# LEARNING SAFE POLICIES FROM HARD COUNTERFACTUAL NEGATIVES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Safe reinforcement learning typically handles unsafe behavior by penalizing it during training or preventing it from being executed. However, when an unsafe action offers competitive task value, these mechanisms do not explicitly require the policy to learn a safe alternative that is better. We introduce **Hard-Negative Policy Optimization (HNPO)**, a world-model-based method that uses such reward-competitive, near-boundary violations, or *tempting failures*, as hard negatives for policy learning. HNPO first identifies unsafe actions that remain credible under model uncertainty and behavioral support. A conservative hard-negative proposer then maintains access to informative negatives as the task policy becomes safer, learning from validated examples rather than directly optimizing through world-model gradients. To learn from these negatives, HNPO evaluates safe policy actions and unsafe alternatives from the same latent state under a matched continuation policy. Hard-Negative Policy Contrast constructs a utility reference from the validated negative group and optimizes safe policy actions to outperform this reference by a margin, without treating proposer actions as on-policy samples. HNPO leaves the backbone safe-RL objective and deployment-time safety mechanism unchanged. Across four Safety-Gymnasium continuous-control tasks, HNPO improves task reward over its backbone while maintaining low episodic cost, with larger gains on tasks exhibiting stronger reward–safety conflict. Ablations further demonstrate the complementary roles of credible negative selection, proposer-based coverage, and the negative-only reference.

## 1 INTRODUCTION

Safe reinforcement learning (RL) aims to maximize task performance while satisfying safety constraints. Existing methods address this objective through constrained policy updates, cost critics, shielding, recovery policies, and safety-aware planning Achiam et al. (2017); Dalal et al. (2018); Thananjeyan et al. (2021); Huang et al. (2024). These mechanisms are effective at discouraging or preventing unsafe behavior, but leave a more specific policy-learning question largely implicit:*when an unsafe action is attractive in task value, has the policy learned a safe alternative that is actually better?*

This question matters because not all violations are equally informative. Severe failures with low reward are usually already dominated by the policy, whereas actions just beyond the safety boundary can remain competitive with, or even outperform, available safe behavior. We call such actions *tempting failures*. Although they should not be executed, they reveal what a competitive safe replacement must achieve. Rather than treating every violation only as behavior to penalize or reject, we therefore ask whether selected violations can serve as *hard negatives* against which safe policy actions are trained. This exposes an *unsafe-negative supervision gap*: reducing the probability of a violation Stooke et al. (2020); Ray et al. (2019), or rejecting it at execution time, does not by itself require the actor to find a high-value safe replacement at the same state.

Learning from such negatives in a world model is not straightforward. Useful negatives must be close enough to the safety boundary to be informative, competitive with the current policy, and credible under model uncertainty and behavioral support; otherwise they may be trivial failures or artifacts of model error Janner et al. (2019); Hafner et al. (2021). Moreover, informative violations become rarer as the task policy improves, so replay data and local perturbations alone may provide



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.

insufficient coverage. Finally, deliberately generated negatives come from a different action distribution than task-policy samples. Pooling the two groups in a conventional relative objective would make the comparison reference depend on the negative sampler, while using proposer actions as ordinary policy samples would yield an invalid policy update.

We introduce *Hard-Negative Policy Optimization (HNPO)* to address these issues. HNPO first identifies reward-competitive, near-boundary unsafe actions that remain credible under uncertainty and behavioral support. It then trains a dedicated hard-negative proposer from validated failures through behavior cloning and elite distillation. The proposer maintains negative coverage as the task policy evolves, but never controls the environment or directly optimizes through world-model reward or utility gradients.

We evaluate HNPO on four continuous-control tasks from Safety-Gymnasium Ji et al. (2023), using USB-RL as the shared world-model safe-RL backbone. HNPO improves late-stage task reward while maintaining a similarly low-cost regime, with larger gains on tasks exhibiting stronger reward–safety conflict. Ablations further distinguish the roles of credible negative selection, proposer-based coverage, and the negative-only reference.

Our contributions are: **Hard-negative learning for safe RL.** We use reward-competitive, near-boundary unsafe actions as state-matched supervision specifying what safe policy actions should outperform.

**Adaptive hard-negative generation.** We introduce a conservative proposer that learns from validated negatives and maintains informative negative coverage as the task policy evolves without directly optimizing through the worldmodel.

**Negative-anchored policy optimization.** We introduce HNPC, which evaluates policy and proposer actions under matched continuation and optimizes only task-policy samples relative to a negative-only utility reference.

