Latent Q-Barrier Shielding for Safe In-Context Reinforcement Learning

Minjae Kwon[*†]
University of Virginia
mjkwon@virginia.edu

Amir Moeini[*†]
University of Virginia
amoeini@virginia.edu

Shangtong Zhang
University of Virginia
shangtong@virginia.edu

Lu Feng
University of Virginia
lu.feng@virginia.edu

Abstract

Safe in-context reinforcement learning (ICRL) adapts online from interaction history without test-time parameter updates while controlling episode cost under a safety budget. Under out-of-distribution (OOD) deployment shifts, pretraining-only safe ICRL can give poor reward-safety tradeoffs because the remaining budget affects behavior only through frozen policy conditioning, not an explicit action-level check against predicted future cost. We propose a latent Q-Barrier shield that learns a context representation, latent dynamics, and an ensemble cost critic before deployment. Without parameter updates, the shield infers context from history and filters or softly reweights candidate actions using the remaining budget and predicted future cost. We prove a conditional one-step barrier-margin bound over realized candidate sets, with explicit Bellman and latent-prediction error terms. Across five safe ICRL benchmarks, the shield improves deployment-time reward-safety tradeoffs over a strong safe-ICRL baseline: after a short context window, it achieves higher return in four of five benchmarks while matching or lowering average episode cost in all five. Code is available at q-barrier.

1 Introduction

Reinforcement learning (RL) [57] agents deployed in the real world must manage safety constraints such as resource limits, physical damage, or constraint violations during deployment. Constrained Markov decision processes (CMDPs) [4] formalize this requirement by assigning each transition both a reward and a cost and constraining expected cumulative cost under a user-specified budget. Most safe RL methods enforce this constraint during training, for example through Lagrangian penalties or constrained policy optimization [1, 47]. However, training-time constraint satisfaction need not generalize under deployment shifts, as demonstrated by safety violations on unseen simulated patients in diabetes control [32].

In-context reinforcement learning (ICRL) provides a different adaptation mechanism: a pretrained agent adapts to new tasks at test time without parameter updates by conditioning its policy on an expanding interaction history [14, 67, 34, 55, 43, 37]. Recent work extends this idea to safe ICRL. In particular, Moeini et al. [41] formulate safe ICRL as a constrained problem across test episodes and propose SCARED, a Lagrangian pretraining method whose fixed-point analysis characterizes budget-respecting policies under its assumptions.

[1]* Equal contribution.
[2]† Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, existing safe ICRL methods, such as SCARED, mainly encode safety through pretraining objectives or cost-conditioning, rather than through an explicit action-level check against the remaining budget. A frozen ICRL policy may still assign probability to actions whose predicted future cost exceeds the remaining budget, especially in generalization settings where the task and safety structure must be inferred from context. We add a budget-aware shielding layer that evaluates candidate actions against the remaining budget using a latent cost barrier and a pessimistic cost-to-go critic. The shield then filters or softly reweights the policy distribution without deployment-time parameter updates, turning learned safety preferences into an explicit action-level check.

This motivates the central question of the paper: *can a learned barrier improve a frozen safe-ICRL policy's reward-safety tradeoff by turning the remaining budget into an explicit action-selection constraint?* We answer this question through a Q-Barrier shielding method, a margin-based analysis, and empirical evaluation on deployment generalization tasks. In summary, our contributions are:

- **Barrier-based runtime shielding for safe ICRL deployment.** We introduce a deployment-time shield that reweights a frozen policy using the remaining safety budget and a learned cost-to-go estimate. The shield uses latent representations, latent dynamics, and an ensemble cost critic learned during training, with all parameters fixed at deployment.
- **Conditional barrier-margin bound.** We prove that for any selected action, the next-step latent-budget margin is lower-bounded by the current margin minus the critic's Bellman upper-bound error and a latent-prediction error term. The bound is conditional on the realized candidate sets and error terms, and is not a safety guarantee for the deployed shield.
- **Improved in-context adaptation reward-safety tradeoffs.** Q-Barrier improves early return and lowers cost after a short context window in four of five benchmarks, with one environment showing a conservative lower-cost/lower-return tradeoff. In budget sweeps, it achieves the highest cumulative return in four of five environments and satisfies the budget across most levels, with violations mostly occurring at near-zero budgets, where even small costs can exceed the budget.

2 Background

Markov Decision Processes. We model the interaction with an environment as a finite-horizon Markov decision process (MDP [45]). An MDP is defined by a state space $\mathcal{S}$, an action space $\mathcal{A}$, a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, a transition kernel $p(\cdot \mid s, a)$, where for each state-action pair (s, a), $p(\cdot \mid s, a)$ denotes a probability distribution over next states in $\mathcal{S}$, an initial state distribution p_0 over $\mathcal{S}$, and a horizon length T. We denote by T_k the terminal timestep of episode k. An agent follows a stochastic policy π, modeled as a Markov kernel from states to actions: for each $s \in \mathcal{S}$, $\pi(\cdot \mid s)$ is a probability distribution over $\mathcal{A}$. When $\mathcal{A}$ is discrete, $\pi(a \mid s)$ denotes the probability mass of action a. When $\mathcal{A}$ is continuous, $\pi(a \mid s)$ denotes the action density, when it exists, which integrates to one. After starting from $S_0 \sim p_0$, at time step t, the agent samples $A_t \sim \pi(\cdot \mid S_t)$, observes reward $R_{t+1} \doteq r(S_t, A_t)$, and transitions to $S_{t+1} \sim p(\cdot \mid S_t, A_t)$. We use k to index episodes and t to index time steps within an episode. For any episode τ, we define its return as $G(\tau) \doteq \sum_{t=1}^{T} R_t$. The performance of a policy π is measured by the expected total rewards $J(\pi) \doteq \mathbb{E}_{\tau\sim\pi}[G(\tau)]$.

Constrained Markov Decision Processes. In addition to the reward function r, CMDPs [4] involve a cost function $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$ with an associated user-given budget δ. At each time step t, after taking an action A_t at a state S_t, the agent receives a cost $C_{t+1} \doteq c(S_t, A_t)$. An episode in a CMDP therefore becomes $\tau = (S_0, A_0, R_1, C_1, S_1, \ldots, S_{T-1}, A_{T-1}, R_T, C_T)$. We denote the total cost of an episode τ by $G_c(\tau) \doteq \sum_{t=1}^{T} C_t$, which should, in expectation, remain below the budget δ. The return-to-go (RTG) is $G_t(\tau) \doteq \sum_{i=t+1}^{T} R_i$ and the cost-to-go (CTG) is $G_{c,t}(\tau) \doteq \sum_{i=t+1}^{T} C_i$. Given δ, we define the remaining cost budget at time t as $B_t \doteq \delta - \sum_{i=1}^{t} C_i,\ B_0 = \delta$.

In-Context Reinforcement Learning. ICRL [40] trains an agent across a distribution of tasks so that, at test time, it can adapt without parameter updates by conditioning on an expanding interaction history. Let θ_* denote the pretrained parameters. During deployment, θ_* is fixed and the policy is written as $\pi_{\theta_*}(\cdot \mid S_t^k, H_t^k)$, where H_t^k is the context available at time t in episode k. A context consists of previous episodes $\{\tau_j\}_{j=1,\ldots k-1}$ and the current episode prefix τ_k^t, i.e., $H_t^k \doteq (\tau_1, \ldots, \tau_{k-1}, \tau_k^t)$. In safe ICRL [41], each task is a CMDP, so the context includes both rewards and costs: $\tau_k^t \doteq (S_0^k, A_0^k, R_1^k, C_1^k, S_1^k, \ldots, S_{t-1}^k, A_{t-1}^k, R_t^k, C_t^k)$. We write the remaining episode budget as $B_t^k \doteq \delta - \sum_{i=1}^{t} C_i^k,\ B_0^k = \delta$. Adaptation occurs as $\pi_{\theta_*}(\cdot \mid S_t^k, H_t^k)$ changes with the growing context, while the policy parameters remain fixed.

2.1 Problem Formulation

We formulate safe ICRL deployment over K test episodes. All parameters are learned before deployment and fixed at test time. The deployed shielded policy π uses the current state S_t^k, context H_t^k, and remaining budget B_t^k for action selection. Our objective is:

$$\max_\pi \ \sum_{k=1}^K \mathbb{E}_\pi[G(\tau_k)] \quad \text{s.t.} \quad \tfrac{1}{K}\sum_{k=1}^K \mathbb{E}_\pi[G_c(\tau_k)] \le \delta. \tag{1}$$

The shield changes actions online to improve reward-safety tradeoffs during deployment. Theorem 1 gives a conditional bound on how the barrier margin propagates one step forward under the learned critic. Proposition 1 extends it to an episode-level budget bound with explicit error terms.

3 Approach

We present the method in three parts. First, we define the latent state and learned safety model used for runtime safety estimation. Second, we introduce a Q-function barrier that uses the learned cost-to-go estimate and the remaining safety budget to filter actions online. Third, we describe how the latent representation, dynamics model, and cost critic are trained to support shielding at deployment.

3.1 Latent State and Projection Spaces

The shield uses two views of the same in-context state: a policy view for proposing actions and a world view for evaluating safety. Let $\mathcal{H}$ denote the context space and let $E_\phi : \mathcal{H} \times \mathcal{S} \to \mathcal{Z} \subseteq \mathbb{R}^{d_z}$ be the shared encoder. Given context H_t^k and state S_t^k, the shared latent state is $Z_t^k \doteq E_\phi(H_t^k, S_t^k) \in \mathcal{Z}$. From this shared latent, we define two projection heads with a common projected dimension d_m: $g_\omega^{\text{world}} : \mathcal{Z} \to \mathcal{Z}^w \subseteq \mathbb{R}^{d_m}$, $g_\psi^{\text{policy}} : \mathcal{Z} \to \mathcal{Z}^p \subseteq \mathbb{R}^{d_m}$. The corresponding projected latents are $Z_t^{w,k} \doteq g_\omega^{\text{world}}(Z_t^k)$, $Z_t^{p,k} \doteq g_\psi^{\text{policy}}(Z_t^k)$. The world-projected latent $Z_t^{w,k}$ is the safety and modeling space used by the latent dynamics model, cost critic, and runtime barrier. The policy-projected latent $Z_t^{p,k}$ is the policy-side representation used by the base policy to form its action distribution. The shield proposes actions through the policy view and evaluates their budget risk through the world view. On $\mathcal{Z}^w$, we learn a probabilistic latent transition model $p_z(\cdot \mid Z_t^{w,k}, A_t^k)$ and an ensemble of cost critics $\{\hat{Q}_{C,i}\}_{i=1}^M$. The barrier uses the pessimistic aggregate $\hat{Q}_C^+(Z, A) \doteq \max_{i\in\{1,\dots,M\}} \hat{Q}_{C,i}(Z, A)$, which reduces the risk of underestimating future cost. We write $f_z(Z_t^{w,k}, A_t^k) \doteq \mathbb{E}_{p_z}[Z_{t+1}^{w,k} \mid Z_t^{w,k}, A_t^k]$ for the predictive mean. The critic $\hat{Q}_C^+$ estimates cost-to-go in the same units as the remaining budget; f_z supports representation learning and analysis, not runtime action selection. All modules are learned during training and frozen at test time.

3.2 Q-Function Barrier and Shield Policy

The shield maps a base policy into a budget-aware action distribution. During training, the base policy π_θ is updated. At each decision time t in episode k, the shield evaluates a finite candidate action set $\mathcal{A}_{t,k}^{\text{cand}} \subseteq \mathcal{A}$. In discrete-action settings, $\mathcal{A}_{t,k}^{\text{cand}} = \mathcal{A}$; in continuous-action settings, it is the finite set of candidate actions sampled from π_θ. At state $Z_t^{w,k}$ with remaining budget B_t^k, actions whose predicted cost-to-go exceeds the available budget are discouraged. This motivates a learned barrier signal derived from the cost critic: the value is positive when the remaining budget suffices to cover the predicted cost-to-go and negative when the action is predicted to overspend the budget. Because the barrier depends on the current latent state and learned cost-to-go estimate, it adapts online as context accumulates, without changing any parameters.

Definition 1 (Q-Function Barrier)**.** *For $Z \in \mathcal{Z}^w$, define the candidate-set cost value $\hat{V}_{C,t}^{+,k}(Z) \doteq \min_{A\in\mathcal{A}_{t,k}^{\text{cand}}} \hat{Q}_C^+(Z, A)$. The state- and state-action barriers are*

$$b_{V,t}^k(Z_t^{w,k}, B_t^k) \doteq B_t^k - \hat{V}_{C,t}^{+,k}(Z_t^{w,k}), \qquad b_{Q,t}^k(Z_t^{w,k}, B_t^k, A_t^k) \doteq B_t^k - \hat{Q}_C^+(Z_t^{w,k}, A_t^k). \tag{2}$$

The state barrier $b_{V,t}^k \ge 0$ means that at least one candidate action has predicted cost-to-go no larger than the remaining budget. The action barrier $b_{Q,t}^k$ evaluates whether a specific action is predicted to

stay within the remaining budget under $\hat{Q}_C^+$. Because this margin is estimated by learned models, we use it as a soft risk signal rather than a hard certificate. The primary variant is a *soft shield* that reweights π_θ over $\mathcal{A}_{t,k}^{\text{cand}}$, preserving policy support while penalizing candidates whose predicted cost exceeds the remaining budget. For notational convention, we use $[x]_+ \doteq \max\{x, 0\}$. For each candidate action, define the soft-shield score as

$$w_t(A) \doteq \rho_t(A) \exp\left(- \left[-b_{Q,t}^k(Z_t^{w,k}, B_t^k, A) \right]_+ \right), \quad \text{where}$$

$$\rho_t(A) = \begin{cases} \pi_\theta(A \mid Z_t^{p,k}), & \text{if the discrete action space is enumerated,} \\ 1, & \text{if } A \in \mathcal{A}_{t,k}^{\text{cand}} \text{ is sampled from } \pi_\theta(\cdot \mid Z_t^{p,k}). \end{cases}$$

The soft-shield distribution over the finite candidate set is

$$\pi_{\text{soft}}(A \mid S_t^k, H_t^k, B_t^k) = \frac{w_t(A)}{\sum_{A' \in \mathcal{A}_{t,k}^{\text{cand}}} w_t(A')}, \quad A \in \mathcal{A}_{t,k}^{\text{cand}}. \tag{3}$$

Remark. We use the soft shield as the default because it preserves support over sampled candidates and avoids sensitive hard-threshold decisions. Hard filtering can be sensitive to model or critic error: when no candidate is considered as feasible, it falls back to the lowest-predicted-cost action. Empirically, both variants achieve similar reward–safety profiles in most environments (Appendix D). Unlike hard filtering in Appendix C, it can still choose an action with $b_{Q,t}^k < 0$. In continuous control, candidates are already drawn from the base policy, so we omit an additional policy-density factor in the finite-candidate weights to avoid double-counting the proposal.

3.3 Training Objective and Asymmetric Gradient Routing

The shield uses $Z_t^{p,k}$ for action proposals and $Z_t^{w,k}$ for latent prediction and cost estimation. All modules are trained before deployment and then frozen. To reduce interference between policy learning and world modeling, we use asymmetric gradient routing: the world-model loss updates the shared encoder, while stop-gradient alignment losses coordinate the projection heads without directly updating the encoder.

Representation-shaping loss. We train an auxiliary one-step world model on the world-projected latent space. The loss is $\mathcal{L}_{\text{wm}} = \mathbb{E}\left[-d_m^{-1} \log p_z\left(Z_{t+1}^{w,k} \mid Z_t^{w,k}, A_t^k\right)\right] + \mathbb{E}\left[(\hat{R}_{t+1}^k - R_{t+1}^k)^2\right] + \mathbb{E}[\sum_{j=1}^2 (\hat{C}_{j,t+1}^k - C_{t+1}^k)^2]$, where d_m is the common output dimension of the policy and world projection heads. This loss trains the encoder and world projection by encouraging one-step predictability of latent dynamics, reward, and cost.

Structural alignment losses. The losses $\mathcal{L}_{\text{distill}}$ and $\mathcal{L}_{\text{conj}}$ align the policy and world projections through detached encoder features: $\mathcal{L}_{\text{distill}}^{\text{sg}} = \frac{1}{d_m} \left\| g_\psi^{\text{policy}}(\text{sg}(Z_t^k)) - \text{sg}(g_\omega^{\text{world}}(Z_t^k)) \right\|_2^2$, and

$$\mathcal{L}_{\text{conj}}^{\text{sg}} = \frac{1}{d_m} \left\| \left(g_\psi^{\text{policy}}(\text{sg}(Z_{t+1}^k)) - \gamma_* g_\psi^{\text{policy}}(\text{sg}(Z_t^k)) \right) - \text{sg}\left(g_\omega^{\text{world}}(Z_{t+1}^k) - \gamma_* g_\omega^{\text{world}}(Z_t^k) \right) \right\|_2^2.$$

We use AMAGO's actor–critic backbone [17], which trains heads at multiple discount factors, and select the deployment discount $\gamma_* = 0.999$ for rollout and evaluation. The conjugacy loss uses the same coefficient to match temporal changes between the policy and world projections. We augment this backbone with SCARED's safety objectives [41]. The actor combines reward improvement, advantage-filtered behavioral cloning, and a masked cost penalty, while the critics minimize reward and cost TD errors. The principal training terms form the core objective

$$\mathcal{L}_{\text{core}} = \mathcal{L}_{\text{actor}} + \lambda_{\text{critic}} \mathcal{L}_{\text{critic}} + \lambda_{\text{wm}} \mathcal{L}_{\text{wm}} + \lambda_{\text{dist}} \mathcal{L}_{\text{distill}}^{\text{sg}} + \lambda_{\text{conj}} \mathcal{L}_{\text{conj}}^{\text{sg}}, \tag{4}$$

where sg denotes stop-gradient. Appendix H.1 details the displayed objectives, their weights, and gradient routing. Asymmetric gradient routing encourages Z^w to support latent prediction and cost estimation while preserving policy-relevant features in Z^p.

