Runtime Safety through Adaptive Shielding: From Hidden Parameter Inference to Provable Guarantees

Minjae Kwon
The University of Virginia
hbt9su@virginia.edu

Tyler Ingebrand
The University of Texas at Austin
tyleringebrand@utexas.edu

Ufuk Topcu
The University of Texas at Austin
utopcu@utexas.edu

Lu Feng
The University of Virginia
lf9u@virginia.edu

Abstract

Variations in hidden parameters, such as a robot's mass distribution or friction, pose safety risks during execution. We develop a runtime shielding mechanism for reinforcement learning, building on the formalism of constrained hidden-parameter Markov decision processes. Function encoders enable real-time inference of hidden parameters from observations, allowing the shield and the underlying policy to adapt online. The shield constrains the action space by forecasting future safety risks (such as obstacle proximity) and accounts for uncertainty via conformal prediction. We prove that the proposed mechanism satisfies probabilistic safety guarantees and yields optimal policies among the set of safety-compliant policies. Experiments across diverse environments with varying hidden parameters show that our method significantly reduces safety violations and achieves strong out-of-distribution generalization, while incurring minimal runtime overhead. Project page: https://kmj1122.github.io/adaptive-shielding-project-page/

1 Introduction

Robots and other autonomous systems must operate safely in open-world environments where the underlying dynamics can vary due to hidden parameters such as mass distribution, friction, or terrain compliance. These parameters often change across episodes and remain unobserved, introducing safety risks and challenging the generalization capabilities of reinforcement learning (RL) systems [Kirk et al., 2023, Benjamins et al., 2023]. Ensuring robust and safe behavior under such uncertainty is essential in domains like autonomous driving and robotic manipulation, where failures can have serious real-world consequences.

Despite recent progress in hidden parameter-aware and safe RL, existing methods often trade off adaptability and safety. Approaches such as hypernetworks, contextual models, or mixtures-of-experts [Rezaei-Shoshtari et al., 2023, Beukman et al., 2023, Celik et al., 2024] demonstrate strong adaptation to varying dynamics, but typically lack explicit mechanisms for guaranteeing safety under uncertainty. Conversely, safe RL frameworks based on constrained Markov decision processes [Wachi and Sui, 2020, Achiam et al., 2017] enforce safety by constraining cumulative costs, but often assume fixed dynamics and fail to adapt in real time to hidden parameter shifts. Methods like constrained policy optimization [Achiam et al., 2017, Yang et al., 2020] and shielding [Alshiekh et al., 2017, Yang et al., 2023] mitigate risk in static environments but remain limited in settings with dynamic or unobserved changes.

To address this gap, we propose a runtime shielding framework for reinforcement learning that adapts online to hidden parameters while offering provable probabilistic safety guarantees. Central to our approach is the use of function encoders [Ingebrand et al., 2024b, 2025], a compact and expressive

Preprint. Under review.

model class that infers environment dynamics from transition data by projecting them onto neural basis functions. This representation enables fast, online adaptation of both the policy and shield without retraining.

To ensure safe learning and adaptation, our approach combines two complementary mechanisms that operate proactively during training and reactively at execution. First, we introduce a safety-regularized objective that augments rewards with a cost-sensitive value estimate, encouraging the policy to avoid unsafe behavior during training. However, this objective alone cannot guarantee safety, particularly under distribution shift. To address this, we augment policy execution with an adaptive shield that samples candidate actions from the policy, predicts their next states using a function encoder, and applies conformal prediction to quantify uncertainty in these forecasts. Actions that fail to meet a safety margin are filtered out, ensuring that only safe actions are executed.

In summary, our main contributions are:

- **Online Hidden-Parameter Adaptation**: We leverage function encoders to infer hidden parameters from transitions, enabling efficient policy and shield adaptation without retraining.
- **Safety-Regularized RL Objective**: We propose a new objective that balances reward and safety by integrating a cost-sensitive value function, encouraging low-violation behavior during training.
- **Adaptive Shield with Probabilistic Guarantees**: We develop an adaptive, uncertainty-aware runtime shield that filters unsafe actions using conformal prediction, ensuring safety during execution with provable probabilistic guarantees.