## 2 Related Work

**Safe and model-based reinforcement learning.** Safe RL is commonly formulated as constrained policy optimization. Trust-region, primal–dual, projection, and related methods incorporate cost constraints directly into policy learning (Achiam et al., 2017; Tessler et al., 2018; Paternain et al., 2019; Yang et al., 2020; Xu et al., 2021; Liu et al., 2022; Yao et al., 2024), while safety layers, shielding, intervention, and recovery mechanisms prevent predicted unsafe actions from being executed (Alshiekh et al., 2018; Dalal et al., 2018; Thananjeyan et al., 2021; Wagener et al., 2021; Yu et al., 2022). HNPO is complementary to these approaches: instead of using unsafe behavior only through constraints or intervention, it uses selected violations as comparative supervision for the task policy.

World-model agents learn latent dynamics and optimize behavior through imagined rollouts (Hafner et al., 2020; 2021; 2025). Safe model-based RL further incorporates predicted costs, safety critics, constrained learning, or safety-aware planning (Thomas et al., 2021; Wilcox et al., 2022; Huang et al., 2024; 2025). SafeDreamer combines constrained policy learning with online safety-aware planning (Huang et al., 2024), whereas USB-RL learns a long-horizon reward–cost utility for policy learning and planning (Wang et al., 2026). We build HNPO on USB-RL while retaining its world model, utility learning, and deployment-time safety mechanism; HNPO instead changes how imagined unsafe alternatives supervise the policy.

**Learning from undesirable behavior.** Failures can provide supervision rather than being discarded. Recovery RL learns a safety critic and recovery policy from violation data (Thananjeyan et al., 2021), and related methods learn representations or auxiliary policies for recognizing undesirable behavior (Rahman et al., 2023). Offline safe RL and trajectory-level preference methods also exploit datasets containing behaviors of different safety or quality (Polosky et al., 2022; Gong et al., 2025), while counterfactual preference methods can construct safe–unsafe pairs from collected experience (Verma et al., 2026). HNPO differs in focusing on an online, evolving setting: useful negatives must remain near the current safety boundary, competitive with the current policy, and credible under the current world model. The learned proposer is used to maintain access to such negatives as the policy distribution changes.

**Relative policy optimization.** PPO stabilizes policy updates using clipped importance ratios (Schulman et al., 2017), while recent relative and preference-based objectives learn from comparisons within sampled groups or between preferred and dispreferred outcomes (Shao et al., 2024; Wu et al., 2025). HNPO adopts the idea of relative comparison but faces an asymmetric sampling problem: safe actions come from the task policy, whereas hard negatives come from a separate proposer. Pooling them into a common baseline would couple the reference to the negative sampler, and proposer actions cannot be treated as on-policy samples. HNPO therefore uses the negative group only to define a state-dependent reference, while policy gradients are applied only to validated task-policy actions.

# 3 Problem Setup

## 3.1 Constrained Reinforcement Learning

We consider a constrained Markov decision process (CMDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, c, \gamma_R, \gamma_C, B)$, where $r$ and $c \geq 0$ denote task reward and safety cost, and $B$ is the discounted-cost budget. A stochastic policy $\pi_\theta$ is optimized according to

$$\begin{aligned} \max_{\theta} \quad & J^R(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t\geq 0} \gamma_R^t r_t\right], \\ \text{s.t.} \quad & J^C(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t\geq 0} \gamma_C^t c_t\right] \leq B. \end{aligned} \tag{1}$$

HNPO does not replace this constrained objective or the deployment-time safety mechanism; it introduces an additional policy-learning signal for high-value actions near the safety boundary.

## 3.2 World-Model Utility Evaluation

We build on a latent world-model agent with latent state $z_t$ and learned dynamics

$$z_{t+1} \sim p_\phi(z_{t+1} \mid z_t, a_t), \qquad \hat{r}_t = R_\phi(z_t, a_t), \qquad \hat{c}_t = C_\phi(z_t, a_t). \tag{2}$$

Following the USB-RL backbone, let $\hat{y}_t = S_\phi(z_t, a_t)$ denote its learned reward–safety utility and $V_\xi$ the corresponding value estimator,

$$V_\xi(z_t) \approx \mathbb{E}\left[\sum_{k\geq 0} \gamma_U^k \hat{y}_{t+k} \,\middle|\, z_t\right]. \tag{3}$$

To compare different initial actions from the same state, we use the finite-horizon action-conditioned utility

$$Q_H(z_t, a_t; \pi) = \mathbb{E}\left[\sum_{k=0}^{H-1} \gamma_U^k \hat{y}_{t+k} + \gamma_U^H V_\xi(z_{t+H})\right], \tag{4}$$

where the first action is fixed to $a_t$ and subsequent actions follow $\pi$. Reward and cost returns $J_H^R$ and $J_H^C$ are defined analogously. HNPO uses this existing utility for comparison and introduces no new reward–cost scalarization.