4 Barrier Margin Propagation

This section gives a decomposition of the next-step barrier value into three quantities: the current barrier margin, the one-step latent prediction error, and the Bellman upper-bound error of the learned cost critic. For any fixed next-step candidate set $\mathcal{A}^{\text{cand}}_{t+1,k}$, define $\hat{V}^{+,k}_{C,t+1}(Z) \doteq \min_{A' \in \mathcal{A}^{\text{cand}}_{t+1,k}} \hat{Q}^{+}_{C}(Z, A')$, with $\hat{V}^{+,k}_{C,T_k} = 0$. In continuous-action domains, $\mathcal{A}^{\text{cand}}_{t+1,k}$ is the finite set of candidate actions sampled by the shield, so all statements are conditional on the sampled candidate set.

We first assume that the learned latent dynamics model predicts the next latent state with small one-step error, motivated by the training objective in Section 3.3.

Assumption 1 (Approximate Latent Dynamics)**.** *Along the trajectories considered, the learned latent transition mean f_z satisfies $\left\| f_z(Z^{w,k}_t, A^k_t) - Z^{w,k}_{t+1} \right\|_2 \le \varepsilon_{\text{pred}}$ for all episodes k and timesteps t .*

We also assume that the learned cost critic changes smoothly when the latent state changes slightly.

Assumption 2 (Latent Q-Regularity)**.** *For A in the candidate action set, the pessimistic cost critic $\hat{Q}^{+}_{C} : \mathcal{Z}^w \times \mathcal{A} \to \mathbb{R}$ is uniformly Lipschitz. That is, there exists $L_Q > 0$ such that*

$$\left| \hat{Q}^{+}_{C}(Z, A) - \hat{Q}^{+}_{C}(Z', A) \right| \le L_Q \|Z - Z'\|_2 \quad \text{for all } Z, Z' \in \mathcal{Z}^w. \tag{5}$$

We next isolate the critic's one-step Bellman error as an explicit residual term.

Definition 2 (Bellman Upper-Bound Error)**.** *For transition $(Z^{w,k}_t, A^k_t, C^k_{t+1})$, define*

$$\Delta^{+,k}_{\text{bell},t} \doteq \left[C^k_{t+1} + \hat{V}^{+,k}_{C,t+1}\left(f_z(Z^{w,k}_t, A^k_t) \right) - \hat{Q}^{+}_{C}(Z^{w,k}_t, A^k_t) \right]_+ . \tag{6}$$

This value measures how much the critic violates the one-step Bellman upper bound. $\Delta^{+,k}_{\text{bell},t} = 0$ when the critic holds the one-step upper-bound condition $\hat{Q}^{+}_{C}(Z^{w,k}_t, A^k_t) \ge C^k_{t+1} + \hat{V}^{+,k}_{C,t+1}(f_z(Z^{w,k}_t, A^k_t))$. The main theorem below shows how the barrier margin propagates forward by losing Bellman error and model-prediction error.

Theorem 1 (Barrier Margin Propagation on the Latent Space)**.** *Under Assumptions 1 and 2, for any selected action A^k_t, $b^k_{V,t+1}(Z^{w,k}_{t+1}, B^k_{t+1}) \ge b^k_{Q,t}(Z^{w,k}_t, B^k_t, A^k_t) - \Delta^{+,k}_{\text{bell},t} - L_Q \varepsilon_{\text{pred}}$.*

The theorem implies an approximate one-step continuation property: if $b^k_{Q,t}(Z^{w,k}_t, B^k_t, A^k_t) \ge 0$, then $b^k_{V,t+1}(Z^{w,k}_{t+1}, B^k_{t+1}) \ge -\Delta^{+,k}_{\text{bell},t} - L_Q \varepsilon_{\text{pred}}$. Thus, the next latent-budget state has a lower-bounded margin under the learned critic, up to Bellman and prediction errors. With enough positive barrier margin, it has a nonnegative continuation margin. This is a conditional statement about the selected action, not a closed-loop guarantee for the soft shield, which may still sample actions with $b^k_{Q,t} < 0$. The bound holds for any selected action and describes how model and critic errors erode the learned margin, not the true cost. For continuous-action settings, action selections are over the finite candidate set evaluated by the shield. See Appendix A for proofs, and Appendix A.2 for an episode-level budget bound.

5 Experiments

We evaluate our method on the safe ICRL benchmark [41], comparing against safe ICRL methods [41] and safe meta-RL baselines augmented with a context encoder [15, 68]. Our experiments are designed to answer two questions: (i) whether shielding improves the reward-safety tradeoff during in-context adaptation without parameter updates, and (ii) how this tradeoff changes as the safety budget varies.

5.1 Baselines

We compare against two baseline groups. The first consists of parameter-update-free safe adaptation methods, which adapt through context at test time without gradient updates. This group includes Safe AD and SCARED, which differ mainly in how safety is learned during pretraining. The second consists of safe meta-RL methods, which adapt through per-task parameter updates at test time.

Safe Algorithm Distillation (Safe AD). Following algorithm distillation [34], Safe AD [41] trains a history-conditioned policy on logged learning trajectories generated by safe RL algorithms. At test time, the policy conditions on the interaction history together with RTG and CTG targets, without any parameter updates.

SCARED. This is the state-of-the-art safe ICRL baseline trained with an exact-penalty dual objective to improve reward-safety tradeoffs through in-context adaptation [41]. We use SCARED as the base policy for our shield; its baseline curves use experimental data supplied by the original authors.

Safe Meta-RL. We compare against gradient-based safe meta-RL methods: MAML with penalty [15] and SafeMeta [68]. MAML with penalty learns a cost-regularized initialization, while SafeMeta meta-trains a closed-form one-step safe adaptation rule. Unlike in-context baselines, both adapt through parameter updates rather than conditioning on cross-episode histories.

5.2 Experimental Setup

Safe ICRL Benchmarks. We evaluate our approach on three safe ICRL benchmarks [41] studying two complementary generalization regimes: *structural OOD* tasks and *unseen ID* tasks. For structural OOD generalization, we use SafeDarkRoom and SafeDarkMujoco (Point, Car), where test tasks involve shifted goal and obstacle configurations under partial observability: Range-based sensing (e.g., lidar) is unavailable, so the agent cannot directly observe goal or obstacle locations and must adapt from in-context history. For unseen ID generalization, we use SafeVelocity (HalfCheetah, Ant), where each task is specified by a target velocity. We train on a subset of target velocities from a fixed range and evaluate on held-out velocities from the same range. See Appendix G for more details.

Training Setup. For the structural OOD environments, all algorithms are trained on source tasks with center-oriented obstacles and goals and evaluated on 100 OOD test tasks with edge-oriented obstacle and goal distributions. For the unseen-ID environments, target-velocity multipliers are sampled uniformly from $[0.5, 2.0]$ during training. We evaluate each method on 100 held-out contexts sampled from the same range, keeping the target velocity fixed across the in-context episode adaptation.

Safe AD follows the training protocol of [41], using trajectories generated by PPO-Lag [47] and the training hyperparameters. We use SCARED as the base policy and keep its original pretraining budgets: 3K epochs for SafeDarkRoom and SafeDarkMujoco, and 2K for SafeVelocity. An epoch is one collection-update cycle, consisting of 1.5K environment steps per training environment followed by 1K optimization batches. The shield-supporting modules in Section 3 are trained during this same pretraining phase, with no additional epochs. At test time, Q-Barrier shield reweights candidate actions using Algorithm 2, without parameter updates. Safe meta-RL baselines are trained for up to 15K meta-training steps on each environment, with most converging earlier. For Safe AD and the safe meta-RL baselines, we perform hyperparameter optimization to ensure a fair comparison. All loss weights in Equation (4) for our shield-supporting modules are fixed across environments. See Appendix H for full hyperparameters and compute details.

Evaluation Setup. To answer the two research questions, we use a consistent target-conditioning setup across methods. For SCARED and our Q-Barrier, we evaluate robustness to safety budgets by conditioning on CTG values sampled uniformly from $[1, 15]$ in SafeDarkRoom, $[10, 50]$ in SafeDarkMujoco, and $[0, 5]$ in SafeVelocity, and we report performance averaged over 100 distinct CTG targets. Q-Barrier uses the soft shield with a candidate-set size of $N_s = 8$ in continuous-action environments. The same CTG intervals and number of targets are used for SafeMeta, MAML with penalty, and Safe AD. For Safe AD, following [41], we use paired RTG-CTG conditioning: each CTG target sampled from the same interval is paired with a corresponding RTG target. In SafeDarkRoom and SafeDarkMujoco, RTG is set as $\max\{0.1, \mathrm{CTG}/\mathrm{CTG}_{\max}\}$ with $\mathrm{CTG}_{\max} = 15$ and 50, respectively. In the unseen-ID tasks, RTG and CTG are jointly scaled from 0.5 to $\mathrm{RTG}_{\max} = 200$, and $\mathrm{CTG}_{\max} = 5$ for both. Figure captions distinguish the evaluation-level shading conventions; Appendix H specifies aggregation horizons and uncertainty scope.

5.3 Results Analysis

RQ1: Does shielding improve the reward-safety tradeoff during in-context adaptation without parameter updates?

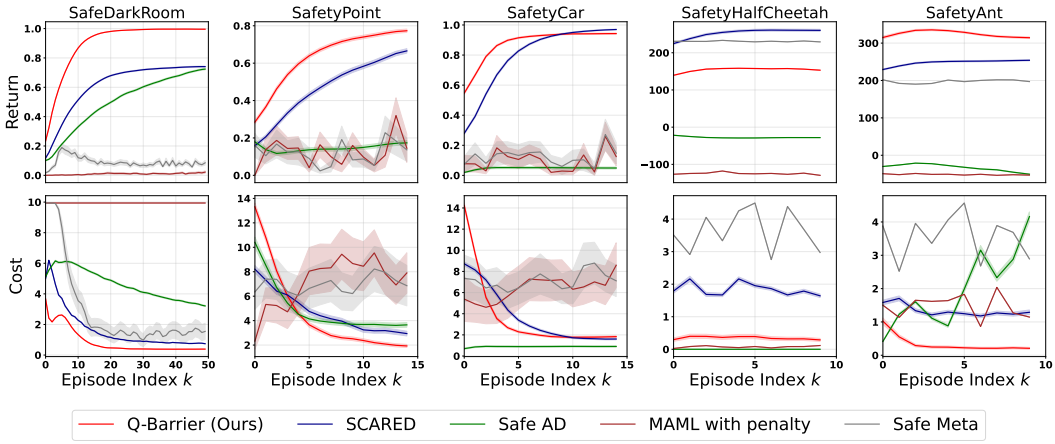


Figure 1: **In-context adaptation dynamics (RQ1).** Per-episode return (top) and cost (bottom) over in-context episode index k on test-time evaluation tasks. Curves show means over 100 evaluation tasks, with shaded regions denoting standard error of the mean. Q-Barrier generally improves adaptation by reaching high return earlier and reducing cost after a short context window, while SafetyHalfCheetah shows a conservative Pareto point with lower cost and lower return

We compare Q-Barrier and the baselines over in-context episodes on OOD test tasks, measuring how return and cost evolve as context accumulates (Figure 1). Since Q-Barrier uses SCARED as its base policy, the most direct comparison is against SCARED. Safe AD, Safe Meta, and MAML with penalty provide additional safe ICRL and safe meta-RL baselines. During OOD adaptation, these baselines fail to reduce cost consistently or become overly conservative sacrificing return.

Q-Barrier generally improves the in-context adaptation by reaching high return earlier while reducing cost after a short context window. In SafeDarkRoom and SafetyAnt, the improvement is immediate. In SafeDarkRoom, Q-Barrier achieves mean return 0.9 compared with 0.62 for SCARED, a 46% gain, while reducing mean cost from 1.66 to 0.87, a 47.2% drop. In SafetyAnt, Q-Barrier achieves mean return 324.36 compared with 247.8 for SCARED, a 30.9% gain, while reducing mean cost from 1.34 to 0.34, a 74.3% drop.

DarkMujoco navigation tasks show a different but informative pattern: Q-Barrier initially encounters higher cost, but this early cost comes with higher return. In SafetyPoint and SafetyCar, Q-Barrier first becomes lower-cost than SCARED at episodes $k = 3$ and $k = 2$, respectively. Averaged over the remaining adaptation, Q-Barrier reduces mean cost from 4.0 to 2.99 in SafetyPoint, a 25.3% drop, and from 2.95 to 2.37 in SafetyCar, a 19.7% drop. At the same time, Q-Barrier maintains an early return advantage in both tasks. For example, in SafetyCar, Q-Barrier exceeds return 0.9 by episode $k = 5$, whereas SCARED reaches this level around episode $k = 7$.

SafetyHalfCheetah is the main conservative case. During the full in-context adaptation, Q-Barrier consistently lowers cost but also lowers return relative to SCARED. Averaged over the adaptation, it reduces mean cost from 1.84 to 0.35, a 81.0% drop, while mean return decreases from 252.75 to 154.26, a 39.0% drop. This places Q-Barrier at a more conservative point on the reward-cost frontier. The results in Appendix B provide a plausible explanation: SafetyHalfCheetah has the largest latent prediction error among the evaluated environments, which can make the learned barrier more conservative.

Takeaway: Q-Barrier improves the OOD reward-safety tradeoff during in-context adaptation, with SafetyHalfCheetah showing a conservative Pareto point that trades lower return for lower cost.

RQ2: How does the reward-safety tradeoff change across safety budgets?

We next evaluate the objective in Equation (1) by sweeping the cost budget and measuring cumulative return together with average episode cost over the in-context adaptation.

Figure 2 shows that Q-Barrier achieves the highest cumulative return in four of five environments across different budgets: SafeDarkRoom, SafetyPoint, SafetyCar, and SafetyAnt. It also satisfies the average episode-cost constraint across most budget settings in the sweep. Violations occur only at

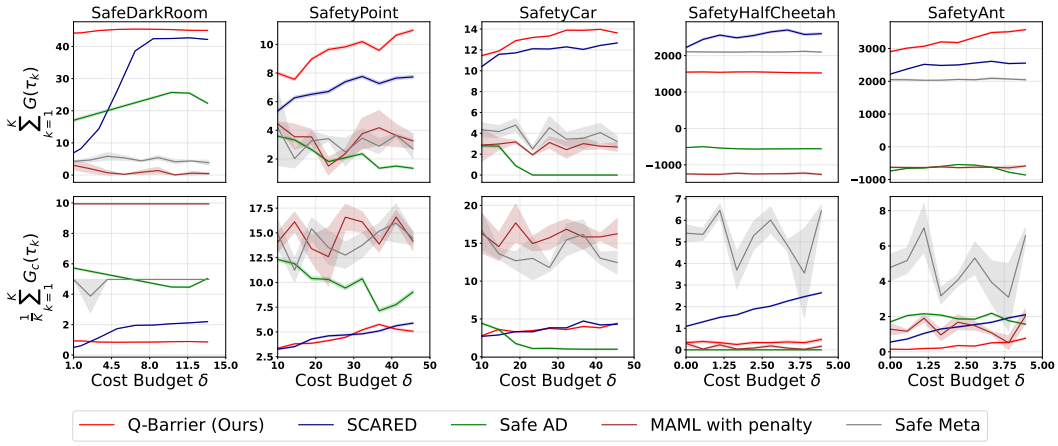


Figure 2: **Reward-cost tradeoffs under budget variation (RQ2).** Cumulative reward (top) and average episode cost (bottom) over varying cost budgets on test-time evaluation tasks. Across different budget levels, Q-Barrier achieves the highest cumulative return in four of five environments while maintaining budget compliance in most cases, with violations mostly limited to near-zero budgets. Shaded regions denote standard errors of the mean over 100 evaluation tasks.

near-zero budgets, where even small residual costs exceed the requested budget. This is the most difficult setting for Q-Barrier, since small model or critic errors can produce a violation when the allowed cost is near zero.

Compared with SCARED base policy, Q-Barrier improves average cumulative return over budget levels by 54.4% in SafeDarkRoom, 36.3% in SafetyPoint, 10.1% in SafetyCar, and 31.0% in SafetyAnt. These gains are not obtained by simply spending more cost. Q-Barrier also reduces average episode cost relative to SCARED by 44.1% in SafeDarkRoom, 2.1% in SafetyPoint, 1.0% in SafetyCar, and 74.2% in SafetyAnt. Thus, in most environments, Q-Barrier improves return without increasing average cost, and in SafeDarkRoom and SafetyAnt it improves both return and cost by large margins.

