter λ, PPO clipping coefficient ϵ, update epochs K
Output: Policy parameters θ and critic parameters ϕ

Initialize policy π_θ, critic V_ϕ
for $i = 1, \dots, I$ **do**
  $\pi_0 \leftarrow \pi_\theta$ ▷ Freeze rollout policy
  collect trajectories under π_0, $\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}_{t=0}^{T-1}$
  $\mathcal{Q} \sim \mathrm{UniformSubset}\Big(\{(t, s_t)\}_{t=0}^{T-1}, \max\{1, \mathrm{round}(qT)\}\Big)$ ▷ Sample q fraction of rollout states
  $m(s_t) \leftarrow \mathbb{I}[(t, s_t) \in \mathcal{Q}], \quad t = 0, \dots, T-1$ ▷ $m(s_t) = 1$ for sampled states and $m(s_t) = 0$ otherwise
  for all $(t, s_t) \in \mathcal{Q}$ **do**
    for all $a \in \mathcal{A}$ **do**
      for $n = 1, \dots, N$ **do**
        Restore the environment state at s_t
        Force action a, then follow π_0 for the remaining $H-1$ steps
        Compute returns: $G_{t,n}^{\mathrm{CF}}(s_t, a) \leftarrow \sum_{\ell=0}^{H-1} \gamma^\ell r_{t+\ell}^{\mathrm{CF}} + \gamma^H \big(1 - d_{t+H-1}^{\mathrm{CF}}\big) V_\phi\Big(s_{t+H}^{\mathrm{CF}}\Big)$
      end for
      $f^{\mathrm{CF}}(s_t, a) \leftarrow \frac{1}{N} \sum_{n=1}^{N} G_{t,n}^{\mathrm{CF}}(s_t, a)$ ▷ Mean counterfactual returns
    end for
    for all $a \in \mathcal{A}$ **do**
      $\tilde{f}^{\mathrm{CF}}(s_t, a) \leftarrow f^{\mathrm{CF}}(s_t, a) - \sum_{a' \in \mathcal{A}} \pi_0(a' \mid s_t) f^{\mathrm{CF}}(s_t, a')$ ▷ Obtain perturbation signal
    end for
  end for

  $\delta_t \leftarrow r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$
  $\widehat{A}_t \leftarrow \sum_{\ell=0}^{T-t-1} (\gamma\lambda)^\ell \delta_{t+\ell}$
  $\widehat{R}_t \leftarrow \widehat{A}_t + V_\phi(s_t)$
  $A_t^{\mathrm{CF}} \leftarrow \widehat{A}_t^{\mathrm{Norm}} + \beta \frac{1}{\mathrm{sd}(\widehat{A}_t)} m(s_t) \tilde{f}^{\mathrm{CF}}(s_t, a_t),$ ▷ **Advantage Perturbation**
  for $k = 1, \dots, K$ **do**
    for all minibatches $\mathcal{B} \subset \mathcal{D}$ **do**
      $\rho_t(\theta) \leftarrow \frac{\pi_\theta(a_t \mid s_t)}{\pi_0(a_t \mid s_t)}, \quad t \in \mathcal{B}$
      $L_{\mathrm{policy}} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \min\Big\{\rho_t(\theta) A_t^{\mathrm{CF}}, \mathrm{clip}\big(\rho_t(\theta), 1-\epsilon, 1+\epsilon\big) A_t^{\mathrm{CF}}\Big\}$
      $L_{\mathrm{critic}} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \Big(V_\phi(s_t) - \widehat{R}_t\Big)^2$
      $\theta \leftarrow \theta + \eta_\pi \nabla_\theta L_{\mathrm{policy}}$
      $\phi \leftarrow \phi - \eta_V \nabla_\phi L_{\mathrm{critic}}$
    end for
  end for
end for

Algorithm 2 TD-CF: Counterfactual-based Perturbation at TD-level for PPO

Input: PPO iterations I, query fraction q, counterfactual horizon H, counterfactual rollouts N, perturbation strength β
Input: Discount factor γ, GAE parameter λ, PPO clipping coefficient ϵ, update epochs K
Output: Policy parameters θ and critic parameters ϕ

Initialize policy π_θ, critic V_ϕ
for $i = 1, \dots, I$ **do**
  $\pi_0 \leftarrow \pi_\theta$ ▷ Freeze rollout policy
  collect trajectories under π_0, $\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}_{t=0}^{T-1}$
  $\mathcal{Q} \sim \mathrm{UniformSubset}\Big(\{(t, s_t)\}_{t=0}^{T-1}, \max\{1, \mathrm{round}(qT)\}\Big)$ ▷ Sample a fraction q of rollout states
  $m(s_t) \leftarrow \mathbb{I}[(t, s_t) \in \mathcal{Q}],\ t = 0, \dots, T-1$ ▷ $m(s_t) = 1$ for sampled states and $m(s_t) = 0$ otherwise
  for all $(t, s_t) \in \mathcal{Q}$ **do**
    for all $a \in \mathcal{A}$ **do**
      for $n = 1, \dots, N$ **do**
        Force action a, then follow π_0 for the remaining $H - 1$ steps
        Compute returns: $G_{t,n}^{\mathrm{CF}}(s_t, a) \leftarrow \sum_{\ell=0}^{H-1} \gamma^\ell r_{t+\ell}^{\mathrm{CF}} + \gamma^H \big(1 - d_{t+H-1}^{\mathrm{CF}}\big) V_\phi\Big(s_{t+H}^{\mathrm{CF}}\Big)$
      end for
      $f^{\mathrm{CF}}(s_t, a) \leftarrow \frac{1}{N} \sum_{n=1}^{N} G_{t,n}^{\mathrm{CF}}(s_t, a)$ ▷ Mean counterfactual returns
    end for
    for all $a \in \mathcal{A}$ **do**
      $\tilde{f}^{\mathrm{CF}}(s_t, a) \leftarrow f^{\mathrm{CF}}(s_t, a) - \sum_{a' \in \mathcal{A}} \pi_0(a' \mid s_t) f^{\mathrm{CF}}(s_t, a')$ ▷ Obtain perturbation signal
    end for
  end for