Empirical evaluations in Safe-Gym benchmarks [Ji et al., 2023], including out-of-distribution scenarios with unseen hidden parameters, demonstrate that our method significantly reduces safety violations compared to baselines, achieving robust generalization with minimal runtime overhead.

1.1 Related Work

Safe Reinforcement Learning. Safe RL methods often employ constrained MDP formulations to ensure compliance with safety constraints. Constrained policy optimization (CPO) remains foundational, effectively balancing performance and safety [Achiam et al., 2017, Wachi and Sui, 2020]. Further techniques use learned recovery policy to ensure safe action execution [Thananjeyan et al., 2020]. Recent zero-violation policy methods in RL aim to minimize safety violations using techniques like genetic cost function search, energy-based action filtering, and primal-dual algorithms [Hu et al., 2023, Zhao et al., 2021, Ma et al., 2024, Liu et al., 2021, Bai et al., 2023]. However, these approaches often face scalability issues, rely on restrictive assumptions, or are limited to simple environments. Unlike these approaches, we introduce the safety regularized-objective that can be integrated into the optimization process of any CMDP-based RL algorithms. Shielding frameworks proactively filter unsafe actions, selectively sampling safe actions [Alshiekh et al., 2017, Carr et al., 2023, Yang et al., 2023]. Recent developments on shielding integrate adaptive conformal prediction into safety frameworks, enhancing uncertainty quantification for safety-critical planning [Sheng et al., 2024a,b]. However, unlike existing methods, which are not explicitly designed to address varying hidden dynamics, our approach concurrently enhances safety through a safety-regularized objective and adaptive shielding while adapting to dynamic hidden parameters using function encoders.

Contextual or Hidden-Parameter Reinforcement Learning. Hidden parameters, often termed context, have been studied in recent context-aware reinforcement learning approaches, demonstrating their importance for generalization [Benjamins et al., 2023]. When algorithms are provided with knowledge of the hidden parameters, they are often directly integrated into the model. For example, contextual recurrent state-space models explicitly incorporate known contextual information to enable zero-shot generalization [Prasanna et al., 2024]. Contextualized constrained MDPs further integrate context-awareness into safety-prioritizing curricular learning [Koprulu et al., 2025]. A common approach to handle unknown context information is to infer it from observational history using transformer models [Chen et al., 2021]. Hypernetwork-based methods utilize adapter modules to adjust policy networks based on inferred contexts [Beukman et al., 2023]. Mixture-of-experts architectures leverage specialized experts, using energy-based models to handle unknown contexts probabilistically [Celik et al., 2024]. However, these works primarily focus on enhancing generalization to varying dynamics without incorporating safety mechanisms during adaptation in contrast to our method.

Generalization in Reinforcement Learning. Generalization in RL, including zero-shot transfer and meta learning, is crucial for robust policy adaptation to varying dynamics. For example, meta-learning approaches, such as MAML [Finn et al., 2017], allow rapid parameter adaptation from minimal interaction data. Recent work in meta-safe RL [Khattar et al., 2023] has established provable guarantees for adapting to new tasks while satisfying constraints. However, meta-learning approaches involve parameter updates during adaptation, whereas our framework focuses on rapid, online inference of hidden parameters without requiring such updates. Hypernetwork-based zero-shot transfer methods explicitly condition policies on task parameters [Rezaei-Shoshtari et al., 2023]. Function encoders, i.e. neural network basis functions, have demonstrated strong zero-shot transfer by using the coefficients of the basis functions as a fully-informative, linear representation of the dynamics [Ingebrand et al., 2024b,a]. Single-episode policy transfer and adaptive methods effectively handle environment changes by encoding historical context [Yang et al., 2019, Chen et al., 2022]. Advanced context encoder designs further improve robustness and fast adaptation capabilities [Luo et al., 2022]. While these methods excel at adapting to varying dynamics, they do not address safety constraints during adaptation, leaving agents vulnerable to unsafe actions in unseen environments.