SafeDarkRoom demonstrates the benefit of explicit budget-aware action selection. SCARED becomes conservative as the budget varies and does not fully convert available budget into return. In particular, given budget near 1, Q-Barrier exceeds return 40 while satisfying budget constraints, but SCARED stays under return 10.

SafetyHalfCheetah is the main exception. Q-Barrier reduces average episode cost by 81.3% relative to SCARED, but its cumulative return is 39.1% lower. Safe Meta also achieves higher cumulative return than Q-Barrier, but Safe Meta often violates the cost budget by a large margin (e.g., average cost around 6 when cost budget is 1). This is consistent with the conservative behavior observed in RQ1, which can be explained by the largest latent prediction error.

Takeaway: Q-Barrier improves the OOD reward-safety frontier across budget levels in most environments, often increasing return while matching or reducing cost. SafetyHalfCheetah remains the conservative case, trading lower return for substantially lower cost.

Ablation Studies. We include ablations and diagnostics for the main shielding design choices. Appendix I adds auxiliary-loss checks, a cost-penalty comparison, executed-action margins, candidate feasibility across N_s, the training context length, and longer adaptation on Point and Car. First, we compare the soft and hard rules in Algorithm 2; hard truncation restricts candidates to those with nonnegative deployment margin (Appendix C). The soft shield matches the hard shield's reward-safety tradeoffs while preserving broader action support during early adaptation. Since hard filtering provides marginal and non-uniform cost reductions, we use soft reweighting in all main experiments (Appendix D). Second, we study how the number of sampled candidate actions N_s affects shielding in continuous-control environments, sweeping $N_s \in \{4, 8, 16, 32\}$. Performance is stable across N_s in SafeDarkMujoco, while SafeVelocity is more affected by N_s: larger candidate sets raise both return and cost (Appendix E). Appendix B reports empirical diagnostics related to the barrier-margin bound (Theorem 1), including latent prediction error, critic sensitivity, and discounted Bellman-residual proxies, consistent with the conservative shielding observed in SafetyHalfCheetah. Appendix F

provides pseudocode for the training and runtime shielding procedures, Appendix G describes the benchmark environments and OOD generation protocol, and Appendix H lists full hyperparameters for all methods.

6 Related Work

Safe Reinforcement Learning. Safe reinforcement learning is commonly formalized through constrained Markov decision processes (CMDPs), where policies maximize reward subject to expected cost constraints. Existing methods include constrained policy optimization, Lagrangian and primal-dual algorithms, projection-based policy optimization, recovery policies, model-based safe RL, and sample-efficient constraint-handling methods [1, 59, 47, 73, 71, 70, 20, 5]. Runtime shielding provides a complementary mechanism by filtering, modifying, or replacing actions before execution [3, 7, 72], and recent safe planning and shielding methods use adaptive conformal prediction to calibrate uncertainty in safety-critical decision making [36, 53, 54, 49]. Control barrier functions provide certificate-based safety guarantees, but constructing valid barriers under learned, uncertain, or varying dynamics often requires explicit model knowledge or robust uncertainty bounds [11, 10, 16, 63, 64]. Our work is complementary to these methods: we introduce a budget-aware Q-Barrier shield for frozen in-context RL policies.

In-Context Reinforcement Learning and Safe Adaptation. In-context reinforcement learning (ICRL) studies agents that adapt to new tasks at test time without updating parameters, by conditioning the policy on an expanding interaction history [40, 48]. This perspective is closely related to meta-RL, and contextual or hidden-parameter RL, where agents infer task or dynamics information from history and use it to improve behavior online [15, 69, 8, 9, 6]. Recent ICRL methods use transformers, recurrent models, state-space models, or algorithm-distillation objectives to induce adaptation behavior in the forward pass of a pretrained policy [14, 60, 67, 35, 34, 46, 28, 12, 13, 30, 17, 18, 61, 62]. Recent theoretical analyses connect TD policy evaluation to specific transformer constructions, including softmax-attention layers [66] and chain-of-thought generation with linear transformers [65]. Safe meta-RL extends fast adaptation to constrained settings, but often relies on explicit task-level adaptation, gradient updates, or additional offline adaptation data [39, 26, 21, 68]. Closest to our setting, SCARED formulates safe ICRL under CMDPs and learns budget-sensitive policies through Lagrangian pretraining without test-time parameter updates [41]. Our method focuses on a deployment-time gap: with a fixed pretrained ICRL policy, action selection may still place probability on actions whose predicted future cost exceeds the remaining budget. We therefore add a runtime barrier that reweights candidate actions using the current latent context, remaining budget, and predicted cumulative cost.

Latent Representations, Safety Filters, and Auxiliary Training. Our shield relies on latent representations that support policy adaptation, latent dynamics prediction, and safety estimation. Prior work on context encoders, function encoders, and world-model representations shows that learned latent variables can capture task or dynamics information useful for zero-shot transfer, in-context adaptation, and hidden-parameter generalization [38, 23, 22, 44, 29, 31]. Closely related to our runtime safety mechanism, recent latent safety-filter methods perform safety analysis directly in learned world-model spaces using latent reachability, uncertainty-aware filtering, or runtime-adaptive safety constraints [42, 52, 2]. These works provide important precedent for latent-space runtime safety reasoning, but our mechanism differs: we use a CMDP-style remaining budget, a world-projected latent state, and a pessimistic cost-to-go critic to define a latent Q-barrier for candidate-action reweighting. Finally, our training objective uses auxiliary dynamics and safety losses to shape the representation. Auxiliary prediction objectives are widely used to improve RL representations [24, 33, 51, 74] but joint policy, dynamics, and safety optimization can create gradient interference. To reduce this interference, we use an asymmetric stop-gradient scheme inspired by self-supervised representation learning methods such as BYOL, where prediction heads are trained against target representations while gradients are selectively blocked [19]. In our setting, grounding losses update the shared encoder, while alignment losses pass through projection heads with stop-gradient, allowing auxiliary supervision to regularize the latent space without directly destabilizing policy learning.

7 Conclusion

We introduced Q-Barrier, a latent shielding mechanism that makes frozen safe ICRL policies explicitly budget-aware under test-time generalization shifts, including both structural OOD and unseen ID settings. The method augments a safe ICRL policy with shield-supporting modules learned during training: a world-projected latent representation, a latent dynamics model, and a cost-critic ensemble. At deployment, all learned parameters remain fixed, and Q-Barrier only reweights finite action candidates using the remaining budget and predicted cost-to-go, so adaptation comes from the growing in-context history rather than online parameter updates.

Our analysis gives a conditional error bound on how the Q-Barrier margin propagates one step forward over the next candidate set, up to Bellman upper bound and latent prediction errors. It does not certify closed-loop safety. Empirically, Q-Barrier improves reward-safety tradeoffs during test-time in-context adaptation. After a short context window, it achieves higher return in four of five environments while matching or lowering average episode cost in all five, with one environment showing a conservative lower-cost/lower-return tradeoff. Across budget sweeps, it achieves the highest return in four of five environments while generally matching or reducing average cost relative to baselines.

Limitations and future work. Q-Barrier depends on latent prediction accuracy, critic calibration, Bellman residuals, and sampled candidate quality. Overly pessimistic cost estimates can downweight useful high-return actions. Future work should improve uncertainty calibration, adapt shield strength to local uncertainty and remaining budget, extend to multiple constraints, and develop stronger closed-loop analyses for stochastic shielded policies.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation grant CCF-1942836.

References

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the International Conference on Machine Learning*, 2017.

[2] Sankalp Agrawal, Junwon Seo, Kensuke Nakamura, Ran Tian, and Andrea Bajcsy. Anysafe: Adapting latent safety filters at runtime via safety constraint parameterization in the latent space. *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

[3] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *AAAI Conference on Artificial Intelligence*, 2018.

[4] Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

[5] Yarden As, Bhavya Sukhija, Lenart Treven, Carmelo Sferrazza, Stelian Coros, and Andreas Krause. Actsafe: Active exploration with safety constraints for reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[6] Carolin Benjamins, Theresa Eimer, Frederik Schubert, Aditya Mohan, Sebastian Döhler, André Biedenkapp, Bodo Rosenhahn, Frank Hutter, and Marius Lindauer. Contextualize me – the case for context in reinforcement learning. *Transactions on Machine Learning Research*, 2023.

[7] Steven Carr, Nils Jansen, Sebastian Junges, and Ufuk Topcu. Safe reinforcement learning via shielding under partial observability. In *The Thirty-Seventh AAAI Conference on Artificial Intelligence*, 2023.

[8] Baiming Chen, Zuxin Liu, Jiacheng Zhu, Mengdi Xu, Wenhao Ding, Liang Li, and Ding Zhao. Context-aware safe reinforcement learning for non-stationary environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[9] Xiaoyu Chen, Xiangming Zhu, Yufeng Zheng, Pushi Zhang, Li Zhao, Wenxue Cheng, Peng Cheng, Yongqiang Xiong, Tao Qin, Jianyu Chen, and Tie-Yan Liu. An adaptive deep RL method for non-stationary environments with piecewise stable context. In *Advances in Neural Information Processing Systems*, 2022.

[10] Yikun Cheng, Pan Zhao, and Naira Hovakimyan. Safe and efficient reinforcement learning using disturbance-observer-based control barrier functions. In *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, 2023.

[11] Jason Choi, Fernando Castañeda, Claire J. Tomlin, and Koushil Sreenath. Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions. In *Robotics: Science and Systems*, 2020.

[12] Zhenwen Dai, Federico Tomasi, and Sina Ghiassian. In-context exploration-exploitation for reinforcement learning. *ArXiv Preprint*, 2024.

[13] Juncheng Dong, Moyang Guo, Ethan X. Fang, Zhuoran Yang, and Vahid Tarokh. In-context reinforcement learning without optimal action labels. In *ICML 2024 Workshop on In-Context Learning*, 2024.

[14] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *ArXiv Preprint*, 2016.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the International Conference on Machine Learning*, 2017.

[16] Milan Ganai, Zheng Gong, Chenning Yu, Sylvia Lee Herbert, and Sicun Gao. Iterative reachability estimation for safe reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[17] Jake Grigsby, Linxi Fan, and Yuke Zhu. Amago: Scalable in-context reinforcement learning for adaptive agents. In *Proceedings of the International Conference on Learning Representations*, 2024.

[18] Jake Grigsby, Justin Sasek, Samyak Parajuli, Ikechukwu D Adebi, Amy Zhang, and Yuke Zhu. Amago-2: Breaking the multi-task barrier in meta-reinforcement learning with transformers. In *Advances in Neural Information Processing Systems*, 2024.

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent a new approach to self-supervised learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

[20] Shangding Gu, Laixi Shi, Yuhao Ding, Alois Knoll, Costas Spanos, Adam Wierman, and Ming Jin. Enhancing efficiency of safe reinforcement learning via sample manipulation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[21] Cong Guan, Ruiqi Xue, Ziqian Zhang, Lihe Li, Yi-Chen Li, Lei Yuan, and Yang Yu. Cost-aware offline safe meta reinforcement learning with robust in-distribution online task adaptation. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 2024.

[22] Tyler Ingebrand, Adam Thorpe, and Ufuk Topcu. Zero-shot transfer of neural ODEs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[23] Tyler Ingebrand, Amy Zhang, and Ufuk Topcu. Zero-shot reinforcement learning via function encoders. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[24] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2017.

[25] 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. In *Advances in Neural Information Processing Systems*, 2023.

[26] Vanshaj Khattar, Yuhao Ding, Bilgehan Sel, Javad Lavaei, and Ming Jin. A CMDP-within-online framework for meta-safe reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2023.

[27] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *ArXiv Preprint*, 2021.

[28] Louis Kirsch, James Harrison, C. Freeman, Jascha Sohl-Dickstein, and Jürgen Schmidhuber. Towards general-purpose in-context learning agents. In *NeurIPS Foundation Models for Decision Making Workshop*, 2023.

[29] Cevahir Koprulu, Thiago D. Simão, Nils Jansen, and Ufuk Topcu. Safety-prioritizing curricula for constrained reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[30] Akshay Krishnamurthy, Keegan Harris, Dylan J Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context? *ArXiv Preprint*, 2024.

[31] Minjae Kwon, Tyler Ingebrand, Ufuk Topcu, and Lu Feng. Adaptive shielding for safe reinforcement learning under hidden-parameter dynamics shifts, 2026. URL https://arxiv.org/abs/2506.11033.

[32] Minjae Kwon, Josephine Lamp, and Lu Feng. Safety generalization under distribution shift in safe reinforcement learning: A diabetes testbed. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

[33] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2020.

[34] Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Stenberg Hansen, Angelos Filos, Ethan Brooks, Maxime Gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. In *Proceedings of the International Conference on Learning Representations*, 2023.

[35] Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

[36] Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J. Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 2023.

[37] Jinmei Liu, Fuhong Liu, Zhenhong Sun, Jianye HAO, Huaxiong Li, Bo Wang, Daoyi Dong, Chunlin Chen, and Zhi Wang. Scalable in-context q-learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

[38] Fan-Ming Luo, Shengyi Jiang, Yang Yu, ZongZhang Zhang, and Yi-Feng Zhang. Adapt to environment sudden changes by learning a context-sensitive policy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[39] Michael Luo, Ashwin Balakrishna, Brijen Thananjeyan, Suraj Nair, Julian Ibarz, Jie Tan, Chelsea Finn, Ion Stoica, and Ken Goldberg. Mesa: Offline meta-rl for safe adaptation and fault tolerance. In *Advances in Neural Information Processing Systems*, 2021.

[40] Amir Moeini, Jiuqi Wang, Jacob Beck, Ethan Blaser, Shimon Whiteson, Rohan Chandra, and Shangtong Zhang. A survey of in-context reinforcement learning. *ArXiv Preprint*, 2025.

[41] Amir Moeini, Minjae Kwon, Alper Kamil Bozkurt, Yuichi Motai, Rohan Chandra, Lu Feng, and Shangtong Zhang. Safe in-context reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2026.

[42] Kensuke Nakamura, Lasse Peters, and Andrea Bajcsy. Generalizing safety beyond collision-avoidance via latent-space reachability analysis. In *Robotics: Science and Systems XXI*, 2025.

[43] Andrei Polubarov, Lyubaykin Nikita, Alexander Derevyagin, Ilya Zisman, Denis Tarasov, Alexander Nikulin, and Vladislav Kurenkov. Vintix: Action model via in-context reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025.

[44] Sai Prasanna, Karim Farid, Raghu Rajan, and André Biedenkapp. Dreaming of many worlds: Learning contextual world models aids zero-shot generalization. *Reinforcement Learning Journal*, 2024.

[45] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[46] Sharath Chandra Raparthy, Eric Hambro, Robert Kirk, Mikael Henaff, and Roberta Raileanu. Generalization to new sequential decision making tasks with in-context learning. *ArXiv Preprint*, 2023.

[47] Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *OpenAI. https://cdn.openai.com/safexp-short.pdf*, 2019.

[48] A Run and Ziluo Ding. In-context reinforcement learning under non-stationarity: A survey, 2026. URL https://arxiv.org/abs/2607.11906.

[49] William Scarbro, Calum Imrie, Sinem Getir Yaman, Kavan Fatehi, Corina Păsăreanu, Radu Calinescu, and Ravi Mangal. Conformal safety shielding for imperfect-perception agents. In *International Conference on Runtime Verification*, 2025.

[50] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv Preprint*, 2017.

[51] Max Schwarzer, Nitarshan Rajkumar, Michael Noukhovitch, Ankesh Anand, Laurent Charlin, R Devon Hjelm, Philip Bachman, and Aaron Courville. Pretraining representations for data-efficient reinforcement learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[52] Junwon Seo, Kensuke Nakamura, and Andrea Bajcsy. Uncertainty-aware latent safety filters for avoiding out-of-distribution failures. *Conference on Robot Learning (CoRL)*, 2025.

[53] Shili Sheng, David Parker, and Lu Feng. Safe pomdp online planning via shielding. In *2024 IEEE International Conference on Robotics and Automation*, 2024.

[54] Shili Sheng, Pian Yu, David Parker, Marta Kwiatkowska, and Lu Feng. Safe pomdp online planning among dynamic agents via adaptive conformal prediction. *IEEE Robotics and Automation Letters*, 2024.

[55] Viacheslav Sinii, Alexander Nikulin, Vladislav Kurenkov, Ilya Zisman, and Sergey Kolesnikov. In-context reinforcement learning for variable action spaces. In *Forty-first International Conference on Machine Learning*, 2024.

[56] Jaehyeon Son, Soochan Lee, and Gunhee Kim. Distilling reinforcement learning algorithms for in-context model-based planning. In *Proceedings of the International Conference on Learning Representations*, 2025.

[57] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction (2nd Edition)*. MIT Press, 2018.

[58] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2012.

[59] Akifumi Wachi and Yanan Sui. Safe reinforcement learning in constrained markov decision processes. In *Proceedings of the International Conference on Machine Learning*, 2020.

[60] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *ArXiv Preprint*, 2016.

[61] Jiuqi Wang, Ethan Blaser, Hadi Daneshmand, and Shangtong Zhang. Transformers can learn temporal difference methods for in-context reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2025.