  $\widehat{A}_t \leftarrow \sum_{\ell=0}^{T-t-1} (\gamma\lambda)^\ell \left[r_{t+\ell} + \gamma V_\phi(s_{t+\ell+1}) - V_\phi(s_{t+\ell})\right]$
  $\kappa \leftarrow \dfrac{\mathrm{RMS}\Big(m(s_t)\tilde{f}^{\mathrm{CF}}(s_t, a_t)\Big)}{\mathrm{RMS}\Big(\sum_{\ell=0}^{T-t-1} (\gamma\lambda)^\ell m(s_{t+\ell}) \tilde{f}^{\mathrm{CF}}(s_{t+\ell}, a_{t+\ell})\Big)}$ ▷ To Match TD and local guidance scales
  $\delta_t^{\mathrm{CF}} \leftarrow r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t) + \beta\, \kappa m(s_t) \tilde{f}^{\mathrm{CF}}(s_t, a_t)$
  ▷ **TD-level perturbation**
  $\widehat{A}_t^{\mathrm{CF}} \leftarrow \sum_{\ell=0}^{T-t-1} (\gamma\lambda)^\ell \delta_{t+\ell}^{\mathrm{CF}}$
  $\widehat{R}_t^{\mathrm{CF}} \leftarrow \widehat{A}_t^{\mathrm{CF}} + V_\phi(s_t)$

  for $k = 1, \dots, K$ **do**
    for all minibatches $\mathcal{B} \subset \mathcal{D}$ **do**
      $\rho_t(\theta) \leftarrow \dfrac{\pi_\theta(a_t \mid s_t)}{\pi_0(a_t \mid s_t)}, \qquad t \in \mathcal{B}$
      $L_{\mathrm{policy}} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \min\Big\{\rho_t(\theta) \widehat{A}_t^{\mathrm{CF\ Norm}}, \mathrm{clip}\big(\rho_t(\theta), 1 - \epsilon, 1 + \epsilon\big) \widehat{A}_t^{\mathrm{CF\ Norm}}\Big\}$
      $L_{\mathrm{critic}} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \Big(V_\phi(s_t) - \widehat{R}_t^{\mathrm{CF}}\Big)^2$
      $\theta \leftarrow \theta + \eta_\pi \nabla_\theta L_{\mathrm{policy}}$
      $\phi \leftarrow \phi - \eta_V \nabla_\phi L_{\mathrm{critic}}$
    end for
  end for
end for

Algorithm 3 Loss-CF: Counterfactual-based Perturbation at loss-level for PPO

Input: PPO iterations I, query fraction q, counterfactual horizon H, counterfactual rollouts N, perturbation strength β
Input: Discount factor γ, GAE parameter λ, PPO clipping coefficient ϵ, update epochs K
Output: Policy parameters θ and critic parameters ϕ

Initialize policy π_θ, critic V_ϕ
for $i = 1, \dots, I$ **do**
  $\pi_0 \leftarrow \pi_\theta$ ▷ Freeze rollout policy
  collect trajectories under π_0, $\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}_{t=0}^{T-1}$
  $\mathcal{Q} \sim \mathrm{UniformSubset}\Big(\{(t, s_t)\}_{t=0}^{T-1}, \max\{1, \mathrm{round}(qT)\}\Big)$ ▷ Sample a fraction q of rollout states
  $m(s_t) \leftarrow \mathbb{I}[(t, s_t) \in \mathcal{Q}], \quad t = 0, \dots, T-1$ ▷ $m(s_t) = 1$ for sampled states and $m(s_t) = 0$ otherwise
  for all $(t, s_t) \in \mathcal{Q}$ **do**
    for all $a \in \mathcal{A}$ **do**
      for $n = 1, \dots, N$ **do**
        Restore the environment state at s_t
        Force action a, then follow π_0 for the remaining $H-1$ steps
        Compute returns: $G_{t,n}^{\mathrm{CF}}(s_t, a) \leftarrow \sum_{\ell=0}^{H-1} \gamma^\ell r_{t+\ell}^{\mathrm{CF}} + \gamma^H \big(1 - d_{t+H-1}^{\mathrm{CF}}\big) V_\phi\Big(s_{t+H}^{\mathrm{CF}}\Big)$
      end for
      $f^{\mathrm{CF}}(s_t, a) \leftarrow \frac{1}{N} \sum_{n=1}^{N} G_{t,n}^{\mathrm{CF}}(s_t, a)$ ▷ Mean counterfactual returns
    end for
    for all $a \in \mathcal{A}$ **do**
      $\tilde{f}^{\mathrm{CF}}(s_t, a) \leftarrow f^{\mathrm{CF}}(s_t, a) - \sum_{a' \in \mathcal{A}} \pi_0(a' \mid s_t) f^{\mathrm{CF}}(s_t, a')$ ▷ Obtain perturbation signal
    end for
  end for