## 3.3 Hard Negatives for Safe Policy Learning

For a latent state $z$, let $\mathcal{P}(z)$ contain task-policy actions validated as safe and $\mathcal{N}(z)$ contain credible, reward-competitive near-boundary unsafe actions. We characterize their state-wise utility gap as

$$\Gamma_\pi(z) = \max_{a^+ \in \mathcal{P}(z)} Q(z, a^+) - \max_{a^- \in \mathcal{N}(z)} Q(z, a^-), \tag{5}$$

where $Q$ denotes the backbone long-horizon utility. A positive $\Gamma_\pi(z)$ indicates that the policy group contains a safe action that outperforms the strongest retained unsafe competitor; a small or negative value identifies a state where such a preference has not yet been established.

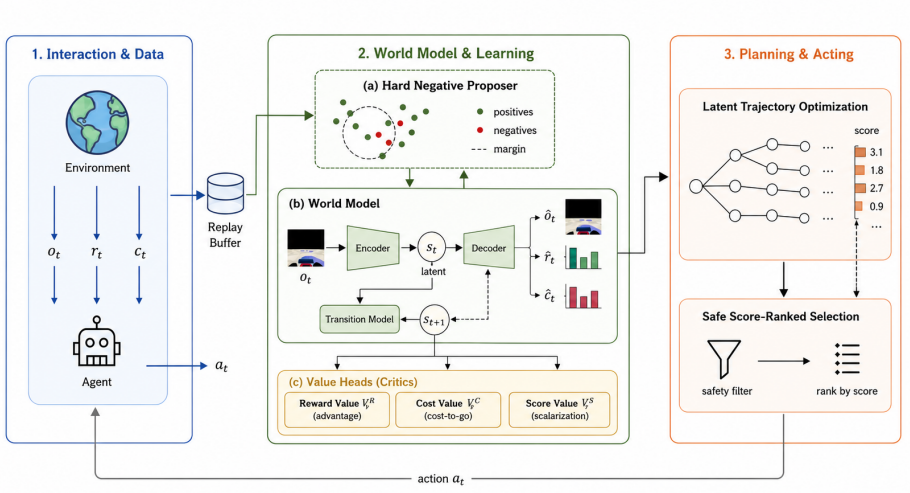



Figure 1: *Overview of HNPO.* The agent stores transitions in replay and updates the world model; (a) credible mining selects near-boundary, reward-competitive unsafe actions; (b) a conservative proposer improves their coverage; (c) matched-continuation evaluation and HNPC train the policy against a negative-only utility anchor, while the safety planner selects the executed action.

HNPO does not optimize $\Gamma_\pi$ directly. Instead, it uses informative unsafe actions to construct a state-dependent reference for policy learning. This requires identifying credible hard negatives, maintaining their coverage as the policy evolves, and using them for relative optimization despite their different sampling distribution. Section 4 addresses these three problems.

# 4 Method

## 4.1 Overview

Hard-Negative Policy Optimization (HNPO) augments a world-model-based safe RL agent with supervision from reward-competitive unsafe actions. It consists of three components. First, HNPO identifies *credible hard negatives*: violations that are near the safety boundary, competitive with the current policy, and supported by the learned model. Second, a dedicated proposer maintains access to such negatives as the task policy becomes safer. Third, Hard-Negative Policy Contrast (HNPC) evaluates policy and proposer actions from matched latent states and uses the negative group as a state-dependent reference for policy optimization. The proposer is used only during training; the backbone world model, base actor objective, and deployment-time safety mechanism remain unchanged.

## 4.2 Credible Hard-Negative Construction

**Candidate generation and robust evaluation.** HNPO maintains a hard-negative buffer

$$\mathcal{D}^- = \mathcal{D}^-_{\text{real}} \cup \mathcal{D}^-_{\text{imag}}. \tag{6}$$

The real component contains violations encountered during ordinary environment interaction; proposer actions are never deliberately executed to enlarge this set. For a posterior latent state $z$, imagined candidates are obtained from replay actions at nearby states, samples from a slowly updated proposer, and local perturbations of task-policy actions. We denote the resulting candidate pool by $\mathcal{C}(z)$.