[62] Jiuqi Wang, Rohan Chandra, and Shangtong Zhang. Towards provable emergence of in-context reinforcement learning. In *Advances in Neural Information Processing Systems*, 2025.

[63] Yixuan Wang, Simon Sinong Zhan, Ruochen Jiao, Zhilu Wang, Wanxin Jin, Zhuoran Yang, Zhaoran Wang, Chao Huang, and Qi Zhu. Enforcing hard constraints with soft barriers: Safe reinforcement learning in unknown stochastic environments. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[64] Wei Xiao, Tsun-Hsuan Wang, Ramin Hasani, Makram Chahine, Alexander Amini, Xiao Li, and Daniela Rus. Barriernet: Differentiable control barrier functions for learning of safe robot control. *IEEE Transactions on Robotics*, 2023.

[65] Zixuan Xie, Xinyu Liu, Rohan Chandra, and Shangtong Zhang. Convergence and emergence of in-context reinforcement learning with chain of thought, 2026. URL https://arxiv.org/abs/2605.07123.

[66] Zixuan Xie, Xinyu Liu, Claire Chen, Shuze Daniel Liu, Rohan Chandra, and Shangtong Zhang. Beyond linear attention: Softmax transformers implement in-context reinforcement learning, 2026. URL https://arxiv.org/abs/2605.07333.

[67] Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In *Proceedings of the International Conference on Machine Learning*, 2022.

[68] Siyuan Xu and Minghui Zhu. Efficient safe meta-reinforcement learning: Provable near-optimality and anytime safety. In *Advances in Neural Information Processing Systems*, 2025.

[69] Jiachen Yang, Brenden Petersen, Hongyuan Zha, and Daniel Faissol. Single episode policy transfer in reinforcement learning. In *International Conference on Learning Representations*, 2020.

[70] Long Yang, Jiaming Ji, Juntao Dai, Linrui Zhang, Binbin Zhou, Pengfei Li, Yaodong Yang, and Gang Pan. Constrained update projection approach to safe policy optimization. In *Advances in Neural Information Processing Systems*, 2022.

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

[72] Wen-Chi Yang, Giuseppe Marra, Gavin Rens, and Luc De Raedt. Safe reinforcement learning via probabilistic logic shields. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 2023.

[73] Yiming Zhang, Quan Vuong, and Keith W. Ross. First order constrained optimization in policy space. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

[74] Yi Zhao, Wenshuai Zhao, Rinu Boney, Juho Kannala, and Joni Pajarinen. Simplified temporal consistency reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[75] Ilya Zisman, Vladislav Kurenkov, Alexander Nikulin, Viacheslav Sinii, and Sergey Kolesnikov. Emergence of in-context reinforcement learning from noise distillation. *ArXiv Preprint*, 2023.

A Proofs

A.1 Proof of Theorem 1

Before proving the main theorem, we first prove a value error lemma. Under Assumptions 1 and 2, the lemma shows that latent prediction error can change the estimated next-step continuation value by at most $L_Q \varepsilon_{\text{pred}}$.

Lemma 1. *Under Assumptions 1 and 2,*

$$\left| \hat{V}^{+,k}_{C,t+1}(f_z(Z^{w,k}_t, A^k_t)) - \hat{V}^{+,k}_{C,t+1}(Z^{w,k}_{t+1}) \right| \le L_Q \varepsilon_{\text{pred}}. \tag{7}$$

Proof. Fix episode k and timestep t, and condition on the realized next-step candidate set $\mathcal{A}^{\text{cand}}_{t+1,k}$. Write $\hat{V}^+_C(Z) \doteq \hat{V}^{+,k}_{C,t+1}(Z) = \min_{A' \in \mathcal{A}^{\text{cand}}_{t+1,k}} \hat{Q}^+_C(Z, A')$. For readability, we omit $+$ and k in intermediate expressions when there is no ambiguity. Since $\mathcal{A}^{\text{cand}}_{t+1,k}$ is fixed in this argument and each function $\hat{Q}^+_C(\cdot, A')$ is L_Q-Lipschitz, their pointwise minimum is also L_Q-Lipschitz. To see this, let $A^\star \in \arg\min_{A' \in \mathcal{A}^{\text{cand}}_{t+1,k}} \hat{Q}^+_C(Z', A')$. Then $\hat{V}^+_C(Z) - \hat{V}^+_C(Z') \le \hat{Q}^+_C(Z, A^\star) - \hat{Q}^+_C(Z', A^\star) \le L_Q \|Z - Z'\|_2$. Swapping Z and Z' leads to $|\hat{V}^+_C(Z) - \hat{V}^+_C(Z')| \le L_Q \|Z - Z'\|_2$. Applying this bound with $Z = f_z(Z^{w,k}_t, A^k_t)$ and $Z' = Z^{w,k}_{t+1}$, and using Assumption 1, proves Equation (7). □

Proof. Fix episode k and timestep t, and condition on the realized next-step candidate set $\mathcal{A}^{\text{cand}}_{t+1,k}$. We follow the same convention used in Proof of Lemma 1.

Starting from the next-step state barrier and using the budget recursion $B^k_{t+1} = B^k_t - C^k_{t+1}$, we have

$$\begin{aligned}
b^k_{V,t+1}(Z^{w,k}_{t+1}, B^k_{t+1}) &= B^k_{t+1} - \hat{V}^+_C(Z^{w,k}_{t+1}) \\
&= B^k_t - C^k_{t+1} - \hat{V}^+_C(Z^{w,k}_{t+1}) \\
&= \underbrace{B^k_t - \hat{Q}^+_C(Z^{w,k}_t, A^k_t)}_{b^k_{Q,t}(Z^{w,k}_t, B^k_t, A^k_t)} \\
&\quad + \left(\hat{Q}^+_C(Z^{w,k}_t, A^k_t) - C^k_{t+1} - \hat{V}^+_C(f_z(Z^{w,k}_t, A^k_t)) \right) \\
&\quad + \left(\hat{V}^+_C(f_z(Z^{w,k}_t, A^k_t)) - \hat{V}^+_C(Z^{w,k}_{t+1}) \right).
\end{aligned}$$

By the definition of the Bellman upper-bound residual,

$$\hat{Q}^+_C(Z^{w,k}_t, A^k_t) - C^k_{t+1} - \hat{V}^+_C(f_z(Z^{w,k}_t, A^k_t)) \ge -\Delta^{+,k}_{\text{bell},t}.$$

By Lemma 1, $\hat{V}^+_C(f_z(Z^{w,k}_t, A^k_t)) - \hat{V}^+_C(Z^{w,k}_{t+1}) \ge -L_Q \varepsilon_{\text{pred}}$. Substituting both inequalities gives

$$b^k_{V,t+1}(Z^{w,k}_{t+1}, B^k_{t+1}) \ge b^k_{Q,t}(Z^{w,k}_t, B^k_t, A^k_t) - \Delta^{+,k}_{\text{bell},t} - L_Q \varepsilon_{\text{pred}},$$

which proves the claim. □

A.2 Episode-Level Budget Bound

Theorem 1 is a one-step result. This subsection shows that it can be extended into an episode-level bound on cumulative budget accumulation by introducing a term that captures the gap between the action actually selected and the critic-minimizing action.

Definition 3 (Selection Gap)**.** *For the action selected at time t in episode k, define*

$$\Delta^k_{\text{sel},t} \doteq \hat{Q}^+_C(Z^{w,k}_t, A^k_t) - \hat{V}^{+,k}_{C,t}(Z^{w,k}_t) \ge 0. \tag{8}$$

This quantity is the excess predicted cost-to-go of the selected action over the minimum available in the candidate set at that state. It is zero when the deployed policy selects the critic-minimizing action, and positive when stochastic shielding picks a higher-cost action.Recall that

$$b^k_{V,t}(Z, B) \doteq B - \hat{V}^{+,k}_{C,t}(Z), \quad b^k_{Q,t}(Z, B, A) \doteq B - \hat{Q}^+_C(Z, A).$$

For continuous-action environments, all quantities are conditioned on the realized finite candidate sets. The candidate set $\mathcal{A}^{\mathrm{cand}}_{t+1,k}$ used in the Bellman residual at time t is the same candidate set used to define $b^k_{V,t+1}$ at the next decision making. We assume that each realized candidate set is nonempty and that the executed action A^k_t is selected from $\mathcal{A}^{\mathrm{cand}}_{t,k}$. Thus $\Delta^k_{\mathrm{sel},t} \geq 0$. In continuous-action domains, the proposition is conditional on the entire realized sequence of finite candidate sets.

Proposition 1 (Episode-Level Budget Bound)**.** *Suppose Theorem 1 holds along episode k with the signed budget recursion $B^k_{t+1} = B^k_t - C^k_{t+1}$. Assume the terminal condition $\hat{V}^{+,k}_{C,T_k}(Z^{w,k}_{T_k}) = 0$. Then*

$$G_{c,0}(\tau_k) - B^k_0 \leq -b^k_{V,0}(Z^{w,k}_0, B^k_0) + \sum_{t=0}^{T_k-1} \Big(\Delta^k_{\mathrm{sel},t} + \Delta^{+,k}_{\mathrm{bell},t} + L_Q \varepsilon_{\mathrm{pred}}\Big). \tag{9}$$

In particular, if $B^k_0 = \delta$ and $b^k_{V,0}(Z^{w,k}_0, \delta) \geq 0$, then

$$G_{c,0}(\tau_k) - \delta \leq \sum_{t=0}^{T_k-1} \Big(\Delta^k_{\mathrm{sel},t} + \Delta^{+,k}_{\mathrm{bell},t} + L_Q \varepsilon_{\mathrm{pred}}\Big). \tag{10}$$

Averaging over K episodes and taking expectation gives

$$\frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\big[G_{c,0}(\tau_k)\big] \leq \delta + \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\left[\sum_{t=0}^{T_k-1} \Big(\Delta^k_{\mathrm{sel},t} + \Delta^{+,k}_{\mathrm{bell},t} + L_Q \varepsilon_{\mathrm{pred}}\Big) - b^k_{V,0}(Z^{w,k}_0, \delta)\right]. \tag{11}$$

Proof. By the definition of the selection gap,

$$\Delta^k_{\mathrm{sel},t} = \hat{Q}^+_C(Z^{w,k}_t, A^k_t) - \hat{V}^{+,k}_{C,t}(Z^{w,k}_t).$$

Therefore,

$$\begin{aligned} b^k_{Q,t}(Z^{w,k}_t, B^k_t, A^k_t) &= B^k_t - \hat{Q}^+_C(Z^{w,k}_t, A^k_t) \\ &= B^k_t - \hat{V}^{+,k}_{C,t}(Z^{w,k}_t) - \Delta^k_{\mathrm{sel},t} \\ &= b^k_{V,t}(Z^{w,k}_t, B^k_t) - \Delta^k_{\mathrm{sel},t}. \end{aligned}$$

Substituting this identity into Theorem 1 gives

$$b^k_{V,t+1}(Z^{w,k}_{t+1}, B^k_{t+1}) \geq b^k_{V,t}(Z^{w,k}_t, B^k_t) - \Delta^k_{\mathrm{sel},t} - \Delta^{+,k}_{\mathrm{bell},t} - L_Q \varepsilon_{\mathrm{pred}}. \tag{12}$$

Summing Equation (12) over $t = 0, \ldots, T_k - 1$ and telescoping provides

$$b^k_{V,T_k}(Z^{w,k}_{T_k}, B^k_{T_k}) \geq b^k_{V,0}(Z^{w,k}_0, B^k_0) - \sum_{t=0}^{T_k-1} \Big(\Delta^k_{\mathrm{sel},t} + \Delta^{+,k}_{\mathrm{bell},t} + L_Q \varepsilon_{\mathrm{pred}}\Big).$$

By the terminal condition $\hat{V}^{+,k}_{C,T_k}(Z^{w,k}_{T_k}) = 0$, we have

$$b^k_{V,T_k}(Z^{w,k}_{T_k}, B^k_{T_k}) = B^k_{T_k}.$$

Using the budget identity

$$B^k_{T_k} = B^k_0 - G_{c,0}(\tau_k),$$

we obtain

$$B^k_0 - G_{c,0}(\tau_k) \geq b^k_{V,0}(Z^{w,k}_0, B^k_0) - \sum_{t=0}^{T_k-1} \Big(\Delta^k_{\mathrm{sel},t} + \Delta^{+,k}_{\mathrm{bell},t} + L_Q \varepsilon_{\mathrm{pred}}\Big).$$

Finally, rearranging gives Equation (9). Setting $B^k_0 = \delta$ and using $b^k_{V,0}(Z^{w,k}_0, \delta) \geq 0$ gives Equation (10). Averaging Equation (9) over k and taking expectations gives Equation (11). □

Remark. Proposition 1 connects the local one-step barrier analysis to the aggregate budget objective through the selection gap $\Delta^k_{\mathrm{sel},t}$, which measures how much the chosen action's predicted cost exceeds the lowest predicted cost among candidates. For the soft shield, this term captures the cost of preserving policy support rather than always selecting the critic-greedy lowest-cost action. The result does not by itself guarantee $\frac{1}{K}\sum_k \mathbb{E}[G_{c,0}(\tau_k)] \leq \delta$. Instead, it decomposes episode-level budget into measurable residuals from action selection, critic conservatism, and latent prediction error.

A.3 Overlap of Consecutive Safe Sets

The following result gives a local interpretation of the Lipschitz regularity assumption. Since candidate sets may be resampled over time, the statement is conditional on an action being present in both consecutive candidate sets.

Definition 4 (Safe Candidate Set)**.** *For the realized candidate set $\mathcal{A}^{\mathrm{cand}}_{t,k}$, world-projected latent state $Z \in \mathcal{Z}^w$, and remaining budget B, define*

$$\mathcal{A}^k_{\mathrm{safe},t}(Z, B) \doteq \left\{ A \in \mathcal{A}^{\mathrm{cand}}_{t,k} : \hat{Q}^+_C(Z, A) \leq B \right\}. \tag{13}$$

Lemma 2 (Conditional Overlap of Consecutive Safe Candidate Sets)**.** *Suppose Assumption 2 holds. Let $A \in \mathcal{A}^{\mathrm{cand}}_{t,k} \cap \mathcal{A}^{\mathrm{cand}}_{t+1,k}$ satisfy*

$$\hat{Q}^+_C(Z^{w,k}_t, A) \leq B^k_t - \eta \tag{14}$$

for some margin $\eta > 0$. If

$$L_Q \|Z^{w,k}_{t+1} - Z^{w,k}_t\|_2 + |B^k_{t+1} - B^k_t| \leq \eta, \tag{15}$$

then

$$A \in \mathcal{A}^k_{\mathrm{safe},t+1}(Z^{w,k}_{t+1}, B^k_{t+1}). \tag{16}$$

Consequently, if such an action exists, then

$$\mathcal{A}^k_{\mathrm{safe},t}(Z^{w,k}_t, B^k_t) \cap \mathcal{A}^k_{\mathrm{safe},t+1}(Z^{w,k}_{t+1}, B^k_{t+1}) \neq \varnothing. \tag{17}$$

Proof. Because $A \in \mathcal{A}^{\mathrm{cand}}_{t,k} \cap \mathcal{A}^{\mathrm{cand}}_{t+1,k}$, it is eligible at both consecutive decision times. By Assumption 2,

$$|\hat{Q}^+_C(Z^{w,k}_{t+1}, A) - \hat{Q}^+_C(Z^{w,k}_t, A)| \leq L_Q \|Z^{w,k}_{t+1} - Z^{w,k}_t\|_2.$$

Using Equation (14), we have

$$\hat{Q}^+_C(Z^{w,k}_{t+1}, A) \leq B^k_t - \eta + L_Q \|Z^{w,k}_{t+1} - Z^{w,k}_t\|_2.$$

Therefore,

$$\hat{Q}^+_C(Z^{w,k}_{t+1}, A) - B^k_{t+1} \leq L_Q \|Z^{w,k}_{t+1} - Z^{w,k}_t\|_2 - \eta + |B^k_{t+1} - B^k_t|.$$

By Equation (15), the right-hand side is less or equal than 0, so

$$\hat{Q}^+_C(Z^{w,k}_{t+1}, A) \leq B^k_{t+1}.$$

Since $A \in \mathcal{A}^{\mathrm{cand}}_{t+1,k}$, this implies $A \in \mathcal{A}^k_{\mathrm{safe},t+1}(Z^{w,k}_{t+1}, B^k_{t+1})$. The overlap claim follows because Equation (14) also implies $A \in \mathcal{A}^k_{\mathrm{safe},t}(Z^{w,k}_t, B^k_t)$. □

This overlap result is mainly relevant for discrete action spaces or continuous implementations that reuse candidate actions across consecutive steps. With independent continuous resampling, the exact overlap condition is less likely to hold.

A.4 Pessimistic Ensemble Cost Critics

This section describes how the ensemble cost critics are trained and how the pessimistic ensemble estimate used by the Q-barrier relates to the Bellman upper-bound error $\Delta^{+,k}_{\mathrm{bell},t}$ in Equation (6).