  $\widehat{A}_t \leftarrow \sum_{\ell=0}^{T-t-1} (\gamma\lambda)^\ell \left[r_{t+\ell} + \gamma V_\phi(s_{t+\ell+1}) - V_\phi(s_{t+\ell})\right]$
  $\widehat{R}_t \leftarrow \widehat{A}_t + V_\phi(s_t)$

  for $k = 1, \dots, K$ **do**
    for all minibatches $\mathcal{B} \subset \mathcal{D}$ **do**
      $\rho_t(\theta) \leftarrow \frac{\pi_\theta(a_t \mid s_t)}{\pi_0(a_t \mid s_t)}, \quad t \in \mathcal{B}$
      $L^{\mathrm{PPO}} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \min\Big\{\rho_t(\theta) \widehat{A}_t^{\mathrm{Norm}}, \mathrm{clip}\big(\rho_t(\theta), 1-\epsilon, 1+\epsilon\big) \widehat{A}_t^{\mathrm{Norm}}\Big\}$
      $L^{\mathrm{CF}} \leftarrow \frac{\beta}{\mathrm{sd}(\widehat{A}_t)} \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} m(s_t) \sum_{a \in \mathcal{A}} \pi_\theta(a \mid s_t) \tilde{f}^{\mathrm{CF}}(s_t, a)$ ▷ All-action counterfactual guidance
      $L_{\mathrm{policy}} \leftarrow L^{\mathrm{PPO}} + L^{\mathrm{CF}}$
      $L_{\mathrm{critic}} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \Big(V_\phi(s_t) - \widehat{R}_t\Big)^2$
      $\theta \leftarrow \theta + \eta_\pi \nabla_\theta L_{\mathrm{policy}}$
      $\phi \leftarrow \phi - \eta_V \nabla_\phi L_{\mathrm{critic}}$
    end for
  end for
end for

B Proofs for Action-Relative Advantage Perturbations

This appendix proves the local population claims in Section 3. We use the main-text notation: $\pi_0 = \pi_{\theta_0}$, d^{π_0} is the normalized discounted state-visitation distribution, and $A^{\pi_0}(s, a)$ is the advantage. For compactness, write

$$\mathbb{E}_{s,a}[\cdot] = \mathbb{E}_{\substack{s \sim d^{\pi_0} \\ a \sim \pi_0(\cdot|s)}}[\cdot].$$

B.1 Definitions and assumptions

For the following analysis, define the importance ratio $\rho_\theta(s,a) \coloneqq \frac{\pi_\theta(a|s)}{\pi_0(a|s)}$, and consider the unclipped policy objective

$$L(\theta) \coloneqq \mathbb{E}_{s,a}[\rho_\theta(s,a)\, A^{\pi_0}(s,a)]\,,$$

where $A^{\pi_0}(s,a)$ denotes the advantage function under the rollout policy π_0. Let the policy score at the rollout parameters be $z_{\theta_0}(s,a) \coloneqq \nabla_\theta \log \pi_\theta(a \mid s)|_{\theta=\theta_0}$. The corresponding local policy gradient is

$$g_0 \coloneqq \nabla_\theta L(\theta)|_{\theta=\theta_0} = \mathbb{E}_{s,a}[A^{\pi_0}(s,a)\, z_{\theta_0}(s,a)]\,. \tag{15}$$

Given an external perturbation function $f(s,a)$ and perturbation strength $\beta \geq 0$, define the perturbed advantage by

$$A^{\mathrm{pert}}(s,a) \coloneqq A^{\pi_0}(s,a) + \beta f(s,a).$$

The resulting policy objective is

$$L_\beta(\theta) \coloneqq \mathbb{E}_{s,a}\big[\rho_\theta(s,a)\, A^{\mathrm{pert}}(s,a)\big]\,,$$

with local policy gradient

$$g_\beta \coloneqq \nabla_\theta L_\beta(\theta)|_{\theta=\theta_0}\,.$$

Assumption 1 (regularity). For every state s in the support of d^{π_0}, where π_0 denotes the frozen rollout policy, we assume that $\pi_0(a \mid s) > 0$ for all $a \in \mathcal{A}$; moreover, $\pi_\theta(a \mid s)$ is differentiable with respect to θ in a neighborhood of θ_0, has fixed support in that neighborhood, and differentiation may be interchanged with the expectations below.

Assumption 2 (Integrability). We assume that all expectations appearing below are finite. In particular,

$$\mathbb{E}_{s,a}[|A^{\pi_0}(s,a)|\,|z_{\theta_0}(s,a)|] < \infty, \qquad \mathbb{E}_{s,a}[|f(s,a)|\,|z_{\theta_0}(s,a)|] < \infty, \qquad \mathbb{E}_{s,a}[|z_{\theta_0}(s,a)|] < \infty.$$

Assumption 3 (Counterfactual estimation error). Let $\widehat{Q}^{\pi_0,\mathrm{CF}}(s,a)$ be a counterfactual action value estimate, and $Q^{\pi_0}(s,a)$ denote the true action-value, and define the counterfactual score as $f^{\mathrm{CF}}(s,a) \coloneqq \widehat{Q}^{\pi_0,\mathrm{CF}}(s,a)$. Its estimation error is $e(s,a) \coloneqq \widehat{Q}^{\pi_0,\mathrm{CF}}(s,a) - Q^{\pi_0}(s,a)$, which we assume is uniformly bounded:

$$\|e\|_\infty \coloneqq \sup_{s,a} |e(s,a)| \leq \epsilon_{\mathrm{CF}}.$$

Equivalently,

$$f^{\mathrm{CF}}(s,a) = Q^{\pi_0}(s,a) + e(s,a).$$

Under exact counterfactual evaluation, $\widehat{Q}^{\pi_0,\mathrm{CF}}(s,a) = Q^{\pi_0}(s,a)$ for every (s,a); hence, $e(s,a) = 0$ and $f^{\mathrm{CF}}(s,a) = Q^{\pi_0}(s,a)$.