2 Problem Formulation

Constrained hidden-parameter MDPs (CHiP-MDPs) model environments with varying transition dynamics, where a cost function is introduced alongside a reward function to address safety constraints. A CHiP-MDP extends the HiP-MDP framework Konidaris and Doshi-Velez [2014] and is defined by the tuple $\mathcal{M} = (S, A, \Theta, T, R, C, \gamma, P_\Theta)$, where S and A are the state and action spaces, $R : S \times A \times S \to \mathbb{R}$ is a reward function, $C : S \times A \times S \to [0, 1]$ is a cost function, and $\gamma \in (0, 1)$ is the discount factor. The transition dynamics $T : S \times A \times \Theta \to S$ depend on a hidden parameter $\theta \in \Theta$. For a specified hidden parameter $\theta \in \Theta$, we denote the transition dynamics as $T_\theta : S \times A \to S$. The prior $P_\Theta(\theta)$ over the parameter space Θ represents the distribution of these hidden parameters. We denote the initial state distribution as μ_0.

The agent follows a policy $\pi : S \times \Phi \to A$, where Φ denotes the set of learned representations of the transition dynamics T. Each representation $b \in \Phi$ infers the transition dynamics T_θ for a given parameter θ. The representation b can be inferred from offline or online samples from T_θ or replaced with θ if direct access is available. The objective of the agent is to maximize expected cumulative discounted reward while satisfying safety constraints in a CHiP-MDP $\mathcal{M}$.

To formalize this objective, we define the reward action-value function, given a parameter θ, as:

$$Q_R^\pi(s, a, \theta) = \mathbb{E}_{\pi, T_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \mid s_0 = s, a_0 = a, \theta\right]. \tag{1}$$

The corresponding reward state-value function, which averages Q_R^π over actions, is:

$$V_R^\pi(s, \theta) = \mathbb{E}_{a \sim \pi(\cdot|s,b)}\left[Q_R^\pi(s, a, \theta)\right]. \tag{2}$$

Finally, the reward objective is defined as :

$$J_R(\pi) = \mathbb{E}_{\theta \sim P_\Theta, s_0 \sim \mu_0(\cdot|\theta), a_0 \sim \pi(\cdot|s_0,b)}\left[Q_R^\pi(s_0, a_0, \theta)\right]. \tag{3}$$

The safety constraints aim to minimize the average cost rate. To this end, we state our problem below.

Problem. Given a CHiP-MDPs $\mathcal{M} = (S, A, \Theta, T, R, C, \gamma, P_\Theta)$ where the transition dynamics T_θ are fully unknown and vary with a hidden parameter θ, find an optimal policy π^* that maximizes the expected cumulative discounted reward $J_R(\pi^*)$ while satisfying the safety constraints on the average cost rate,

$$\phi^{\pi^*}(s, \theta) = \lim_{H \to \infty} \frac{1}{H} \mathbb{E}_{\pi^*, T_\theta}\left[\sum_{t=0}^{H-1} C(s_t, a_t, s_{t+1}) \mid s_0 = s, \theta\right] \le \delta, \tag{4}$$

where $\delta \in (0, 1)$ is a failure probability. Note that the transition dynamics depend on the hidden parameter θ, but the policy depends on a representation of the hidden parameter derived from any previously observed transitions by T_θ.

Lagrangian Approach. To enforce the safety constraint, we define the cost action-value function Q_C^π and cost state-value function V_C^π by replacing the reward function R with the cost function C from Equations 1 and 2. Minimizing the average cost rate can be achieved by minimizing the cost-value function V_C^π (see Appendix I). To this end, we use a Lagrangian approach, optimizing the objective $\mathcal{L}(\pi, \lambda) = J_R(\pi) - \lambda(J_C(\pi) - \epsilon)$ with $\lambda \ge 0$, where $J_C(\pi) = \mathbb{E}_{\theta \sim P_\Theta, s_0 \sim \mu_0(\cdot|\theta)}[V_C^\pi(s_0, \theta)]$.