Critic training. The discounted, episode-masked targets and PopArt-normalized critic loss are defined in Equation (28). We use $M = 4$ cost critics, enumerate discrete target-policy actions, and use one action sample for continuous actions. The target takes the maximum over two target critics sampled with replacement. This training target is distinct from the undiscounted candidate-minimum residual in Equation (6).

Ensemble pessimism. The Q-barrier uses the ensemble maximum in the budget's cost units, $\hat{Q}_C^+(Z_t^{w,k}, A) \doteq \max_{i\in\{1,\dots,M\}} \hat{Q}_{C,i}(Z_t^{w,k}, A)$. The barrier margin is then computed as $b_{Q,t}^k(Z_t^{w,k}, B_t^k, A) = B_t^k - \hat{Q}_C^+(Z_t^{w,k}, A)$. This worst-case aggregation makes the shield conservative when critic heads disagree, reducing the chance that an action is treated as safe because of a single underestimated cost prediction.

Effect on $\Delta_{\mathrm{bell},t}^k$. The pessimistic ensemble is a deployment-time heuristic intended to reduce $\Delta_{\mathrm{bell},t}^k$ empirically by biasing cost estimates conservatively in states where critic disagreement is large. We do not claim that it enforces Bellman conservatism theoretically. Instead, the role of the ensemble maximum is to make underestimation less likely in uncertain states; related empirical behavior is examined through the discounted critic diagnostics in Appendix B. Accordingly, this decoupled strategy may reduce the Bellman upper-bound error in practice, but it does not guarantee $\Delta_{\mathrm{bell},t}^k = 0$.

Environment	$e_{\text{pred}} \downarrow$	$e_V^+ \downarrow$	$\widehat{L}_Q^{\text{local},+} \downarrow$	Bellman sat. $\hat{Q}_C^+ \uparrow$	$\Delta_{\text{diag}}^+ \downarrow$
SafeDarkRoom	0.00252 ± 0.00002	0.3595 ± 0.0056	228.8 ± 3.0	72.2 ± 1.0	67.19 ± 3.20
SafeDarkMujoco-Point	0.00766 ± 0.00056	0.0847 ± 0.0053	14.10 ± 0.22	89.8 ± 0.9	0.248 ± 0.048
SafeDarkMujoco-Car	0.00906 ± 0.00058	0.0972 ± 0.0061	13.08 ± 0.14	89.8 ± 0.4	0.104 ± 0.004
SafeVelocity-HalfCheetah	0.0460 ± 0.0008	0.0432 ± 0.0009	0.872 ± 0.003	70.5 ± 0.2	0.386 ± 0.005
SafeVelocity-Ant	0.00478 ± 0.00013	0.0514 ± 0.0019	10.35 ± 0.14	89.9 ± 0.5	0.719 ± 0.032

Table 1: **Model and critic diagnostics on OOD trajectories.** We report latent prediction and value-perturbation diagnostics, together with discounted Bellman diagnostics. Bellman satisfaction is the percentage of transitions with $\Delta_{\text{diag},t} \leq 10^{-6}$. $\hat{Q}_C^+$ denotes the pessimistic critic used by the deployed Q-barrier. Residual magnitudes are reported in critic-scaled cost units. SafetyHalfCheetah has the largest prediction error, which is consistent with its more conservative reward-cost behavior under Q-Barrier shielding.

Environment	Bellman sat. mean $\uparrow$	Bellman sat. $\hat{Q}_C^+ \uparrow$	$\Delta_{\text{diag}}^{\text{mean}} \downarrow$	$\Delta_{\text{diag}}^+ \downarrow$
SafeDarkRoom	66.7 ± 1.1	72.2 ± 1.0	169.0 ± 8.2	67.19 ± 3.20
SafeDarkMujoco-Point	93.2 ± 0.6	89.8 ± 0.9	0.053 ± 0.007	0.248 ± 0.048
SafeDarkMujoco-Car	78.6 ± 0.5	89.8 ± 0.4	11.0 ± 0.51	0.104 ± 0.004
SafeVelocity-HalfCheetah	72.8 ± 0.2	70.5 ± 0.2	0.257 ± 0.004	0.386 ± 0.005
SafeVelocity-Ant	88.6 ± 0.5	89.9 ± 0.5	1.089 ± 0.038	0.719 ± 0.032

Table 2: **Effect of pessimistic critic aggregation on Bellman diagnostics.** We use the same layout in Table 1.

B Empirical Diagnostics for the Barrier-Margin Analysis

The barrier-margin bound in Theorem 1 depends on learned quantities that are not guaranteed by construction: one-step latent prediction error, local sensitivity of the learned cost critic, and the Bellman upper-bound residual of the critic. We therefore report related empirical diagnostics on evaluation trajectories using the deployed soft Q-barrier shield with $N_s = 8$ candidate actions. In the main text, $\hat{Q}_C^+$ denotes the pessimistic ensemble critic used by the deployed barrier. In this appendix, we also report diagnostics for the ensemble mean critic $\hat{Q}_C^{\text{mean}}$ to isolate the effect of pessimistic aggregation. In this subsection, critic values and costs use the same training scale s_C; we omit the scale superscript and k for simplicity.

For each selected transition $(Z_t^w, A_t, C_{t+1}, Z_{t+1}^w)$, we compute the one-step latent prediction error

$$e_{\text{pred},t} = \left\| f_z(Z_t^w, A_t) - Z_{t+1}^w \right\|_2 . \tag{18}$$

To evaluate the value perturbation induced by this latent error, we use the same next-step candidate set $\mathcal{A}_{t+1}^{\text{cand}}$ for both the predicted and realized next latent states:

$$\hat{V}_{C,t+1}^+(Z) = \min_{A' \in \mathcal{A}_{t+1}^{\text{cand}}} \hat{Q}_C^+(Z, A'). \tag{19}$$

The directly induced value perturbation is

$$e_{V,t}^+ = \left| \hat{V}_{C,t+1}^+(f_z(Z_t^w, A_t)) - \hat{V}_{C,t+1}^+(Z_{t+1}^w) \right| . \tag{20}$$

This is the most direct empirical analogue of the value perturbation term in Lemma 1.

We also report a finite-difference proxy for local critic sensitivity:

$$\widehat{L}_{Q,t}^{\text{local},+} = \max_{A' \in \mathcal{A}_{t+1}^{\text{cand}}} \frac{\left| \hat{Q}_C^+(f_z(Z_t^w, A_t), A') - \hat{Q}_C^+(Z_{t+1}^w, A') \right|}{\left\| f_z(Z_t^w, A_t) - Z_{t+1}^w \right\|_2 + \epsilon}, \qquad \epsilon = 10^{-8}. \tag{21}$$

This quantity should be interpreted only as a local finite-difference diagnostic: it can become large when the latent prediction error in the denominator is very small. For this reason, we use $e_{V,t}^+$ as the primary diagnostic for the critic-scaled analogue of the value perturbation in the theorem.

Finally, we compute the discounted Bellman diagnostic:

$$\Delta_{\text{diag},t}^+ = \left[C_{t+1} + \gamma_* \chi_{t+1} \hat{V}_{C,t+1}^+(f_z(Z_t^w, A_t)) - \hat{Q}_C^+(Z_t^w, A_t) \right]_+ . \tag{22}$$

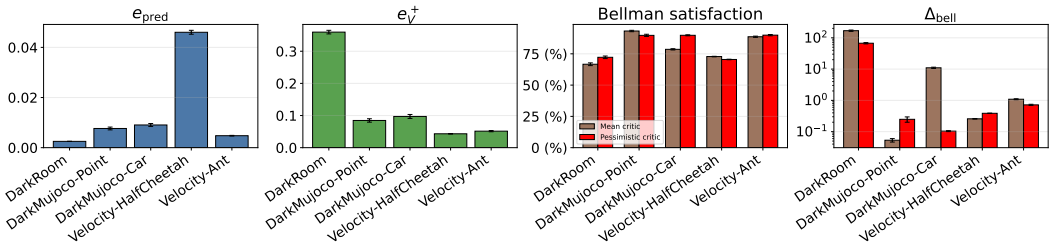


Figure 3: **Model and critic diagnostics on OOD evaluation trajectories.** We report one-step latent prediction error, induced value perturbation, Bellman upper-bound satisfaction, and positive Bellman residual. Error bars show standard error over rollout-level summaries. We compare the ensemble mean critic with the pessimistic critic used by the deployed Q-barrier. Bellman residuals are plotted on a log scale because their magnitudes differ across critic scales and environments.

Here χ_{t+1} is the evaluation bootstrap mask. Unlike Equation (6), this diagnostic includes discounting and critic cost scaling; it is not a direct measurement of that undiscounted residual. A transition satisfies this discounted upper-bound check when $\Delta^{+}_{\text{diag},t} \leq 10^{-6}$. We also compute the same residual for the ensemble mean critic $\hat{Q}_C^{\text{mean}}$ to isolate the effect of pessimistic aggregation.

All diagnostics are computed after action selection and are not used by the policy or shield during deployment. We first aggregate metrics within each evaluation rollout, using 50 tasks, and then report mean $\pm$ standard errors across rollouts.

Table 1 shows that the theorem-relevant quantities vary across environments. SafeHalfCheetah has the largest latent prediction error, which is consistent with the more conservative behavior observed in the main experiments. SafeDarkRoom has a small latent prediction error but a large local finite-difference sensitivity and a large Bellman residual in critic-scaled units, indicating that its learned critic changes sharply even under small latent perturbations. This is why we interpret $\widehat{L}_Q^{\text{local},+}$ as a diagnostic rather than as a verified global Lipschitz constant.

Table 2 isolates the effect of pessimistic critic aggregation on the Bellman upper-bound residual. The pessimistic critic improves Bellman satisfaction or reduces residual magnitude in SafeDarkRoom, SafetyCar, and SafetyAnt. The effect is not uniform: in SafetyPoint and SafetyHalfCheetah, pessimistic aggregation slightly lowers Bellman satisfaction and increases the positive residual. This is expected because max aggregation increases predicted cost and therefore acts as a deployment-time safety bias, but it does not by itself enforce Bellman upper-bound consistency on every transition.

These discounted, critic-scaled diagnostics describe learned-model and critic consistency, not direct measurements of the undiscounted bound in Theorem 1 or independent safety certificates. This distinction is important for SafetyHalfCheetah, where weaker Bellman diagnostics and larger latent prediction error are consistent with the more conservative reward-cost behavior observed in the main experiments.

C Hard Shield Variant

The main empirical results use the soft rule in Algorithm 2, which preserves support over the finite candidate set while downweighting actions with negative Q-barrier margin. Here, we define its hard truncation variant. The hard shield restricts action selection to the barrier-feasible subset of the candidate set whenever that subset is nonempty.

For the realized finite candidate set $\mathcal{A}_{t,k}^{\text{cand}}$, define the hard safe set

$$\mathcal{A}^k_{\text{safe},t}(Z_t^{w,k}, B_t^k) \doteq \left\{ A \in \mathcal{A}_{t,k}^{\text{cand}} : b^k_{Q,t}(Z_t^{w,k}, B_t^k, A) \geq 0 \right\} = \left\{ A \in \mathcal{A}_{t,k}^{\text{cand}} : \hat{Q}_C^{+}(Z_t^{w,k}, A) \leq B_t^k \right\}. \tag{23}$$

As in the soft shield, define the finite-candidate base weight

$$\rho_t(A) = \begin{cases} \pi_\theta(A \mid Z_t^{p,k}), & \text{if the discrete action space is enumerated,} \\ 1, & \text{if } A \in \mathcal{A}_{t,k}^{\text{cand}} \text{ is sampled from } \pi_\theta(\cdot \mid Z_t^{p,k}). \end{cases}$$

The hard-shield policy over the finite candidate set is

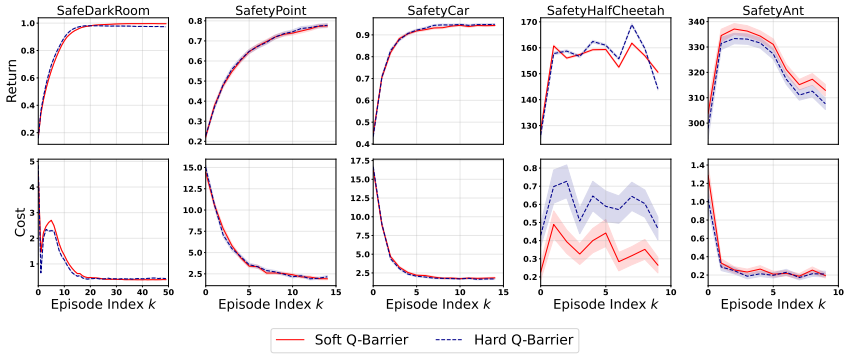
$$\pi_{\mathrm{hard}}(A \mid S_t^k, Z_t^k, B_t^k) = \begin{cases} \dfrac{\rho_t(A)\,\mathbb{I}\Big[A \in \mathcal{A}^k_{\mathrm{safe},t}(Z_t^{w,k}, B_t^k)\Big]}{\sum_{A' \in \mathcal{A}^{\mathrm{cand}}_{t,k}} \rho_t(A')\,\mathbb{I}\Big[A' \in \mathcal{A}^k_{\mathrm{safe},t}(Z_t^{w,k}, B_t^k)\Big]}, & \text{if the denominator} > 0, \\ \dfrac{\mathbb{I}\big[A \in \mathcal{A}^k_{\min,t}\big]}{|\mathcal{A}^k_{\min,t}|}, & \text{otherwise,} \end{cases} \tag{24}$$

where

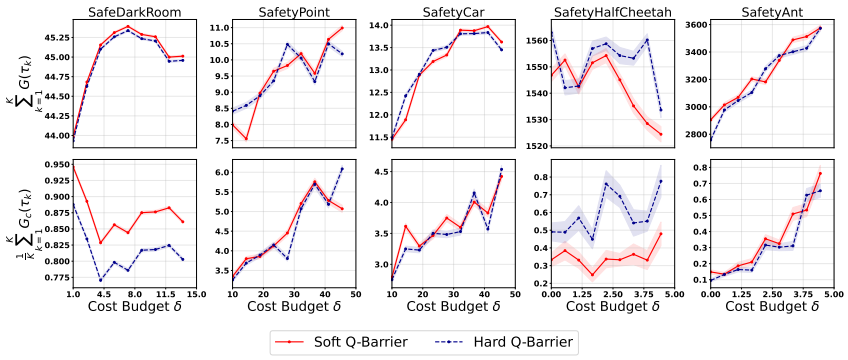
$$\mathcal{A}^k_{\min,t} \doteq \arg\min_{A' \in \mathcal{A}^{\mathrm{cand}}_{t,k}} \hat{Q}^+_C(Z_t^{w,k}, A'). \tag{25}$$

If the safe candidate set is nonempty, the hard shield renormalizes the finite-candidate base weights over barrier-feasible candidates only. If the safe set is empty, it falls back to the candidate with the smallest pessimistic predicted cost, with uniform tie-breaking if multiple candidates attain the minimum.

In discrete-action settings, $\rho_t(A)$ is the base-policy probability mass, so Equation (24) is a standard truncation and renormalization of the backbone policy. In continuous-action settings, the candidate actions are already sampled from the base policy, so $\rho_t(A) = 1$ and the hard shield is a finite-sample rejection/resampling rule over the sampled candidates. It should therefore be interpreted as a hard filter on the sampled candidate set, not as a truncation of the full continuous policy density.



(a) In-context adaptation dynamics



(b) Reward-cost tradeoffs under budget variation

Figure 4: **Soft versus hard Q-barrier shielding.** We compare the soft shield used in the main experiments with a hard truncation variant that selects only from barrier-feasible candidates when possible. The two variants have similar reward–safety profiles in most environments. Hard shielding gives marginal cost reductions in several environments, such as SafeDarkRoom, but these gains can come with small return loss and are not uniform. In SafetyHalfCheetah, hard shielding achieves the highest return but also increases cost, consistent with the larger latent prediction error reported in Table 1. This suggests that hard truncation is not automatically safer when the learned barrier is affected by model or critic error.

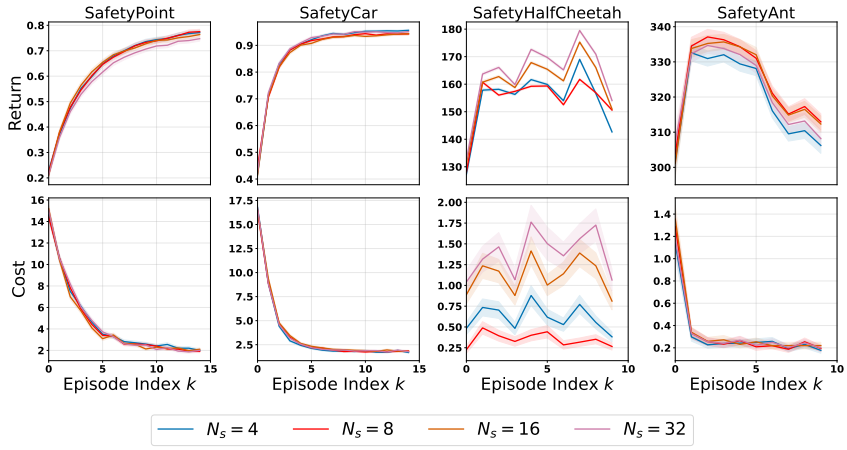
D Ablation: Hard Q-Barrier vs. Soft Q-Barrier

The main experiments use the soft Q-barrier shield in Algorithm 2. The soft shield preserves support over the finite candidate set by exponentially downweighting actions with negative barrier margin. In contrast, the hard shield from Algorithm 2 truncates the candidate distribution to actions that satisfy the learned barrier whenever such actions exist, and otherwise falls back to the candidate with the smallest pessimistic predicted cost. Both variants are deployment-time action-selection rules and use the same pretrained model, base policy, encoder, latent dynamics model, and pessimistic cost critic. We evaluate both shield types on the OOD tasks from the main experiments. The only difference between the two variants is the deployment-time action-selection rule: soft reweighting versus hard truncation. All other components are shared.