B.2 Action-relative perturbations

Lemma 1 (Policy-score identity)**.** *For every state s, $\mathbb{E}_{a\sim\pi_0(\cdot|s)}\,[z_{\theta_0}(s,a)] = 0$. Consequently, for any state-only function $c(s)$, $\mathbb{E}_{s,a}\,[c(s) z_{\theta_0}(s,a)] = 0$.*

Proof of Lemma 1. For fixed s,

$$\begin{aligned}
\mathbb{E}_{a\sim\pi_0(\cdot|s)}\,[z_{\theta_0}(s,a)] &= \sum_{a\in\mathcal{A}} \pi_0(a \mid s)\; \nabla_\theta \log \pi_\theta(a \mid s)|_{\theta=\theta_0} \\
&= \sum_{a\in\mathcal{A}} \nabla_\theta \pi_\theta(a \mid s)\Big|_{\theta=\theta_0} \\
&= \nabla_\theta \sum_{a\in\mathcal{A}} \pi_\theta(a \mid s)\Big|_{\theta=\theta_0} \\
&= \nabla_\theta 1 = 0.
\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{E}_{s,a}\left[c(s)z_{\theta_0}(s,a)\right] &= \mathbb{E}_{s\sim d^{\pi_0}}\left[c(s)\mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[z_{\theta_0}(s,a)\right]\right] \\ &= 0.\end{aligned}$$

□

Proof of Proposition 1. Proposition 1 states that an external advantage perturbation changes the local policy gradient only through its action-relative component. Any state-only component acts as a baseline and therefore has zero expected contribution to the policy gradient.

To derive the policy-gradient shift shown in equation 4, we begin with the local gradient induced by the perturbed advantage:

$$g_\beta = \nabla_\theta L_\beta(\theta)|_{\theta=\theta_0}.$$

Since the rollout distribution and the perturbed advantage $A^{\text{pert}}(s,a)$ are held fixed during this local derivative,

$$g_\beta = \mathbb{E}_{s,a}\left[A^{\text{pert}}(s,a)\ \nabla_\theta\rho_\theta(s,a)|_{\theta=\theta_0}\right].$$

Consider the gradient importance ratio wrt θ,

$$\begin{aligned}\nabla_\theta\rho_\theta(s,a)|_{\theta=\theta_0} &= \nabla_\theta\frac{\pi_\theta(a\mid s)}{\pi_0(a\mid s)}\bigg|_{\theta=\theta_0} \\ &= \frac{\pi_\theta(a\mid s)}{\pi_0(a\mid s)}\nabla_\theta\log\pi_\theta(a\mid s)\bigg|_{\theta=\theta_0} \\ &= z_{\theta_0}(s,a),\end{aligned}$$

where the final equality follows from $\pi_{\theta_0}=\pi_0$. Therefore,

$$\begin{aligned}g_\beta &= \mathbb{E}_{s,a}\left[A^{\text{pert}}(s,a)z_{\theta_0}(s,a)\right] \\ &= \mathbb{E}_{s,a}\left[\left(A^{\pi_0}(s,a)+\beta f(s,a)\right)z_{\theta_0}(s,a)\right] \\ &= \mathbb{E}_{s,a}\left[A^{\pi_0}(s,a)z_{\theta_0}(s,a)\right]+\beta\mathbb{E}_{s,a}\left[f(s,a)z_{\theta_0}(s,a)\right] \\ &= g_0+\beta\mathbb{E}_{s,a}\left[f(s,a)z_{\theta_0}(s,a)\right], \qquad \text{by equation 15.}\end{aligned}$$

Using the decomposition $f(s,a)=\tilde{f}(s,a)+c(s)$,

$$g_\beta = g_0+\beta\mathbb{E}_{s,a}\left[\tilde{f}(s,a)z_{\theta_0}(s,a)\right]+\beta\mathbb{E}_{s,a}\left[c(s)z_{\theta_0}(s,a)\right].$$

By Lemma 1,

$$\mathbb{E}_{s,a}\left[c(s)z_{\theta_0}(s,a)\right]=0.$$

Hence,

$$g_\beta = g_0+\beta\mathbb{E}_{s,a}\left[\tilde{f}(s,a)z_{\theta_0}(s,a)\right],$$

which proves equation 4. □

B.3 COUNTERFACTUAL PERTURBATIONS AND LOCAL ALIGNMENT

We use the counterfactual (CF) notation from the main text. The CF perturbation is constructed from the action-value estimate $\widehat{Q}^{\pi_0,\mathrm{CF}}(s,a)$ for action a at state s. Since this estimate may differ from the true action-value $Q^{\pi_0}(s,a)$, define the CF score by $f^{\mathrm{CF}}(s,a) \coloneqq \widehat{Q}^{\pi_0,\mathrm{CF}}(s,a)$ and write $f^{\mathrm{CF}}(s,a) = Q^{\pi_0}(s,a) + e(s,a)$, where $e(s,a)$ is the estimation error and $\|e\|_\infty \le \epsilon_{\mathrm{CF}}$. Under exact counterfactual evaluation, $e(s,a) = 0$ and $f^{\mathrm{CF}}(s,a) = Q^{\pi_0}(s,a)$.