A candidate that appears attractive under a learned model is not necessarily useful: high predicted reward may reflect model error, and an apparent violation may simply result from uncertainty in cost prediction. We therefore evaluate each candidate through short world-model rollouts across multiple stochastic samples or model heads. Let $(\mu_R, \sigma_R)$, $(\mu_C, \sigma_C)$, and $(\mu_Q, \sigma_Q)$ denote the estimated mean and standard deviation of reward return, cost return, and long-horizon utility, respectively. We

use

$$\begin{aligned} J^R_{\mathrm{rob}}(z,a) &= \mu_R(z,a) - \kappa_{\mathrm{val}}\sigma_R(z,a), \\ Q_{\mathrm{rob}}(z,a) &= \mu_Q(z,a) - \kappa_{\mathrm{val}}\sigma_Q(z,a), \\ \mathrm{LCB}_C(z,a) &= \mu_C(z,a) - \kappa_C\sigma_C(z,a), \\ \mathrm{UCB}_C(z,a) &= \mu_C(z,a) + \kappa_C\sigma_C(z,a). \end{aligned} \tag{7}$$

Reward and utility use the same uncertainty coefficient in our implementation, denoted compactly by $\kappa_{\mathrm{val}}$. We keep $\kappa_C$ separate because cost uncertainty determines whether an action can be identified as a credible violation.

**Boundary, competitiveness, and credibility.** Not every unsafe action is an informative negative. HNPO retains candidates according to three properties.

*Boundary.* A hard negative should be confidently unsafe but remain close to the constraint boundary:

$$\mathrm{LCB}_C(z,a) > B + \delta_{\mathrm{unsafe}}, \qquad \mathrm{UCB}_C(z,a) \le B + \Delta_C. \tag{8}$$

The first condition prevents uncertain candidates from being labeled unsafe solely because of predictive noise, while the second excludes severe violations that provide little information about the local reward–safety trade-off.

*Competitiveness.* Let

$$b^+(z) = \mathrm{Median}_{a^+ \in \mathcal{P}(z)}\, Q_{\mathrm{rob}}(z,a^+) \tag{9}$$

denote the utility center of validated task-policy actions, and define

$$\tau_R(z) = \mathrm{Median}_{a^+ \in \mathcal{P}(z)}\, J^R_{\mathrm{rob}}(z,a^+) - \delta_R. \tag{10}$$

A candidate must satisfy

$$J^R_{\mathrm{rob}}(z,a) \ge \tau_R(z), \qquad \left|Q_{\mathrm{rob}}(z,a) - b^+(z)\right| \le \delta_Q. \tag{11}$$

The reward condition removes unsafe actions that are already dominated in task performance. The utility window further focuses learning on negatives that are genuine competitors to the current policy, excluding both trivial failures and candidates whose predicted value lies implausibly far above the current policy range.

*Credibility.* Finally, we require

$$D_{\mathrm{sup}}(z,a) \le \delta_{\mathrm{sup}}, \tag{12}$$

where $D_{\mathrm{sup}}$ measures behavioral support relative to replay actions at similar latent states. Together with the uncertainty-adjusted estimates above, this condition reduces reliance on actions far outside the region where the world model has training support.

We summarize these requirements as

$$\mathcal{N}(z) = \{a \in \mathcal{C}(z) \mid \mathrm{Boundary}(z,a), \mathrm{Competitive}(z,a), \mathrm{Credible}(z,a)\}\,. \tag{13}$$

Thus, a hard negative is not simply an action whose predicted cost exceeds the budget. It is a credible near-boundary violation that remains a plausible competitor to the current policy.

Because the world model, critics, and policy evolve throughout training, retained negatives can become stale. We therefore periodically re-evaluate both real and imagined negatives and remove samples that no longer satisfy Eqs. (8)–(12). The buffer also preserves a minimum fraction of real violations to keep proposer learning grounded in observed behavior. Priority functions and buffer-management details are provided in the appendix.

### 4.3 Conservative Hard-Negative Proposer

Hard-negative supervision becomes progressively harder to sustain as the task policy improves. Violations that are simultaneously near the safety boundary and competitive in task value become increasingly rare under the policy and replay distributions. Relying only on previously observed failures or local perturbations can therefore reduce negative coverage precisely when the policy reaches more difficult parts of the reward–safety frontier.

To maintain access to informative negatives, HNPO learns a dedicated conditional action distribution

$$q_\psi(a \mid z), \tag{14}$$

which proposes candidate hard negatives near states visited by the current policy. It uses the same action parameterization as the task policy but has independent parameters. Importantly, its objective is not to maximize cost or predicted reward; it adapts toward actions that repeatedly satisfy the hard-negative criteria in Section 4.2.

**Grounding in validated failures.** The proposer is first trained from the hard-negative buffer through weighted behavior cloning:

$$\mathcal{L}_{\mathrm{BC}}(\psi) = -\mathbb{E}_{d^- \sim \mathcal{D}^-}\left[\bar{p}(d^-) \log q_\psi(a^- \mid z)\right], \tag{15}$$

where $\bar{p}(d^-)$ is the normalized and clipped priority of the negative sample. Real violations receive greater weight during initialization so that the initial proposal distribution is grounded primarily in observed behavior rather than world-model extrapolation.