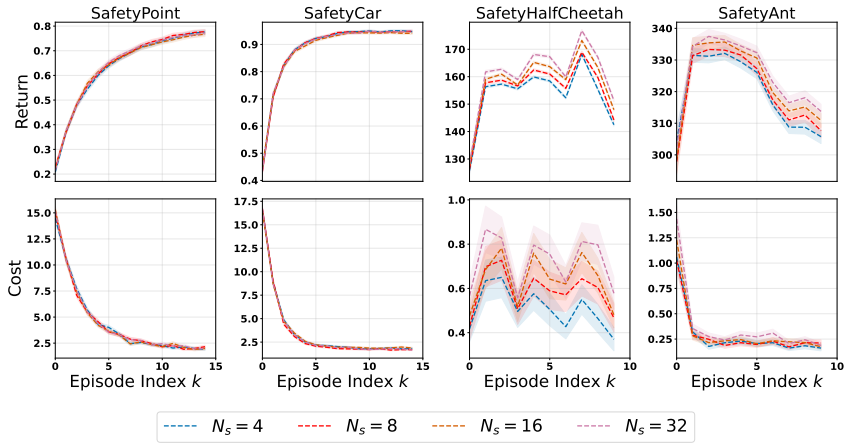
Figure 4 shows that hard and soft shielding perform similarly overall. In SafeDarkRoom, hard shielding slightly reduces cost relative to soft shielding, but with a small return drop, suggesting that the soft shield already assigns most probability to low-risk candidates.

SafetyHalfCheetah is the main exception: hard shielding achieves higher return than soft shielding, but also increases cost. This is consistent with the diagnostics in Table 1, where SafetyHalfCheetah has the largest latent prediction error. With less accurate latent prediction, hard thresholding can be less stable: small barrier errors may flip candidates between safe and unsafe, and fallback or truncation can amplify these mistakes. Soft shielding is smoother because it changes candidate probabilities continuously with the predicted barrier margin.

Overall, this ablation supports using the soft shield in the main experiments. Hard shielding can slightly reduce cost in some cases, but the gains are small and inconsistent. Soft shielding provides a more stable deployment-time tradeoff by biasing selection toward lower predicted cost while avoiding the discontinuities and empty-safe-set fallback behavior of hard filtering.



(a) Soft Q-Barrier



(b) Hard Q-Barrier

Figure 5: **Effect of candidate-set size on OOD adaptation dynamics.** We vary the number of sampled candidate actions $N_s \in \{4, 8, 16, 32\}$ used by our Q-barrier shield during deployment. **(a)** soft filtering, which reweights candidates by their barrier values, and **(b)** hard filtering, which removes candidates that fail the barrier test. Curves show per-episode return and cost over the in-context OOD evaluation horizon, with shaded regions denoting standard errors across rollouts.

E Number of Candidate Actions in Continuous Control

The Q-barrier shield evaluates a finite candidate set before selecting an action. In discrete-action environments this set is the full action space, but in continuous-action environments the shield approximates the budget-aware action distribution using N_s actions sampled from the base policy. This ablation studies how the number of sampled candidates affects the reward-safety tradeoff.

We evaluate $N_s \in \{4, 8, 16, 32\}$ on the continuous-control environments including SafeDarkMujoco and SafeVelocity domains. For each N_s, we compare the soft shield used in the main experiments with a hard shield that rejects candidate actions whose learned Q-barrier is negative whenever possible.

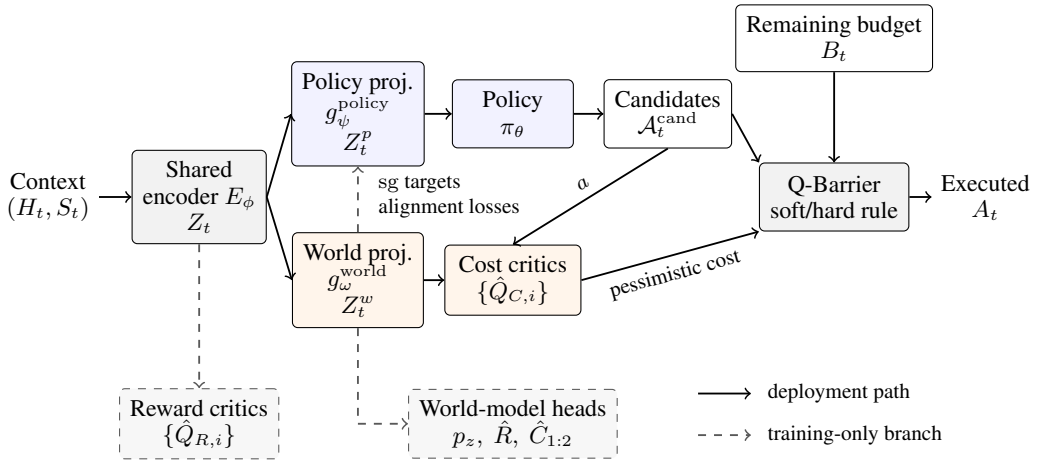
Figure 5 studies the effect of the number of sampled candidate actions N_s under soft and hard Q-barrier shielding. The sensitivity is environment dependent. In SafeDarkMujoco, both return and cost are stable across N_s for both filtering rules. Hard filtering (Figure 5b) is especially insensitive because it removes candidates that fail the learned barrier test, so larger candidate sets mainly affect which feasible action is selected. Soft filtering (Figure 5a) instead reweights all sampled candidates

according to their barrier values, making the selected action more susceptible to additional high-return but higher-cost samples.

The SafeVelocity environments show a stronger sampling effect. In SafetyHalfCheetah, increasing N_s improves return during adaptation, but also increases cost. This reward-cost tradeoff is more ordered under hard filtering: larger candidate sets consistently increase return, at the expense of higher cost, in both SafetyHalfCheetah and SafetyAnt. Under soft filtering, the ordering is less consistent because unsafe or near-unsafe candidates are downweighted rather than removed, leaving more dependence on the learned barrier scale. The stronger sensitivity in locomotion is expected, since these tasks have higher-dimensional action spaces and a finite candidate set more directly limits the shield's ability to find high-return safe actions. Overall, the ablation supports using $N_s = 8$ as a conservative default in the main experiments. We do not tune N_s per environment or report the best sampling number.

F Pseudocode

We provide pseudocode for the training procedure and the deployment-time action-selection rule used by the proposed shielding framework. Algorithm 1 summarizes the principal actor–critic and representation-learning objectives. Its critic notation follows the scaled-head convention in Appendix H.1. Algorithm 2 summarizes the runtime Q-barrier shielding rule used at deployment. Figure 6 illustrates the architecture and core loss routing as formulated in the paper.



Core loss	E_ϕ	g_ψ^{policy}	g_ω^{world}	π_θ	$\hat{Q}_R$	$\hat{Q}_C$	$p_z, \hat{R}, \hat{C}$
Actor	+	+	−	+	−	−	−
Critic TD	+	−	+	−	+	+	−
World-model	+	−	+	−	−	−	+
Distillation / conjugacy	−	+	−	−	−	−	−

Figure 6: **Conceptual architecture and core-loss routing.** The schematic follows the formulation in Section 3 and Appendix H.1. Arrows show forward or auxiliary-supervision paths; the table marks parameter updates implied by the displayed objectives (+: updated; −: not updated). The world model uses encoded replay transitions and actions, not deployment-time lookahead. Alignment losses detach encoder inputs and world-side targets; actor losses do not update critic parameters. Target-network copies are omitted for clarity, and all parameters are frozen at deployment.

Here $\mathcal{L}_{\text{DPG}}$ is the online policy-improvement loss, $\mathcal{L}_{\text{AWBC}}$ is the advantage-filtered behavioral cloning loss, and $\mathcal{L}_C^\pi$ is the Lagrangian cost penalty computed from the pessimistic cost critic. We use $M = 4$ critics, one target action sample, and $\alpha_{\text{BC}} = 0.1$. Bellman targets are computed with Polyak-averaged target networks and detached from the critic update. Discrete-action expectations are enumerated rather than sampled.

Algorithm 1 Training Procedure (Core Objectives Shown)

Require: Batch of history-conditioned trajectories $\mathcal{B}$

Initialize shared encoder E_ϕ, policy head π_θ, reward critic ensemble $\{\hat{Q}_{R,i}\}_{i=1}^{M}$, cost critic ensemble $\{\hat{Q}_{C,i}\}_{i=1}^{M}$, and target copies $\bar{\pi}_{\bar{\theta}}, \{\bar{Q}_{R,i}\}_{i=1}^{M}, \{\bar{Q}_{C,i}\}_{i=1}^{M}$

Initialize projection heads g_ω^{world}, g_ψ^{policy}

Initialize latent dynamics model p_z, reward head $\hat{R}$, and cost heads $\hat{C}_1, \hat{C}_2$

for each training update **do**

  Sample batch $\mathcal{B}$ of padded trajectories indexed by (k, t)

  Encode transitions: $Z_t^k \leftarrow E_\phi(H_t^k, S_t^k)$ and $Z_{t+1}^k \leftarrow E_\phi(H_{t+1}^k, S_{t+1}^k)$

  Project to world and policy latents:

$$Z_t^{w,k} \leftarrow g_\omega^{\text{world}}(Z_t^k), \quad Z_{t+1}^{w,k} \leftarrow g_\omega^{\text{world}}(Z_{t+1}^k)$$
$$Z_t^{p,k} \leftarrow g_\psi^{\text{policy}}(Z_t^k), \quad Z_{t+1}^{p,k} \leftarrow g_\psi^{\text{policy}}(Z_{t+1}^k)$$

  Compute policy outputs $\pi_\theta(\cdot \mid Z_t^{p,k})$

  Compute world-model outputs:

$$f_z(Z_t^{w,k}, A_t^k), \quad \text{predicted reward } \hat{R}_{t+1}^k, \quad \text{predicted costs } \hat{C}_{1,t+1}^k, \hat{C}_{2,t+1}^k$$

  Evaluate target-policy actions: enumerate discrete actions; sample once for continuous actions

  Compute detached reward/cost targets and normalized critic loss using Equation (28)

  Compute actor loss:

$$\mathcal{L}_{\text{actor}} \leftarrow \mathcal{L}_{\text{DPG}} + \alpha_{\text{BC}}\mathcal{L}_{\text{AWBC}} + \lambda_C\mathcal{L}_C^\pi \qquad \triangleright \text{ terms and masks: Equation (30)}$$

  Compute world-model loss $\mathcal{L}_{\text{wm}}$

  Compute distillation loss $\mathcal{L}_{\text{distill}}^{\text{sg}}$

  Compute conjugacy loss $\mathcal{L}_{\text{conj}}^{\text{sg}}$

$$\mathcal{L}_{\text{core}} \leftarrow \mathcal{L}_{\text{actor}} + 10.0\,\mathcal{L}_{\text{critic}} + 1.0\,\mathcal{L}_{\text{wm}} + 0.1\,\mathcal{L}_{\text{distill}}^{\text{sg}} + 0.1\,\mathcal{L}_{\text{conj}}^{\text{sg}}$$

  Update trainable parameters (core objectives shown)

  Update target networks by Polyak averaging

end for

Remark. SCARED uses the same base policy training pipeline as Q-Barrier, including the actor/critic updates, target networks, target-conditioning protocol, and training schedule. The difference is that Q-Barrier augments this backbone with shield-specific modules and losses: the world/policy projections, latent dynamics model, reward/cost prediction heads, structural alignment losses, and the runtime Q-barrier action-selection rule.

Notes on presentation. Here $\hat{Q}_{C,i}^{\text{sc}} = s_C\hat{Q}_{C,i}$ is the PopArt-denormalized prediction in critic-scaled cost units. The fixed budget factor is $\alpha = 1.25$ for SafeDarkRoom/SafeVelocity and 1.5 for SafeDarkMujoco, with $m_{\text{sh}} = 0$. The deployed margin $\tilde{b}_Q$ is distinct from the signed margin b_Q analyzed in Theorem 1. Algorithm 1 expands the base history-conditioned actor-critic forward path, making the interaction between policy optimization and auxiliary shielding objectives explicit. Algorithm 2 is the deployment-time rule used by the proposed method: the shield encodes the current state and context into shared and projected latents, computes the remaining budget, evaluates candidate actions with a pessimistic ensemble cost critic, and biases action selection toward actions with nonnegative barrier margin.

G Environment Details

Out-of-Distribution Environment Generation. For SafeDarkRoom and SafeDarkMujoco, we adopt the distance-based spawning protocol used in the safe ICRL benchmark of Moeini et al. [41]. This benchmark creates a controlled distributional shift by sampling training and evaluation environments from spatial distributions with opposite concentration patterns.

During training, obstacle and goal locations are sampled from a center-concentrated distribution,

$$p_{\text{train}}(x) \propto \exp(-\alpha d(x, c)),$$

Algorithm 2 Q-Barrier Shielding for Finite Candidate Action Selection

Require: State S_t^k, context H_t^k, budget δ, frozen policy π_{θ_*}, scaled ensemble critics $\{\hat{Q}_{C,i}^{\mathrm{sc}}\}_{i=1}^{M}$, shield type $q \in \{\mathrm{soft}, \mathrm{hard}\}$

Encode shared latent: $Z_t^k \leftarrow E_\phi(H_t^k, S_t^k)$
Project latents: $Z_t^{w,k} \leftarrow g_\omega^{\mathrm{world}}(Z_t^k)$, $Z_t^{p,k} \leftarrow g_\psi^{\mathrm{policy}}(Z_t^k)$
Compute remaining budget: $B_t^k \leftarrow \delta - \sum_{\tau=1}^{t} C_\tau^k$
Match critic units and adjust budget: $T_t^k \leftarrow [\alpha[s_C B_t^k]_+ - m_{\mathrm{sh}}]_+$
if action space is discrete **then**
    $\mathcal{A}_{t,k}^{\mathrm{cand}} \leftarrow \mathcal{A}$
    $\rho_t(A) \leftarrow \pi_{\theta_*}(A \mid Z_t^{p,k})$ for each $A \in \mathcal{A}_{t,k}^{\mathrm{cand}}$
else
    Sample $\mathcal{A}_{t,k}^{\mathrm{cand}} = \{A^{(j)}\}_{j=1}^{N_s}$ from $\pi_{\theta_*}(\cdot \mid Z_t^{p,k})$
    $\rho_t(A) \leftarrow 1$ for each $A \in \mathcal{A}_{t,k}^{\mathrm{cand}}$
end if
for each $A \in \mathcal{A}_{t,k}^{\mathrm{cand}}$ **do**
    $\hat{Q}_C^{\mathrm{sc},+}(Z_t^{w,k}, A) \leftarrow \max_i \hat{Q}_{C,i}^{\mathrm{sc}}(Z_t^{w,k}, A)$
    $\tilde{b}_{Q,t}^k(A) \leftarrow T_t^k - \hat{Q}_C^{\mathrm{sc},+}(Z_t^{w,k}, A)$
end for
if $q = \mathrm{soft}$ **then**
    for each $A \in \mathcal{A}_{t,k}^{\mathrm{cand}}$ **do**
        $w_t(A) \leftarrow \rho_t(A) \exp\Big(-[-\tilde{b}_{Q,t}^k(A)]_+\Big)$
    end for
    Normalize: $\pi_{\mathrm{soft}}(A) \leftarrow w_t(A) \big/ \sum_{A' \in \mathcal{A}_{t,k}^{\mathrm{cand}}} w_t(A')$
    return $A_t^k \sim \pi_{\mathrm{soft}}$
else
    $\mathcal{A}_{\mathrm{safe},t}^k \leftarrow \{A \in \mathcal{A}_{t,k}^{\mathrm{cand}} : \tilde{b}_{Q,t}^k(A) \geq 0\}$
    if $\mathcal{A}_{\mathrm{safe},t}^k \neq \emptyset$ **then**
        for each $A \in \mathcal{A}_{t,k}^{\mathrm{cand}}$ **do**
            $w_t(A) \leftarrow \rho_t(A) \mathbb{I}[A \in \mathcal{A}_{\mathrm{safe},t}^k]$
        end for
        Normalize: $\pi_{\mathrm{hard}}(A) \leftarrow w_t(A) \big/ \sum_{A' \in \mathcal{A}_{t,k}^{\mathrm{cand}}} w_t(A')$
        return $A_t^k \sim \pi_{\mathrm{hard}}$
    else
        $\mathcal{A}_{\mathrm{min},t}^k \leftarrow \arg\min_{A \in \mathcal{A}_{t,k}^{\mathrm{cand}}} \hat{Q}_C^{\mathrm{sc},+}(Z_t^{w,k}, A)$
        return $A_t^k \sim \mathrm{Uniform}(\mathcal{A}_{\mathrm{min},t}^k)$ ▷ lowest predicted-cost fallback
    end if
end if

where $d(x, c)$ denotes Euclidean distance from the map center c. Evaluation environments invert this distribution,

$$p_{\text{test}}(x) \propto \exp(\alpha d(x, c)),$$

forcing the agent to extrapolate into regions that are rarely observed during training. As discussed by Moeini et al. [41], increasing α sharpens the separation between training and test distributions and makes the resulting shift increasingly severe. We refer readers to that work for the full benchmark construction and derivations. For convenience, Figure 7 reproduces representative training and evaluation layouts.