Lemma 2 (Counterfactual decomposition)**.** *Define*

$$\tilde{f}^{\mathrm{CF}}(s,a) = f^{\mathrm{CF}}(s,a) - \mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[f^{\mathrm{CF}}(s,a')\right]$$

and

$$\tilde{e}(s,a) = e(s,a) - \mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[e(s,a')\right].$$

Then

$$\tilde{f}^{\mathrm{CF}}(s,a) = A^{\pi_0}(s,a) + \tilde{e}(s,a), \qquad \|\tilde{e}\|_\infty \le 2\epsilon_{\mathrm{CF}}.$$

Proof. By the definition of $\tilde{f}^{\mathrm{CF}}(s,a)$ and the CF error model,

$$\begin{aligned}
\tilde{f}^{\mathrm{CF}}(s,a) &= f^{\mathrm{CF}}(s,a) - \mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[f^{\mathrm{CF}}(s,a')\right] \\
&= \left[Q^{\pi_0}(s,a) + e(s,a)\right] - \mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[Q^{\pi_0}(s,a') + e(s,a')\right] \\
&= \left[Q^{\pi_0}(s,a) - \mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[Q^{\pi_0}(s,a')\right]\right] + \tilde{e}(s,a) \\
&= Q^{\pi_0}(s,a) - V^{\pi_0}(s) + \tilde{e}(s,a) \\
&= A^{\pi_0}(s,a) + \tilde{e}(s,a).
\end{aligned}$$

Moreover,

$$\begin{aligned}
|\tilde{e}(s,a)| &= \left|e(s,a) - \mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[e(s,a')\right]\right| \\
&\le |e(s,a)| + \left|\mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[e(s,a')\right]\right| \qquad \text{(Using triangle inequality)} \\
&\le |e(s,a)| + \mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[|e(s,a')|\right] \qquad \text{(Using Jensen's inequality)} \\
&\le 2\epsilon_{\mathrm{CF}}, \qquad \text{since } \|e\|_\infty \le \epsilon_{\mathrm{CF}}.
\end{aligned}$$

Therefore,

$$\|\tilde{e}\|_\infty = \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} |\tilde{e}(s,a)| \le 2\epsilon_{\mathrm{CF}},$$

which proves the result. □

Proof of Proposition 2. For the CF score $f^{\mathrm{CF}}(s,a)$, define the perturbed advantage

$$A^{\mathrm{pert}}(s,a) = A^{\pi_0}(s,a) + \beta f^{\mathrm{CF}}(s,a).$$

Let

$$c^{\mathrm{CF}}(s) = \mathbb{E}_{a'\sim\pi_0(\cdot|s)}\left[f^{\mathrm{CF}}(s,a')\right].$$

Since

$$f^{\mathrm{CF}}(s,a) = \tilde{f}^{\mathrm{CF}}(s,a) + c^{\mathrm{CF}}(s),$$

Lemma 1 removes the state-only term $c^{\mathrm{CF}}(s)$ from the local actor gradient. Therefore,

$$g_\beta = g_0 + \beta\mathbb{E}_{s,a}\left[\tilde{f}^{\mathrm{CF}}(s,a) z_{\theta_0}(s,a)\right].$$

By Lemma 2,

$$\begin{aligned}
g_\beta &= g_0 + \beta\mathbb{E}_{s,a}\left[\left(A^{\pi_0}(s,a) + \tilde{e}(s,a)\right) z_{\theta_0}(s,a)\right] \\
&= (1+\beta)g_0 + \beta g_e,
\end{aligned}$$

where

$$g_e = \mathbb{E}_{s,a}\left[\tilde{e}(s,a) z_{\theta_0}(s,a)\right].$$

The norm of g_e is bounded by

$$\begin{aligned}\|g_e\| &= \|\mathbb{E}_{s,a}\left[\tilde{e}(s,a) z_{\theta_0}(s,a)\right]\| \\ &\leq \mathbb{E}_{s,a}\left[|\tilde{e}(s,a)| \|z_{\theta_0}(s,a)\|\right] \\ &\leq 2\epsilon_{\mathrm{CF}} \mathbb{E}_{s,a}\left[\|z_{\theta_0}(s,a)\|\right].\end{aligned}$$

This proves equation 10.

Finally, consider the inner product between g_0 and g_β. Using $g_\beta = (1+\beta)g_0 + \beta g_e$,

$$\begin{aligned}\langle g_0, g_\beta\rangle &= \langle g_0, (1+\beta)g_0 + \beta g_e\rangle \\ &= (1+\beta)\langle g_0, g_0\rangle + \beta\langle g_0, g_e\rangle \\ &= (1+\beta)\|g_0\|^2 + \beta\langle g_0, g_e\rangle \\ &\geq (1+\beta)\|g_0\|^2 - \beta\left|\langle g_0, g_e\rangle\right| \qquad \text{(since } x \geq -|x|\text{)} \\ &\geq (1+\beta)\|g_0\|^2 - \beta\|g_0\|\|g_e\| \qquad \text{(Cauchy–Schwarz inequality)} \\ &\geq (1+\beta)\|g_0\|^2 - 2\beta\epsilon_{\mathrm{CF}}\|g_0\|\mathbb{E}_{s,a}\left[\|z_{\theta_0}(s,a)\|\right] \\ &\geq \|g_0\|\left[(1+\beta)\|g_0\| - 2\beta\,\epsilon_{\mathrm{CF}}\mathbb{E}_{s,a}\left[\|z_{\theta_0}(s,a)\|\right]\right]\end{aligned}$$

Thus, a sufficient condition for the dot product $\langle g_0, g_\beta\rangle$ to remain positive is

$$2\beta\epsilon_{\mathrm{CF}}\mathbb{E}_{s,a}\left[\|z_{\theta_0}(s,a)\|\right] < (1+\beta)\|g_0\|.$$

Thus, under this bound, counterfactual estimation error cannot make the perturbed update oppose the ordinary PPO update, and $\langle g_0, g_\beta\rangle > 0$. □

Corollary 1 (Consistent counterfactual evaluation)**.** *For fixed $\beta \geq 0$, if the counterfactual estimation error satisfies $\epsilon_{\mathrm{CF}} \to 0$, then*

$$g_\beta \longrightarrow (1+\beta)g_0$$

Equivalently,

$$\|g_\beta - (1+\beta)g_0\| \longrightarrow 0.$$

Under exact counterfactual evaluation,

$$g_\beta = (1+\beta)g_0$$

Proof. From Proposition 2,

$$g_\beta = (1+\beta)g_0 + \beta g_e$$

Therefore,

$$g_\beta - (1+\beta)g_0 = \beta g_e$$

and taking norms gives

$$\begin{aligned}\|g_\beta - (1+\beta)g_0\| &= \beta\|g_e\| \\ &\leq 2\beta\epsilon_{\mathrm{CF}}\mathbb{E}_{s,a}\left[\|z_{\theta_0}(s,a)\|\right]\end{aligned}$$

The expectation is finite by Assumption 2. Hence, as $\epsilon_{\mathrm{CF}} \to 0$,

$$\|g_\beta - (1+\beta)g_0\| \longrightarrow 0$$

Thus, as the CF evaluator becomes increasingly accurate ($\epsilon_{\mathrm{CF}} \to 0$), the CF-perturbed gradient g_β converges to $(1+\beta)g_0$, a positive rescaling of the ordinary PPO gradient.