**Validated elite distillation.** Behavior cloning alone reproduces previously discovered negatives but does not adapt sufficiently as the task policy visits new states. After warm-up, HNPO therefore maintains a slowly updated target proposer $q_{\bar{\psi}}$. At a latent state $z$, we draw

$$\tilde{a}_j^- \sim q_{\bar{\psi}}(\cdot \mid z) \tag{16}$$

and evaluate these candidates using the same hard-negative criteria. Only valid candidates are retained, and the highest-priority survivors form an elite set $\mathcal{E}(z)$.

The online proposer is refined by distilling these validated elites:

$$\begin{aligned}\mathcal{L}_{\mathrm{prop}}(\psi) = & -\mathbb{E}_{z,a \sim \mathcal{E}(z)}\left[\bar{p}(z,a) \log q_\psi(a \mid z)\right] \\ & + \beta_{\mathrm{KL}} D_{\mathrm{KL}}\left(q_\psi(\cdot \mid z) \| q_{\mathrm{ref}}(\cdot \mid z)\right),\end{aligned} \tag{17}$$

where $q_{\mathrm{ref}}$ is a frozen or slowly updated behavior-cloning reference. The KL term limits rapid drift away from the behavior-supported negative distribution.

A key design choice is that candidate evaluation is detached from the proposer optimization graph. The world model determines which proposals survive, but predicted reward or utility is not differentiated through the model into $q_\psi$. This reduces the incentive to exploit errors in the learned dynamics while still allowing the proposal distribution to track useful negative regions.

The slow proposer is updated by exponential moving average,

$$\bar{\psi} \leftarrow \tau_\psi \bar{\psi} + (1 - \tau_\psi)\psi, \tag{18}$$

which reduces feedback between newly proposed actions and subsequent updates. Detailed update frequencies and stability rules are deferred to the appendix.

### 4.4 Hard-Negative Policy Contrast

The proposer supplies informative unsafe competitors, but its actions cannot be inserted directly into an on-policy objective because they are sampled from $q_{\bar{\psi}}$ rather than the task policy. HNPC therefore separates the distribution used to define the comparison target from the distribution used to update the policy.

**State-matched counterfactual evaluation.** At a selected latent state $z_t$, we sample candidate actions from a frozen copy of the task policy,

$$a_{t,i}^+ \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid z_t). \tag{19}$$

Only candidates satisfying

$$\mathrm{UCB}_C(z_t, a_{t,i}^+) \le B - \delta_{\mathrm{safe}} \tag{20}$$

are retained, giving the validated policy group $\mathcal{P}_t = \{a_{t,i}^+\}_{i=1}^{K_t}$. Thus, being sampled from the task policy is not by itself sufficient for positive supervision. In parallel, candidate negatives are sampled from the slow proposer and revalidated according to Eqs. (8)–(12), yielding $\mathcal{N}_t = \{a_{t,j}^-\}_{j=1}^{M_t}$.

Comparing independently generated trajectories would confound the initial action with differences in later behavior. HNPO instead uses a matched continuation: every candidate starts from the same $z_t$, only the first action changes, and subsequent actions follow the same frozen policy,

$$a_{t+k} \sim \pi_{\theta_{\text{old}}}(\cdot \mid z_{t+k}), \qquad k \geq 1. \tag{21}$$

Each candidate is therefore evaluated by $Q_H(z_t, a_t; \pi_{\theta_{\text{old}}})$ from Eq. (4), with repeated model evaluations yielding $Q_{\text{rob}}$. This comparison isolates, as far as possible within the learned model, the long-horizon consequence of changing the initial action while holding the continuation policy fixed.

**Negative-only utility reference.** A straightforward group-relative objective could pool $\mathcal{P}_t$ and $\mathcal{N}_t$ and center their utilities around a shared statistic. However, the two groups are generated by different distributions, and the number and difficulty of valid negatives can vary across states. A pooled reference would therefore move with the negative sampler.

HNPO instead constructs the comparison reference from the negative group alone. Let $\omega^-_{t,j} \geq 0$ assign greater weight to negatives that are confidently predicted, close to the safety boundary, and competitive with the current policy. After normalization,

$$\bar{\omega}^-_{t,j} = \frac{\omega^-_{t,j}}{\sum_{\ell=1}^{M_t} \omega^-_{t,\ell}}, \tag{22}$$

we define

$$b^-_t = \text{sg}\left[\sum_{j=1}^{M_t} \bar{\omega}^-_{t,j} Q_{\text{rob}}(z_t, a^-_{t,j})\right], \tag{23}$$

where $\text{sg}[\cdot]$ denotes stop-gradient. The anchor therefore summarizes the utility attained by credible unsafe competitors at $z_t$, rather than the relative ranking of the policy group itself.