Measuring OOD Generalization. Goal-discovery environments such as DarkRoom are widely used in prior ICRL work [34, 75, 56]. In those settings, test-time generalization is typically evaluated by holding out goal locations during training. However, such held-out tasks may still be solvable by interpolation across nearby training tasks [27].

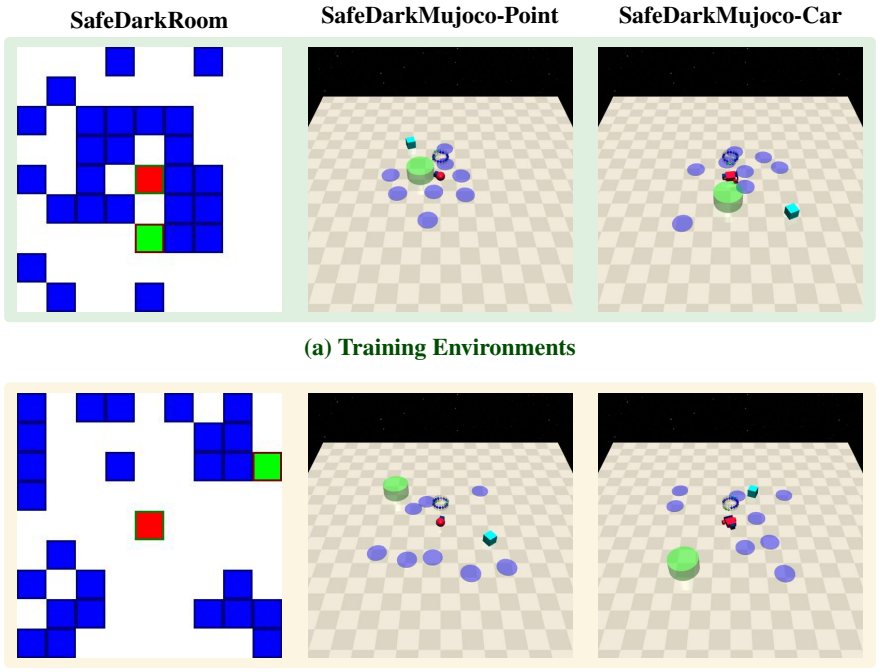


Figure 7: Visualization of the environment layouts, directly adapted from Moeini et al. [41]. The agent (red) must navigate to the target location (green) while avoiding obstacles (blue shades). To evaluate out-of-distribution generalization, training configurations sample goals and obstacles using a center-oriented distribution (α=0.5), whereas evaluation configurations strictly employ edge-oriented placements.

The structural OOD setup used here is stricter. Rather than withholding individual goals, we change the spatial distribution from which both goals and obstacles are sampled. During training, the agent starts near the map center and encounters center-oriented tasks more often. During evaluation, both goals and obstacles are drawn from edge-oriented distributions. This shift applies to both SafeDarkRoom and SafeDarkMujoco and is designed to test extrapolative generalization to regions that are systematically underrepresented during training.

Safe ICRL Benchmarks. SafeDarkRoom is a partially observed grid-world adaptation of DarkRoom [34]. The agent observes only its own position and cannot directly observe either the goal or the obstacles. Rewards are sparse, and costs arise from stepping on hazards distributed throughout the map. The agent must therefore infer both where to go and where not to go from the reward and cost signals accumulated in context.

SafeDarkMujoco is the continuous-control counterpart built on MuJoCo [58]. The robot observes internal physical states such as position, velocity, acceleration, and rotation angle, but does not receive direct sensing of the goal or obstacles. As in SafeDarkRoom, successful adaptation requires using sparse reward and cost observations from previous interaction to infer the hidden task structure.

SafeVelocity follows the Safety Gym style of safe locomotion tasks [25], but evaluation is performed on held-out target velocities from the same task family. We treat this benchmark as unseen ID generalization rather than structural OOD generalization because the deployment tasks are new parameter settings within the same family, not tasks sampled from a qualitatively shifted spatial distribution.

H Training Details

For the main Q-Barrier/SCARED comparisons, $K = 50$ in SafeDarkRoom, 15 in Point/Car, and 10 in SafeVelocity. SCARED curves use data supplied by its authors. Figure 1 shows smoothed

Parameter	SafeDarkRoom	SafeDarkMujoco & SafeVelocity
Episode time limit $t_{\max}$	30	75 (navigation), 200 (velocity)
Replay buffer capacity	100,000	100,000
Embedding dim	64	64
Hidden dim	64	64
Num layers	4	4
Num heads	8	8
Seq len	1500	1500
Attention dropout	0	0
Residual dropout	0	0
Embedding dropout	5	5
Learning rate	3×10^{-4}	3×10^{-4}
Betas	$(0.9, 0.99)$	$(0.9, 0.99)$
Clip grad	1.0	1.0
Batch size	32	32
Optimizer	Adam	Adam

Table 3: Hyperparameters for SCARED. Navigation uses 75 policy steps. Velocity uses 200 environment steps.

evaluation-level standard error of the mean. Figure 2 retains archived bands whose aggregation repeats trajectory summaries.

SCARED. We use the original implementation of Moeini et al. [41]. During SCARED pretraining, we resample a new environment from the training distribution every K episodes. For each sampled environment, we also sample a CTG target. The CTG range is $[1, 15]$ for SafeDarkRoom and $[10, 50]$ for SafeDarkMujoco.

SCARED follows the AMAGO-style history-conditioned architecture of Grigsby et al. [17]. An MLP time-step encoder maps each tuple (S_t, A_t, R_t, C_t) to an embedding, and a transformer-based trajectory encoder processes the resulting sequence. The prediction head outputs either an action distribution in discrete-action environments or a value prediction in continuous-action environments. Table 3 lists the remaining hyperparameters.

Q-Barrier. Shield-supporting modules, including the encoder E_ϕ, projection heads g_ω^{world} and g_ψ^{policy}, latent dynamics model p_z, and ensemble cost critic $\{\hat{Q}_{C,i}\}_{i=1}^{M}$, train jointly with SCARED during pretraining, without additional epochs. Discrete-action training collection uses the soft shield over the full action space. Continuous-action collection uses $N_s = 1$, so reweighting cannot change the base-policy action; representation and critic learning remain active. At test time, the frozen continuous-action shield uses $N_s = 8$ by default and applies Algorithm 2. The core loss weights in Equation (4) are fixed across all environments: $\lambda_{\text{critic}} = 10.0$, $\lambda_{\text{wm}} = 1.0$, $\lambda_{\text{dist}} = 0.1$, and $\lambda_{\text{conj}} = 0.1$. The remaining backbone hyperparameters follow Table 3.

SafeMeta and MAML with penalty. We use the original implementations of Xu and Zhu [68] with three hidden layers of sizes $(64, 512, 64)$ for the policy, value, and cost networks, for a total of 205,510 parameters. We train for 15k meta-training iterations, sampling 20 tasks from the training distribution at each iteration. During meta-testing, both algorithms perform K gradient updates to adapt to new tasks with unseen cost-to-go values.

Safe Algorithm Distillation. We use the original implementation of Moeini et al. [41]. For Safe AD, we collect a dataset $\mathcal{D} = \{\Xi_i\}$ of learning histories, where each trajectory $\Xi_i \doteq (\tau_1, \tau_2, \ldots, \tau_K)$ is a sequence of episodes generated by running safe RL algorithms on CMDPs. Each episode τ_k contains states, actions, rewards, and costs.

We train the Safe AD policy autoregressively to distill the behavior present in these logged learning histories, following Laskin et al. [34]. Each training example contains a trajectory segment together with RTG and CTG conditioning. The policy takes as input $(S_t^k, H_t^k, G_t(\tau_k), G_{c,t}(\tau_k))$ and predicts the action at time t in episode k.

Parameter	SafeDarkRoom	SafeDarkMujoco & SafeVelocity
Episode time limit $t_{\max}$	30	75 (navigation), 200 (velocity)
Hidden layers	$(64, 512, 64)$	$(64, 512, 64)$
Min/max batch size	500/1500	1500/1500
Policy learning rate	10^{-3}	10^{-3}
Value/cost learning rate	$3 \times 10^{-2}/10^{-1}$	$3 \times 10^{-2}/10^{-1}$
Discount factor γ	0.99	0.99
GAE parameter τ	0.95	0.95
Max KL divergence	10^{-3}	10^{-3}
Lagrangian weight λ	1.0	1.0
Num meta-training iterations	15k	15k
Policy optimizer	Adam	Adam
Value/cost optimizer	L-BFGS	L-BFGS

Table 4: Hyperparameters for SafeMeta and MAML with penalty. Episode-step conventions match Table 3.

Parameter	SafeDarkRoom	SafeDarkMujoco & SafeVelocity
Embedding dim	64	512
Hidden dim	512	256
Num layers	8	8
Num heads	8	8
Seq len	100	200
Attention dropout	0.5	0.5
Residual dropout	0.1	0.1
Embedding dropout	0.3	0.3
Learning rate	3×10^{-4}	3×10^{-4}
Betas	$(0.9, 0.99)$	$(0.9, 0.99)$
Clip grad	1.0	1.0
Batch size	512	128
Optimizer	Adam	Adam

Table 5: Hyperparameters for Safe Algorithm Distillation.

Discrete action spaces. For environments with discrete actions, the model outputs categorical logits and is trained with cross-entropy:

$$\mathcal{L}_{\text{disc}}(\theta) = \mathbb{E}_{\Xi_i \sim \mathcal{D}} \left[-\log \pi_\theta \big(A_t^k \mid S_t^k, H_t^k, G_t(\tau_k), G_{c,t}(\tau_k) \big) \right]. \tag{26}$$

Continuous action spaces. For environments with continuous actions, the model predicts an action mean and is trained with an ℓ_2 regression loss:

$$\mathcal{L}_{\text{cont}}(\theta) = \mathbb{E}_{\Xi_i \sim \mathcal{D}} \left[\left\| A_t^k - \mu_\theta \big(S_t^k, H_t^k, G_t(\tau_k), G_{c,t}(\tau_k) \big) \right\|_2^2 \right]. \tag{27}$$

These objectives let the transformer absorb constraint-aware goal-seeking behavior into its forward pass.

Dataset collection. We follow the same training protocol from [41]. We collect Safe AD data by running PPO-Lagrangian [50, 47] on the training environments. For SafeDarkRoom, we vary the cost limit across $\{0, 2.5, 5.0\}$ and collect 50,000 steps of learning history for each setting. For SafeDarkMujoco and SafeVelocity, we use larger datasets of 2–5 million environment steps to capture a broader range of safety-relevant behavior in continuous control. In these continuous-control environments, we keep the cost limit fixed at 0 during data collection so that the logged histories cover the full trajectory from high-cost exploration to near-feasible behavior later in training. This larger and more behaviorally diverse dataset is necessary because conditioning jointly on RTG and CTG induces a large space of reward-cost tradeoffs that the model must represent.

Compute resources. All experiments were run on NVIDIA A100 (80GB) or H100 (80GB) GPUs. SCARED pretraining uses 3K epochs for SafeDarkRoom, and SafeDarkMujoco, and 2K for SafeVelocity, with the shield-supporting modules (encoder, projection heads, latent dynamics, ensemble cost

critic) trained jointly during the same pretraining phase. Safe AD data collection via PPO-Lagrangian requires 2–5 million environment steps per environment. Safe meta-RL baselines (SafeMeta, MAML with penalty) train for up to 15K meta-training iterations.

H.1 Core Loss Definitions

We detail the principal terms displayed in Equation (4). The reward actor–critic update, filtered behavioral cloning, PopArt normalization, and multi-discount training follow AMAGO [17]; the cost objectives extend this backbone following SCARED [41]. We suppress the episode index and write z_t for a head's latent input. Expectations average over nonpadding replay transitions and, for actor–critic terms, the trained discount heads. We write one discount γ explicitly, while deployment uses $\gamma_* = 0.999$.

Critic losses. Let $\hat{Q}_{h,i}$ be critic i for $h \in \{R, C\}$ in scaled reward/cost units, and let $\bar{Q}_{h,i}$ and $\bar{\pi}$ denote Polyak-averaged target heads. Here R, C, δ are before critic scaling; the locally denoted scaled cost head equals s_C times the critic used in the barrier analysis. Write $\mathcal{N}_h(x) = (x - \mu_h)/\sigma_h$ for PopArt normalization and $q_{h,i} = \mathcal{N}_h(\hat{Q}_{h,i})$ for the normalized prediction used for optimization. With immediate reward/cost scales s_R, s_C, the targets are

$$\begin{aligned} Y_t^R &= s_R R_{t+1} + \gamma(1 - D_{t+1}) \min_{j \in J_R} \mathbb{E}_{a' \sim \bar{\pi}(\cdot|z_{t+1})}[\bar{Q}_{R,j}(z_{t+1}, a')], \\ Y_t^C &= s_C C_{t+1} + \gamma(1 - D_{t+1}^{\text{ep}}) \max_{j \in J_C} \mathbb{E}_{a' \sim \bar{\pi}(\cdot|z_{t+1})}[\bar{Q}_{C,j}(z_{t+1}, a')], \\ \mathcal{L}_{\text{critic}} &= \mathbb{E}_t \left[\frac{1}{M} \sum_{i=1}^{M} \sum_{h \in \{R,C\}} \left(q_{h,i}(z_t, a_t) - \text{sg}(\mathcal{N}_h(Y_t^h)) \right)^2 \right]. \end{aligned} \tag{28}$$

Here a_t is the replay action, D is the replay trajectory's termination flag, and D^{ep} is its inner-episode reset flag. Cost bootstrapping therefore stops at each episode boundary. Each J_h contains two critic indices sampled with replacement. Reward minima limit overestimation; cost maxima discourage underestimation. These targets use the *observed* next encoded state, not a latent-model rollout. Target evaluation and PopArt statistic updates are detached. The TD losses update the online critics and encoder. Target-head Polyak averaging uses $\tau = 0.003$.

Actor losses. Define each normalized critic's policy value and the replay-action advantage as

$$v_{h,i}(z_t) = \mathbb{E}_{a \sim \pi_\theta(\cdot|z_t)}[q_{h,i}(z_t, a)], \qquad A_t = \frac{1}{M} \sum_i \big(q_{R,i}(z_t, a_t) - v_{R,i}(z_t) \big). \tag{29}$$

The actor terms are

$$\begin{aligned} \mathcal{L}_{\text{DPG}} &= -\mathbb{E}_t[u_t \min_i v_{R,i}(z_t)], \\ \mathcal{L}_{\text{AWBC}} &= -\mathbb{E}_t[\text{sg}(\mathbf{1}\{A_t > 0\}) \log \pi_\theta(a_t \mid z_t)], \\ \mathcal{L}_C^\pi &= \mathbb{E}_t[u_t m_t \max_i v_{C,i}(z_t)], \qquad \mathcal{L}_{\text{actor}} = \mathcal{L}_{\text{DPG}} + 0.1\mathcal{L}_{\text{AWBC}} + \lambda_C \mathcal{L}_C^\pi. \end{aligned} \tag{30}$$

DPG improves reward using a conservative critic value. AWBC stabilizes off-policy learning by cloning only replay actions with positive estimated advantage; its binary filter is detached. The cost term discourages high predicted cost on the transitions selected by the training masks. The mask m_t maps episode-cost labels to replay transitions: a label is $+1$ for total cost above δ, -1 for total cost below $(1 - \eta)\delta$, and zero otherwise, where η is the maximum unused budget fraction. The supplied configurations use $\eta \geq 1$, so nonnegative costs do not activate the -1 branch. The optional detached mask $u_t \in \{0, 1\}$ excludes online reward and cost terms when the minimum cost-critic value at an episode's start, using γ_* and scaled cost units, exceeds $s_R\delta$, otherwise $u_t = 1$. This inherited mask uses the reward scaling factor s_R, even if s_C differs. AWBC uses neither mask. The fixed λ_C is specified in the environment configuration.

For discrete actions, expectations are exact sums over policy probabilities; for continuous actions, they use one reparameterized sample. The critic min/max is taken *after* this expectation. Actor gradients pass through the proposed action or probabilities; actor optimization does not update critic parameters, and the critic's latent input is detached. The policy-input path still updates the encoder. Continuous-action BC log probabilities are summed over action dimensions.