Under exact counterfactual evaluation, $e(s,a) = 0$, and hence $\tilde{e}(s,a) = 0$ and $g_e = 0$. Substituting $g_e = 0$ into the gradient decomposition gives

$$\begin{aligned}g_\beta &= (1+\beta)g_0 + \beta g_e \\ &= (1+\beta)g_0.\end{aligned}$$

Thus, as counterfactual estimation error vanishes, the CF-perturbed gradient converges to a positive rescaling of the ordinary PPO gradient. □

Remark (Uniform state subsampling). In the implementation (for computational feasibility), counterfactual guidance is applied only to a uniformly sampled fraction $q \in (0, 1]$ of rollout states. Let $m(s) \in \{0, 1\}$ denote the corresponding query mask, with

$$\mathbb{E}_m[m(s) \mid s] = q,$$

independently of the sampled action. Replacing $\widetilde{f}_{\mathrm{CF}}(s, a)$ by $m(s)\widetilde{f}_{\mathrm{CF}}(s, a)$ and taking expectation also over the random mask gives

$$\mathbb{E}_m[g_{\beta,m}] = g_0 + \beta q\, \mathbb{E}_{s,a}\left[\widetilde{f}_{\mathrm{CF}}(s, a) z_{\theta_0}(s, a)\right].$$

Hence, under exact counterfactual evaluation, where $\widetilde{f}_{\mathrm{CF}}(s, a) = A^{\pi_0}(s, a)$,

$$\mathbb{E}_m[g_{\beta,m}] = (1 + \beta q) g_0.$$

Thus, uniform state subsampling scales the counterfactual contribution by the query fraction q in expectation and preserves its population gradient direction; a particular realized mask may introduce additional sampling variability.

C Controlled CartPole Methodology

This appendix provides the implementation and measurement details required to reproduce the controlled CartPole experiments in Section 3.3.

C.1 Environment and PPO implementation

We use the standard CartPole-v1 environment from Gym (Brockman et al., 2016). The observation is $s = (x, \dot{x}, \theta, \dot{\theta})$, where x is the cart position and θ is the pole angle. The action space is $\mathcal{A} = \{0, 1\}$, corresponding to pushing the cart left or right. Table 3 summarizes the training and PPO configuration used in all controlled CartPole experiments.

Table 3: Training and PPO configuration for the controlled CartPole experiments.

Quantity	Value
Parallel environments	4
Rollout length	256 steps per environment
Training budget	150,000 environment transitions
Policy network	$(64, 64)$ $\tanh$ MLP
Value network	$(64, 64)$ $\tanh$ MLP
Learning rate	3×10^{-4}
Discount factor	$\gamma = 0.99$
GAE parameter	$\lambda = 0.95$
PPO epochs	10
Minibatches per epoch	4
PPO clipping coefficient	0.2
Value clipping	Enabled
Entropy coefficient	0.01
Value-loss coefficient	0.5
Gradient-norm limit	0.5
Advantage normalization	Rollout-batch standardization

C.2 Perturbation implementation

For the controlled CartPole experiments, we use the perturbation $f(s, a) = h(s)\,[(1 - \chi) + \chi u(a \mid s)]$, as defined in Section 3.3. The state-dependent component $h(s)$ is fixed throughout training and shared across all seeds and perturbation conditions. We use the smooth nonlinear function

$$h(s) = c\sqrt{2} \cos\left(w^\top \frac{s - \mu_{\mathrm{ref}}}{\sigma_{\mathrm{ref}}} + b\right),$$

where $w = (-0.0277348, -0.4115472, 1.2225838, -0.9448223)$ and $b = 4.2634294$ were drawn once using a fixed random seed, with $w \sim \mathcal{N}(0, I_4)$ and $b \sim \mathrm{Uniform}(0, 2\pi)$, and then held fixed across all experiments. The scale $c = 1.0376130$ normalizes $h(s)$ to have unit empirical standard deviation on the fixed reference dataset. Here μ_{ref} and σ_{ref} are the element-wise mean and standard deviation of CartPole state vectors, used to standardize the observation. We use this cosine form simply as a convenient fixed, bounded, and nonconstant state-dependent signal. Its particular form is not important to the experiment or theory: it is held fixed so that the comparison isolates the effect of adding action-relative information.

For the action-relative component, we assign fixed raw scores $\bar{u}(0) = -1$ and $\bar{u}(1) = 1$ and define the action-relative component under rollout policy π_0 as:

$$u(a \mid s) = \bar{u}(a) - \sum_{a' \in \{0,1\}} \pi_0(a' \mid s)\bar{u}(a').$$

Thus, $u(a \mid s)$ is computed using the rollout policy π_0 and remains fixed throughout the PPO optimization epochs associated with that rollout. Because this action-relative component assigns preferences independently of the task reward, it can oppose reward-supported action choices, and thereby slow PPO learning as its strength increases.