**Negative-relative advantage.** For each validated task-policy action, HNPO defines

$$A^{\text{HNPC}}_{t,i} = \text{clip}\left(\text{sg}\left[\frac{Q_{\text{rob}}(z_t, a^+_{t,i}) - b^-_t}{s_t} - m\right], -A_{\max}, A_{\max}\right), \tag{24}$$

where $s_t$ is a robust state-wise utility scale and $m > 0$ is a dimensionless target margin. A policy action receives a positive auxiliary target only when its utility exceeds the negative reference by the desired margin.

We normalize by scale but deliberately avoid state-wise centering. To see why, let

$$\widetilde{A}_{t,i} = \frac{Q^+_{t,i} - b^-_t}{s_t} - m.$$

Centering these values within the positive group gives

$$\widetilde{A}_{t,i} - \text{Median}_k\, \widetilde{A}_{t,k} = \frac{Q^+_{t,i} - \text{Median}_k\, Q^+_{t,k}}{s_t}, \tag{25}$$

so the shared negative reference $b^-_t$ and margin $m$ both cancel. The resulting objective would rank safe actions only against one another, rather than asking whether they outperform the unsafe competitors. We therefore retain scale normalization without centering.

**Proximal policy update.** Because positive actions are sampled from the frozen task policy, their importance ratios are well defined:

$$\rho_{t,i}(\theta) = \frac{\pi_\theta(a^+_{t,i} \mid z_t)}{\pi_{\theta_{\text{old}}}(a^+_{t,i} \mid z_t)}. \tag{26}$$

HNPO applies the clipped proximal objective

$$J_{\text{HNPC}}(\theta) = \mathbb{E}_t\left[g_t \frac{1}{K_t} \sum_{i=1}^{K_t} \min\Big(\rho_{t,i} A^{\text{HNPC}}_{t,i}, \text{clip}(\rho_{t,i}, 1-\epsilon, 1+\epsilon) A^{\text{HNPC}}_{t,i}\Big)\right], \tag{27}$$

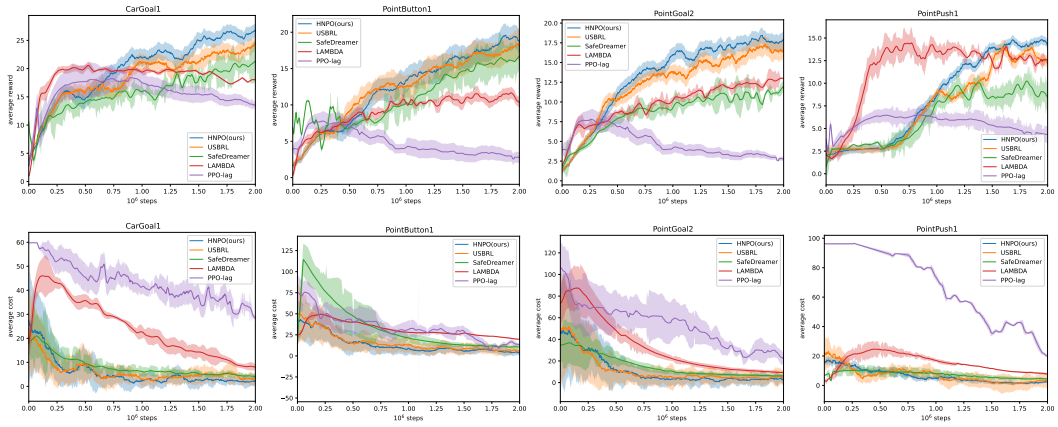



Figure 2: *Results on Safety-Gymnasium.* HNPO achieves higher task reward (top) while maintaining low episodic cost (bottom). Shaded areas indicate 95% confidence intervals over five seeds.

where $g_t$ reduces the contribution of states for which the retained negative evidence is insufficient or unreliable. The full gate definition is given in the appendix.

Crucially, only validated task-policy actions enter the importance-ratio objective. Proposer actions determine $b_t^-$ but are never treated as samples from $\pi_{\theta_{\text{old}}}$: *negatives define the reference; policy samples receive the gradient*.

Finally, HNPC augments rather than replaces the backbone actor objective:

$$\mathcal{L}_{\text{actor}} = \mathcal{L}_{\text{base}} - \alpha_{\text{HNPC}}(n) J_{\text{HNPC}}(\theta) + \beta_\pi \mathbb{E}_z \left[ D_{\text{KL}} \left( \pi_\theta(\cdot \mid z) \| \pi_{\theta_{\text{old}}}(\cdot \mid z) \right) \right]. \tag{28}$$

Here $\mathcal{L}_{\text{base}}$ is the original USB-RL actor loss. The HNPC coefficient is introduced gradually as reliable hard-negative supervision becomes available. HNPO therefore retains the backbone reward–safety utility learning and deployment-time safety mechanism while adding a distinct learning signal that trains safe policy actions against attractive unsafe alternatives.