World-model loss. Let $w_t = g_\omega^{\text{world}}(Z_t)$. The latent model predicts a mean $\mu_t = f_z(w_t, a_t)$ and coordinate log-variances ℓ_{tj}, clipped to $[-10, 5]$. With d_m coordinates and two immediate-cost heads,

$$\mathcal{L}_{\text{wm}} = \mathbb{E}_t \left[\frac{1}{2d_m} \sum_{j=1}^{d_m} \left(e^{-\ell_{tj}} (\mu_{tj} - w_{t+1,j})^2 + \ell_{tj} \right) + (\hat{R}_{t+1} - R_{t+1})^2 + \sum_{i=1}^{2} (\hat{C}_{i,t+1} - C_{t+1})^2 \right]. \tag{31}$$

The first term is the coordinate-averaged Gaussian negative log likelihood, omitting its constant. The other terms predict observed reward and cost without the critic's scaling factors; each cost head has its own regression loss. Both current and next world latents remain differentiable. These objectives train the encoder, world projection, and prediction heads to retain transition-relevant information, without model-based planning at deployment.

Structural alignment losses. Distillation matches projected states. Conjugacy matches their discounted temporal changes. Let $p_t = g_\psi^{\text{policy}}(\text{sg}(Z_t))$. The reduced losses are

$$\begin{aligned} \mathcal{L}_{\text{distill}}^{\text{sg}} &= \mathbb{E}_t \left[d_m^{-1} \| p_t - \text{sg}(w_t) \|_2^2 \right], \\ \mathcal{L}_{\text{conj}}^{\text{sg}} &= \mathbb{E}_t \left[d_m^{-1} \| p_{t+1} - \gamma_* p_t - \text{sg}(w_{t+1} - \gamma_* w_t) \|_2^2 \right]. \end{aligned} \tag{32}$$

Conjugacy encourages consistent descriptions of an observed transition; it does not predict the next state. Detaching the encoder inputs and world-side targets makes both losses update only the policy projection.

Loss weights and implementation scope. The core weights in Equation (4) are $(\lambda_{\text{critic}}, \lambda_{\text{wm}}, \lambda_{\text{dist}}, \lambda_{\text{conj}}) = (10, 1, 0.1, 0.1)$. The displayed objectives and training pseudocode summarize the principal losses; the released code and configurations specify the complete auxiliary objective, including training-only implementation terms omitted here. Optional latent-smoothness and random-network-distillation losses in the code have zero weight in these configurations.

I Additional Shielding Ablations

I.1 Component Ablations

Table 6 summarizes the contributions of dynamics-based representation learning, runtime shielding, and the cost-critic ensemble on SafeDarkRoom. The latent dynamics model is used only during training to shape the representation used by the cost critic, not for model rollouts at deployment.

Method	Average Return ↑	Average Cost ↓
Full Q-Barrier ($M = 4$)	0.898 ± 0.003	0.920 ± 0.034
Two critics ($M = 2$)	0.836 ± 0.006	1.360 ± 0.045
Single critic ($M = 1$)	0.770 ± 0.011	1.240 ± 0.034
Barrier without dynamics learning	0.770 ± 0.009	1.747 ± 0.041
Dynamics learning without barrier	0.808 ± 0.007	2.002 ± 0.066
SCARED	0.625 ± 0.012	1.660 ± 0.010

Table 6: Component ablations on SafeDarkRoom, reproducing the mean $\pm$ reported standard error of the mean over 50 in-context episodes and 100 evaluation rollouts. The no-dynamics variant removes the associated auxiliary representation objectives.

Full Q-Barrier has the highest reported return and lowest cost. Even one critic outperforms SCARED in both metrics, so ensembling alone does not explain the improvement. Increasing M improves return, although cost is not monotonic in ensemble size. Removing dynamics-based representation learning or the barrier worsens both metrics relative to the full model, supporting their complementary roles.

I.2 Comparison with Cost-Penalty Reweighting

We compare our barrier penalty with a simpler rule that penalizes predicted cost no matter how much budget is left. Using the scaled critic and threshold from Algorithm 2, the two SafeDarkRoom rules

weight each action as

$$w_{\text{barrier}}(a) = \pi_\theta(a \mid z_t) \exp\{-\beta[\hat{Q}_C^{\text{sc},+}(z_t, a) - T_t]_+\},$$
$$w_{\text{penalty}}(a) = \pi_\theta(a \mid z_t) \exp\{-\beta \hat{Q}_C^{\text{sc},+}(z_t, a)\},$$

and the weights are normalized over actions. Each model is trained with its own rule at $\beta = 1$, using only the base actor–critic objectives. We then vary β at evaluation time without retraining.

	Cost penalty		Barrier	
β	Return ↑	Cost ↓	Return ↑	Cost ↓
0.1	0.647	2.043	0.780	1.816
0.5	0.644	1.972	0.765	1.800
1.0	0.633	1.833	0.769	1.750
2.0	0.590	1.633	0.593	1.489
5.0	0.434	1.379	0.382	1.057
10.0	0.128	0.859	0.218	0.948

Table 7: Cost-penalty versus barrier reweighting on SafeDarkRoom, in the same return and cost units as the main results.

At $\beta = 1$, the barrier gets higher return (0.769 versus 0.633) and slightly lower cost (1.750 versus 1.833). A larger β lowers both return and cost for either rule, so the two metrics have to be read together, and neither rule is best at every β. The barrier does not penalize actions whose predicted cost fits in the remaining budget, while the cost penalty keeps favoring low cost even when the budget is large.

I.3 Executed-Action Barrier Margins

The soft shield can execute an action with negative deployment margin $\tilde{b}_Q$ (Algorithm 2). Table 8 reports how often this occurs, separately over all decisions and those with nonnegative remaining budget.

Benchmark	Negative executed margin all decisions	Negative executed margin when $B_t \geq 0$
SafeDarkRoom	1.6%	0.9%
SafetyPointGoal	1.3%	0.2%
SafetyCarGoal	2.0%	0.4%
SafetyAnt	4.5%	2.7%
SafetyHalfCheetah	25.8%	22.5%

Table 8: Executed-action negative-margin frequencies reported during the rebuttal, using the implemented margin $\tilde{b}_Q$. DarkRoom, Point, and Car each use 15,000 logged decisions. Ant and HalfCheetah each use 2,000,000.

HalfCheetah remains the outlier even with nonnegative remaining budget. A negative margin is a learned risk flag, not a realized budget violation; these frequencies do not establish calibration or correlation with future violations. Point/Car execution and candidate-feasibility rates come from different logged runs; we do not subtract these tables to infer joint events.

I.4 Candidate-Set Feasibility

We also check how often at least one sampled candidate passes the barrier ($\tilde{b}_Q \geq 0$ in Algorithm 2), among decisions with nonnegative remaining budget ($B_t \geq 0$). Table 9 varies the number of proposals N_s from 1 to 16. With $N_s = 1$, reweighting cannot change the action.

The percentage barely changes with N_s. SafetyHalfCheetah is the only exception showing that about 20 percent of decisions has no barrier-passing candidate, and more candidates do not fix this

Benchmark	$N_s = 1$	$N_s = 4$	$N_s = 8$	$N_s = 16$
SafetyPointGoal	99.90	99.97	99.91	99.89
SafetyCarGoal	99.51	99.05	99.65	99.98
SafetyAnt	97.12	97.28	97.38	97.47
SafetyHalfCheetah	78.52	79.69	79.29	79.49

Table 9: Percentage of decisions with at least one barrier-passing candidate, among decisions with $B_t \geq 0$.

even though the number slightly increases. A passing margin only means the action looks feasible under the learned critic, and does not mean that the episode cost is safe. This result correlates to the conservativeness of Q-Barrier in the SafetyHalfCheetah domain.

I.5 Training Context Length

We compare Q-Barrier trained with 300- and 1,500-step context sequences (approximately 10 and 50 episodes). Both are evaluated on SafeDarkRoom for 50 episodes with a 1,500-step rolling context window.

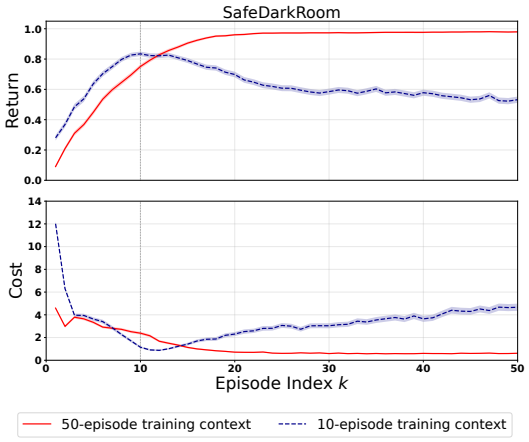


Figure 8: **Training-context ablation on SafeDarkRoom.** Per-episode return (top) and cost (bottom) for the two training context lengths. Shaded regions denote standard errors across rollouts. The dotted line marks $k = 10$, the approximate 10-episode equivalent of the shorter sequence at the 30-step episode cap.

The 300-step model performs better at $k = 10$ (return 0.834, cost 1.148, versus 0.751 and 2.384), but degrades substantially later in evaluation. At $k = 50$, it achieves a return of 0.532 and a cost of 4.664, against 0.980 and 0.617 for the 1,500-step model. These results highlight a limitation of context-length extrapolation: the short-context model's strong early adaptation does not persist when evaluation uses histories longer than its training context.

I.6 Extended Adaptation on Point and Car

The Point and Car policies use 1,500-step training context sequences (approximately 20 episodes). We run the frozen policies learnt by Q-Barrier for 100 episodes with the same 1,500-step window, without any retraining. Each episode has at most 75 policy steps, and the cost budget resets after every episode.

On Point, return drops and cost rises from $k = 20$ (0.642, 1.736) to $k = 100$ (0.339, 11.546). On Car, return and cost are 0.911 and 1.445 at $k = 20$, 0.039 and 0.203 at $k = 50$, and 0.045 and 2.343 at $k = 100$. Performance degrades substantially as adaptation extends beyond the training context's approximate 20-episode coverage, with a particularly severe return collapse on Car. This occurs even

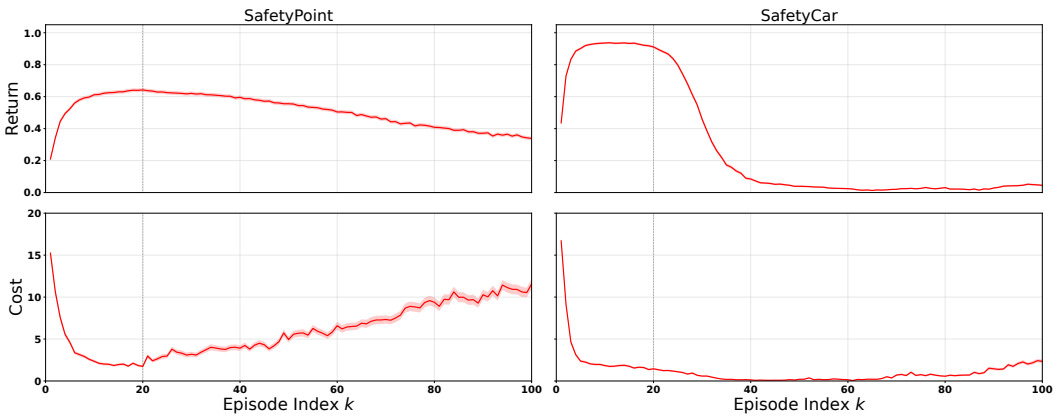


Figure 9: **Extended adaptation on Point and Car.** Per-episode return (top) and cost (bottom) over 100 episodes. Shaded regions denote standard errors across rollouts. The dotted line marks $k = 20$, the approximate episode coverage of the training context window.

with the rolling window fixed at 1,500 steps: keeping the input within the training context length does not by itself ensure sustained performance during longer deployment.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Section 1) state three main contributions: (i) barrier-based runtime shielding for safe ICRL deployment (Section 3), (ii) a conditional barrier-margin bound (Section 4), and (iii) empirical evaluation on safe ICRL deployment-generalization benchmarks (Section 5). The experimental claims are scoped to the five evaluated environments, covering structural OOD settings in SafeDarkRoom and SafeDark-Mujoco and unseen-ID target-velocity generalization in SafeVelocity. The theoretical claims are presented as conditional margin-propagation results whose dependence on learned model error, Bellman residuals, and candidate sets is stated explicitly.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 ("Limitations and future work") discusses dependence on latent prediction accuracy, critic calibration, Bellman residuals, conservative shielding behavior, and candidate-set quality. Appendix I adds empirical diagnostics and context-length limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 4 states the assumptions used by the main theoretical result, including approximate latent dynamics (Assumption 1) and latent Q-regularity (Assumption 2), and defines the Bellman upper-bound residual appearing in the bound. Theorem 1 is stated as a conditional one-step margin-propagation result over a fixed next-step candidate set, and Appendix A provides the proof with the supporting value-perturbation lemma (Lemma 1). Appendix A.2 also gives the episode-level extension (Proposition 1), and Appendix A.3 provides the conditional safe-set overlap lemma (Lemma 2).

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides the main information needed to reproduce the experimental setup: training and runtime shielding pseudocode (Algorithms 1–2, Appendix F), method architecture and losses (Section 3, Equation 4), hyperparameter tables for SCARED, SafeMeta/MAML, and Safe AD (Tables 3–5), and benchmark descriptions with the structural OOD and unseen-ID evaluation protocols (Appendix G). The public code release provides the Q-Barrier implementation, training/evaluation instructions, and plotting data.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Public code, pretrained checkpoints, and main-figure plotting data are available at https://github.com/kmj1122/q-barrier. The repository includes Q-Barrier and its unshielded constrained backbone, training/evaluation instructions, and plotting scripts; raw experiment archives are not redistributed. The paper also provides training and shielding pseudocode (Appendix F), hyperparameter tables (Appendix H), and benchmark descriptions including the structural OOD and unseen-ID evaluation protocols (Appendix G).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 describes training/evaluation protocols, task splits (OOD generalization tasks and unseen in-distribution tasks), OOD environment generation (center-oriented training vs. edge-oriented evaluation), CTG/RTG conditioning ranges, and how unseen

in-distribution tasks are designed. Appendix H provides full hyperparameter tables with optimizer (Adam/L-BFGS), learning rates, network architectures, batch sizes, sequence lengths, and training durations for SCARED (Table 3), SafeMeta/MAML (Table 4), and Safe AD (Table 5). Loss weights in Equation (4) are fixed across environments. Appendix G details the benchmark environments and OOD protocol.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main results in Figure 1 reports smoothed evaluation-level standard error of the mean. The diagnostics in Appendix B aggregate metrics within each evaluation rollout and report mean $\pm$ standard errors across rollouts. Additional quantitative ablations report uncertainty where applicable. We do not perform formal hypothesis testing. The reported standard errors are used to show variability of the evaluation estimates.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix H details the experimental setup, including the use of shared Slurm-managed GPU clusters with NVIDIA A100 80 GB and H100 80 GB GPUs for SCARED, Q-Barrier, and Safe AD. It also reports per-environment training scales such as SCARED pretraining epochs, Safe AD data-collection steps, optimization steps, and SafeMeta/MAML meta-training iterations.

   We do not provide exact wall-clock runtimes, storage requirements, or total GPU-hour estimates for every run. Experiments were executed on dynamically allocated shared hardware, making precise per-job resource accounting difficult.

For rough guidance, representative successful runs required approximately per algorithm:

(a) SafeDarkRoom: 4–6 days
(b) SafeDarkMuJoCo: 8 days
(c) SafeVelocity: 2–4 days

Failed runs roughly doubled these times. All experiments were conducted on institutional shared clusters. We did not use dedicated exclusive hardware.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The research uses only standard simulated RL benchmarks and does not involve human subjects, personally identifiable data, real-world deployment, or safety-critical applications. All datasets and environments are publicly available and used for academic purposes, consistent with the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: Section 7 discusses technical limitations and clarifies that the learned shield does not certify closed-loop safety. We do not provide a separate assessment of positive and negative societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our code and model assets concern simulated RL benchmarks rather than high-risk generative models or scraped datasets. They are research artifacts, not validated safety systems for real-world deployment.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: We credit the original papers and codebases for all reused benchmarks and implementations, including the Safe ICRL benchmark/SCARED [41], MuJoCo [58], Safety Gymnasium [25], AMAGO [17], SafeMeta [68]. Where public repositories were available and directly reused/adapted, we followed their terms:

    (a) Safety Gymnasium: Apache License 2.0

    (b) AMAGO: MIT License

    MuJoCo is under Apache License 2.0. For SCARED and SafeMeta, we relied on the papers and supplementary materials provided by the authors. Readers should consult the corresponding upstream repositories or documentation for the exact applicable licenses and usage terms, as we have not modified or redistributed the original assets beyond standard academic experimentation and citation.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The accompanying code is documented through the paper's pseudocode (Algorithms 1–2 in Appendix F), loss definitions and hyperparameters (Appendix H), and training and evaluation protocols (Section 5, Appendix G). The release includes environment configurations, training scripts, and evaluation scripts.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve any crowdsourcing or research with human subjects. All experiments are conducted entirely on simulated RL benchmarks (SafeDarkRoom, SafeDarkMujoco, SafeVelocity) using standard evaluation protocols.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve human subjects research. All work is conducted on simulated environments with no human participants.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as any component of the core methods in this research. The ideas provided in this paper is derived by authors. LLMs are used for writing, editing and plotting assistance.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.