Within each PPO minibatch, we normalize the perturbation signal $f(s, a)$ and rescale it to match the standard deviation of the sampled advantage $\widehat{A}(s, a)$. Here, β controls the perturbation strength and χ controls its action-relative fraction: $\chi = 0$ gives a state-only perturbation, whereas $\chi = 1$ gives a fully action-relative perturbation. The perturbation modifies only the advantage supplied to the PPO actor; environment rewards and critic targets remain unchanged. For Figure 1A, we compare state-only ($\chi = 0$) and fully action-relative ($\chi = 1$) perturbations at $\beta \in \{0.02, 0.05, 0.10, 0.50\}$. For Figure 1B, we vary $\chi \in \{0, 0.25, 0.5, 1\}$ at $\beta \in \{0.05, 0.10\}$. Each condition is trained using five independent random seeds.

For the directional steering experiment, we use the targeted action-relative perturbations defined in Section 3.3. The perturbation is nonzero only within the trigger region $\mathcal{T}$ and is zero elsewhere; all other rollout, reward, action-sampling, optimizer, and training settings are unchanged. We train standard PPO, Perturb-1, and Perturb-2 using 15 independent random seeds. For Figure 1D, we measure the angle between the ordinary PPO actor-gradient estimate g_0 and the perturbation-modified estimate g_β for each seed during training: $\angle(g_\beta, g_0) = \cos^{-1}(\langle g_\beta, g_0 \rangle / (\|g_\beta\|_2 \|g_0\|_2))$.

C.3 Manual-oracle implementation

The manual oracle is a task-specific symbolic action scorer. It does not restore the environment, run counterfactual trajectories, or use the critic to estimate action values. Instead, it reads the encoded MiniGrid state and applies hand-written task rules to rank the available actions.

For a rollout state s, let $x(s)$ denote the task-relevant symbolic state extracted from s. In MiniGrid, $x(s)$ contains the grid layout, agent position and direction, carried object, relevant object locations, and door states. It omits information that does not affect task completion.

For each environment, we define a deterministic symbolic transition function T. Given a symbolic state $x(s)$ and an action a, $T(x(s), a)$ applies the corresponding hand-written MiniGrid rule and returns the updated symbolic state. Turning changes the agent direction; moving forward changes its position only when the front cell is passable; pickup transfers a front object to the agent's hand; drop places a carried object in the front cell; and toggle updates a door or object according to the task rules. Invalid actions leave the relevant symbolic state components unchanged. We write the symbolic state after applying action a as

$$\mathrm{successor}(s, a) \coloneqq T(x(s), a)\,.$$

The oracle defines the completion cost $C(x)$ as the minimum number of further valid actions needed to complete the task from symbolic state x:

$$C(x) = \begin{cases} 0, & \text{if the task is complete in } x, \\ 1 + \min_{a \in \mathcal{A}} C(T(x, a))\,, & \text{otherwise.} \end{cases}$$

Table 4: PPO hyperparameters across the four MiniGrid environments.

Hyperparameter	RedBlueDoors 6×6	KeyCorridor S3R3	DoorKey 6×6	UnlockPickup
Training budget	1.5M	1.0M	750k	1.5M
Seeds	5	5	5	5
Parallel environments	16	8	16	16
Rollout steps per environment	512	128	128	128
Rollout batch size	8192	1024	2048	2048
PPO epochs per rollout	4	10	4	4
Minibatches per epoch	8	16	8	8
Minibatch size	1024	64	256	256
Optimizer	Adam	Adam	Adam	Adam
Adam ϵ	10^{-8}	10^{-8}	10^{-8}	10^{-8}
Learning rate	10^{-3}	5×10^{-4}	10^{-3}	10^{-3}
Discount factor, γ	0.99	0.99	0.99	0.99
GAE parameter, λ	0.95	0.95	0.95	0.95
PPO clipping coefficient	0.20	0.20	0.20	0.20
Value-loss coefficient	0.50	0.50	0.50	0.50
Entropy coefficient	0.01	0.10	0.10	0.10
Encoder	Shared CNN	Shared CNN	Shared CNN	Shared CNN
Perturbation strength, β	0	0	0	0

Equivalently, the symbolic states form a graph, valid actions form unit-cost edges, and task-complete states have cost zero. The oracle computes the shortest-path cost to completion for each reachable symbolic state and caches these costs for repeated states from the same layout.

$$f^{\text{manual}}(s,a) = -C(\text{successor}(s,a)) \,.$$

Actions that reduce the remaining completion cost therefore receive higher scores.

UnlockPickup illustrates this construction. The task requires the agent to pick up a key, unlock a door, drop the key, and pick up a box. Suppose that the agent faces the key, has an empty hand, and the door is locked. The pickup action produces a successor state in which the agent carries the key and has fewer remaining actions before task completion. It therefore receives a high score. An invalid toggle action leaves the door locked and does not reduce the completion cost. After the door is unlocked, dropping the key is scored highly because the agent needs an empty hand to pick up the box. The symbolic state retains the location of a dropped key, since it can block movement and may need to be picked up again. Toggling the box destroys the target and leaves no completion path; this action is assigned a failure cost slightly larger than the current completion cost and is ranked below all actions that preserve task completion.

Before actor optimization, the manual scores are centered under the frozen rollout policy:

$$\tilde{f}^{\text{manual}}(s,a) = f^{\text{manual}}(s,a) - \sum_{a' \in \mathcal{A}} \pi_0(a' \mid s) f^{\text{manual}}(s,a').$$

This retains only the oracle's relative action preferences at each state. The centered score is rescaled over active rollout states so that its policy-weighted root-mean-square matches the standard deviation of the unshaped rollout GAE advantages. States that are already terminal or have no finite symbolic completion path receive zero guidance.