## 5 Experiments

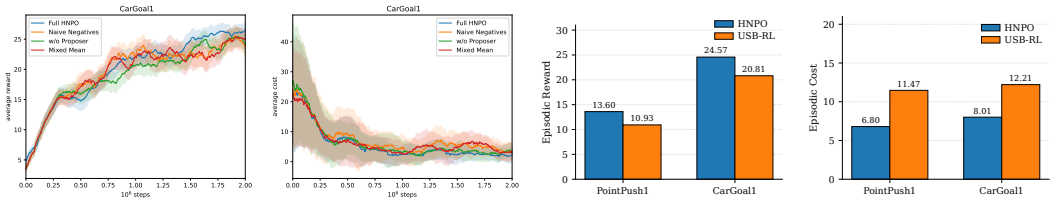
We evaluate whether HNPO improves the reward–safety trade-off of its safe-RL backbone, whether the improvement is reflected in the learned actor itself, and which components are responsible for the gains.

### 5.1 Experimental Setup

We evaluate on four continuous-control tasks from Safety-Gymnasium (Ji et al., 2023): PointGoal2, PointButton1, PointPush1, and CarGoal1. We compare with PPO-Lag, LAMBDA, SafeDreamer, and USB-RL. USB-RL is also the backbone of HNPO: the two methods share the world model, critics, base actor objective, and deployment-time safety planner, while HNPO adds only the hard-negative learning branch. All methods use the same training budget of 8 million environment steps. Results are reported over five random seeds; HNPO and USB-RL use paired seeds and identical initial replay data. Learning curves show the mean across seeds with 95% confidence intervals. Further implementation details are provided in the appendix.

### 5.2 Reward–Safety Performance

Figure 2 compares reward and episodic cost on all four tasks. HNPO improves late-stage reward over its USB-RL backbone while maintaining a similarly low-cost regime. The gains are most pronounced on PointPush1 and CarGoal1, where effective task behavior more frequently occurs near unsafe regions, whereas the improvement is smaller on the milder PointGoal2 task. This pattern is consistent with the intended role of hard negatives: they provide the strongest additional supervision when high-value unsafe alternatives remain competitive with the current policy.




(a) Ablation Reward (b) Ablation Cost (c) Raw Actor Reward (d) Raw Actor Cost

Figure 3: **Policy and component analysis.** (a–b) Ablations on CARGOAL1 isolate negative quality, proposer-based coverage, and the negative-only reference. (c–d) Raw-actor evaluation on POINTPUSH1 and CARGOAL1 disables the shared deployment-time safety planner without additional training; bars report mean performance over the final 100 evaluation points.

### 5.3 POLICY AND COMPONENT ANALYSIS

We next examine whether HNPO improves the task actor itself and whether its three main design choices contribute complementary benefits. Figure 3 summarizes both analyses.

**Raw actor.** Because HNPO and USB-RL share the same deployment-time safety planner, we re-evaluate their trained checkpoints with the planner disabled and without additional training. On POINTPUSH1, HNPO increases raw-actor reward from 10.93 to 13.60 while reducing episodic cost from 11.47 to 6.80. On CARGOAL1, reward increases from 20.81 to 24.57 while cost decreases from 12.21 to 8.01. These results show that the observed gains are reflected in the learned actor itself rather than arising solely from deployment-time correction. Notably, these are also the two tasks where HNPO shows its clearest planner-assisted gains in Figure 2, linking the main improvement to stronger policy behavior in the settings where hard-negative supervision is most relevant.

**Ablations.** We ablate the three main components on CARGOAL1. *Naive Negatives* removes the credibility-aware negative construction and exhibits less stable late-stage behavior; *w/o Proposer* retains credible negatives but loses adaptive negative coverage, leading to slower and lower reward improvement; and *Mixed Mean* replaces the negative-only anchor with a pooled policy–negative reference, producing more oscillatory optimization. The three variants respectively isolate the roles of negative quality, adaptive coverage, and reference stability.

Their distinct failure modes also suggest that these components address different bottlenecks rather than providing redundant regularization.

## 6 CONCLUSION

We introduced Hard-Negative Policy Optimization (HNPO), a world-model-based safe RL method that uses reward-competitive, near-boundary unsafe actions as hard negatives for policy learning. HNPO combines credible hard-negative construction, a conservative proposer that maintains negative coverage as the policy evolves, and Hard-Negative Policy Contrast, which trains validated policy actions to outperform a negative-only utility reference under matched continuation. Across four Safety-Gymnasium tasks, HNPO improves task reward over its USB-RL backbone while maintaining low episodic cost, with larger gains when reward and safety are more strongly coupled. Ablations further support the complementary roles of negative quality, proposer-based coverage, and the negative-only reference.

HNPO remains dependent on the accuracy and uncertainty calibration of the learned world model, and its current counterfactual construction varies only the initial action rather than an action sequence. Moreover, HNPO improves policy learning rather than replacing the deployment-time safety mechanism. Extending hard negatives to multi-step behaviors and improving uncertainty calibration are promising directions for future work.

## REFERENCES


Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. Pmlr, 2017.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces, 2018.

Ze Gong, Akshat Kumar, and Pradeep Varakantham. Offline safe reinforcement learning using trajectory classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16880–16887, 2025.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *9th International Conference on Learning Representations, ICLR 2021*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Dongchi Huang, WANG Jiaqi, Yang Li, Chunhe Xia, Tianle Zhang, and Kaige Zhang. Pigdreamer: Privileged information guided world models for safe partially observable reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025.

Weidong Huang, Jiaming Ji, Chunhe Xia, Borong Zhang, and Yaodong Yang. Safedreamer: Safe reinforcement learning with world models. In *International Conference on Learning Representations*, volume 2024, pp. 53839–53869, 2024.

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.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety gymnasium: A unified safe reinforcement learning benchmark. *Advances in Neural Information Processing Systems*, 36:18964–18993, 2023.

Zuxin Liu, Zhepeng Cen, Vladislav Isenbaev, Wei Liu, Steven Wu, Bo Li, and Ding Zhao. Constrained variational policy optimization for safe reinforcement learning. In *International Conference on Machine Learning*, pp. 13644–13668. PMLR, 2022.

Santiago Paternain, Luiz F. O. Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 7555–7565, 2019.

Nicholas Polosky, Bruno C Da Silva, Madalina Fiterau, and Jithin Jagannath. Constrained offline policy optimization. In *International Conference on Machine Learning*, pp. 17801–17810. PMLR, 2022.

Md Asifur Rahman, Tongtong Liu, and Sarra M Alqahtani. Adversarial behavior exclusion for safe reinforcement learning. In *IJCAI*, pp. 483–491, 2023.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning.(2019). *URL https://cdn. openai. com/safexp-short. pdf*, 3:14–28, 2019.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In *International conference on machine learning*, pp. 9133–9143. PMLR, 2020.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization, 2018.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6 (3):4915–4922, 2021.

Garrett Thomas, Yuping Luo, and Tengyu Ma. Safe reinforcement learning by imagining the near future. *Advances in Neural Information Processing Systems*, 34:13859–13869, 2021.

Richa Verma, Bavish Kulur, Sanjay Chawla, and Balaraman Ravindran. Prefine: Preference-based implicit reward and cost fine-tuning for safety alignment. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1004–1012, 2026.

Nolan C Wagener, Byron Boots, and Ching-An Cheng. Safe reinforcement learning using advantage-based intervention. In *International Conference on Machine Learning*, pp. 10630–10640. PMLR, 2021.

Yuetian Wang, Dianxi Shi, Yuanze Wang, Huanhuan Yang, and Shiming Song. Learning reward–cost balance in safe RL via score-based world models. In *International Conference on Machine Learning*. PMLR, 2026.

Albert Wilcox, Ashwin Balakrishna, Brijen Thananjeyan, Joseph E Gonzalez, and Ken Goldberg. Ls3: Latent space safe sets for long-horizon visuomotor control of sparse reward iterative tasks. In *Conference on Robot Learning*, pp. 959–969. PMLR, 2022.

Xiao-Kun Wu, Min Chen, Wanyi Li, Rui Wang, Limeng Lu, Jia Liu, Kai Hwang, Yixue Hao, Yanru Pan, Qingguo Meng, et al. Llm fine-tuning: Concepts, opportunities, and challenges. *Big Data and Cognitive Computing*, 9(4):87, 2025.

Tengyu Xu, Yingbin Liang, and Guanghui Lan. Crpo: A new approach for safe reinforcement learning with convergence guarantee. In *International Conference on Machine Learning*, pp. 11480–11491. PMLR, 2021.

Tsung Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Projection-based constrained policy optimization. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

Yihang Yao, Zuxin Liu, Zhepeng Cen, Peide Huang, Tingnan Zhang, Wenhao Yu, and Ding Zhao. Gradient shaping for multi-constraint safe reinforcement learning. In *6th annual learning for dynamics & control conference*, pp. 25–39. PMLR, 2024.

Dongjie Yu, Haitong Ma, Shengbo Li, and Jianyu Chen. Reachability constrained reinforcement learning. In *International conference on machine learning*, pp. 25636–25655. PMLR, 2022.