The resulting $\tilde{f}^{\text{manual}}(s,a)$ is used in the manual advantage perturbation defined in the main text. Thus, manual guidance modifies only the actor advantage; environment rewards, GAE computation, return targets, and the critic objective remain unchanged.

D Training Parameters

Table 5: PPO-manual hyperparameters across the four MiniGrid environments.

Hyperparameter	RedBlueDoors 6×6	KeyCorridor S3R3	DoorKey 6×6	UnlockPickup
Training budget	1.5M	1.0M	750k	1.5M
Seeds	5	5	5	5
Parallel environments	16	8	16	16
Rollout steps per environment	512	128	128	128
Rollout batch size	8192	1024	2048	2048
PPO epochs per rollout	4	10	4	4
Minibatches per epoch	8	16	8	8
Minibatch size	1024	64	256	256
Optimizer	Adam	Adam	Adam	Adam
Adam ϵ	10^{-8}	10^{-8}	10^{-8}	10^{-8}
Learning rate	10^{-3}	5×10^{-4}	10^{-3}	10^{-3}
Discount factor, γ	0.99	0.99	0.99	0.99
GAE parameter, λ	0.95	0.95	0.95	0.95
PPO clipping coefficient	0.20	0.20	0.20	0.20
Value-loss coefficient	0.50	0.50	0.50	0.50
Entropy coefficient	0.01	0.10	0.10	0.10
Encoder	Shared CNN	Shared CNN	Shared CNN	Shared CNN
Manual guidance strength, β	0.75	0.75	0.75	0.75

Table 6: Loss-CF hyperparameters across the four MiniGrid environments. Loss-CF corresponds to the implementation label PPO-CF.

Hyperparameter	RedBlueDoors 6×6	KeyCorridor S3R3	DoorKey 6×6	UnlockPickup
Training budget	1.5M	1.0M	750k	1.5M
Seeds	5	5	5	5
Parallel environments	16	8	16	16
Rollout steps per environment	512	128	128	128
Rollout batch size	8192	1024	2048	2048
PPO epochs per rollout	4	10	4	4
Minibatches per epoch	8	16	8	8
Minibatch size	1024	64	256	256
Optimizer	Adam	Adam	Adam	Adam
Adam ϵ	10^{-8}	10^{-8}	10^{-8}	10^{-8}
Learning rate	10^{-3}	5×10^{-4}	10^{-3}	10^{-3}
Discount factor, γ	0.99	0.99	0.99	0.99
GAE parameter, λ	0.95	0.95	0.95	0.95
PPO clipping coefficient	0.20	0.20	0.20	0.20
Value-loss coefficient	0.50	0.50	0.50	0.50
Entropy coefficient	0.01	0.10	0.10	0.10
Encoder	Shared CNN	Shared CNN	Shared CNN	Shared CNN
CF guidance strength, β	0.10	0.50	0.50	0.50
CF branch horizon, H	64	96	16	64
CF rollouts per action, R	2	2	2	2
CF query fraction, q	0.025	0.035	0.10	0.05

Table 7: Advantage-CF hyperparameters across the four MiniGrid environments.

Hyperparameter	RedBlueDoors 6×6	KeyCorridor S3R3	DoorKey 6×6	UnlockPickup
Training budget	1.5M	1.0M	750k	1.5M
Seeds	5	5	5	5
Parallel environments	16	8	16	16
Rollout steps per environment	512	128	128	128
Rollout batch size	8192	1024	2048	2048
PPO epochs per rollout	4	10	4	4
Minibatches per epoch	8	16	8	8
Minibatch size	1024	64	256	256
Optimizer	Adam	Adam	Adam	Adam
Adam ϵ	10^{-8}	10^{-8}	10^{-8}	10^{-8}
Learning rate	10^{-3}	5×10^{-4}	10^{-3}	10^{-3}
Discount factor, γ	0.99	0.99	0.99	0.99
GAE parameter, λ	0.95	0.95	0.95	0.95
PPO clipping coefficient	0.20	0.20	0.20	0.20
Value-loss coefficient	0.50	0.50	0.50	0.50
Entropy coefficient	0.01	0.10	0.10	0.10
Encoder	Shared CNN	Shared CNN	Shared CNN	Shared CNN
CF guidance strength, β	0.75	0.75	0.75	0.75
CF branch horizon, H	64	96	16	64
CF continuations per action, R	2	2	2	2
CF query fraction, q	0.025	0.035	0.10	0.05

Table 8: TD-CF hyperparameters across the four MiniGrid environments.

Hyperparameter	RedBlueDoors 6×6	KeyCorridor S3R3	DoorKey 6×6	UnlockPickup
Training budget	1.0M → 1.5M	1.0M	750k	1.5M
Seeds	5	5	5	5
Parallel environments	16	8	16	16
Rollout steps per environment	512	128	128	128
Rollout batch size	8192	1024	2048	2048
PPO epochs per rollout	4	10	4	4
Minibatches per epoch	8	16	8	8
Minibatch size	1024	64	256	256
Optimizer	Adam	Adam	Adam	Adam
Adam ϵ	10^{-8}	10^{-8}	10^{-8}	10^{-8}
Learning rate	10^{-3}	5×10^{-4}	10^{-3}	10^{-3}
Discount factor, γ	0.99	0.99	0.99	0.99
GAE parameter, λ	0.95	0.95	0.95	0.95
PPO clipping coefficient	0.20	0.20	0.20	0.20
Value-loss coefficient	0.50	0.50	0.50	0.50
Entropy coefficient	0.01	0.10	0.10	0.10
Encoder	Shared CNN	Shared CNN	Shared CNN	Shared CNN
CF guidance strength, β	0.75	0.75	0.75	0.75
CF branch horizon, H	64	96	16	64
CF continuations per action, R	2	2	2	2
CF query fraction, q	0.025	0.035	0.10	0.05