3 Background

We introduce key concepts essential for understanding our methods. First, function encoders have demonstrated robust performance in estimating varying underlying dynamics [Ingebrand et al., 2024b,a, 2025]. Second, conformal prediction provides a rigorous framework for quantifying uncertainty [Vovk et al., 2005, Tibshirani et al., 2019, Gibbs and Candès, 2024].

Function Encoder. A function encoder (FE) offers a compact and computationally efficient framework for representing functions in terms of neural network basis functions. Consider a set of functions $\mathcal{F} = \{f \mid f : \mathcal{X} \to \mathbb{R}\}$, where $\mathcal{X} \subset \mathbb{R}^n$ is an input space with finite volume. When $\mathcal{F}$ forms a Hilbert space with the inner product $\langle f, g \rangle = \int_{\mathcal{X}} f(x)g(x)dx$, any $f \in \mathcal{F}$ can be expressed using a basis $\{g_1, g_2, \ldots, g_k\}$ as $f(x) = \sum_{i=1}^{k} b_i g_i(x)$, where b_i are unique coefficients. To determine the coefficients, we solve the following least-squares optimization problem:

$$(b_1, b_2, \cdots, b_k) := \underset{(b_1, b_2, \cdots, b_k) \in \mathbb{R}^k}{\arg\min} \left\| f - \sum_{j=1}^{k} b_j g_j \right\|_2^2 . \tag{5}$$

For more information on how to train the neural network basis functions, see Ingebrand et al. [2025].

Conformal Prediction. Conformal Prediction (CP) allows for the construction of prediction intervals (or regions) that are guaranteed to cover the true outcome with a user-specified probability, under minimal assumptions. For exchangeable random variables $\{Z_i\}_{i=1}^{t+1}$, CP constructs a region satisfying: $\mathbb{P}(Z_{t+1} \leq \Gamma_t) \geq 1 - \delta$, where $\delta \in (0, 1)$ is the failure probability, and the threshold $\Gamma_t = Z_{(q)}$ is the q-th order statistic of $\{Z_1, \ldots, Z_t\}$, with $q = \lceil (t+1)(1-\delta) \rceil$. Adaptive Conformal Prediction (ACP) extends this to non-stationary settings by making the threshold learnable. For more information on conformal prediction, see Shafer and Vovk [2008], Gibbs and Candès [2021].

4 Approach

Our approach has three main components. First, we introduce a novel *safety-regularized objective*. This objective is used during optimization and encourages the policy to converge toward a zero-violation policy. Second, we use a function encoder to represent underlying dynamics T_θ, enabling *online adaptation*. Finally, we leverage this dynamics representation to construct an *adaptive shield*. The shield adjusts safe regions by conformal prediction and blocks unsafe actions online.

4.1 Safety-Regularized Objective

To promote safe policy learning, we introduce a safety measure, $Q^\pi_{\text{safe}}(s, a, \theta)$, which quantifies the relative safety of action a sampled from the policy π in state s under parameter θ. Higher values of Q^π_{safe} indicate actions with lower long-term costs under policy π. Since we aim to minimize the cost action-value function Q^π_C, higher Q^π_{safe} values correspond to lower Q^π_C values. Based on this intuition, we define $Q^\pi_{\text{safe}}(s, a, \theta)$ for an action a sampled by $\pi(\cdot \mid s, b)$ as:

$$Q^\pi_{\text{safe}}(s, a, \theta) = 1 - \frac{\pi(a \mid s, b) Q^\pi_C(s, a, \theta)}{V^\pi_C(s, \theta) + \epsilon} \tag{6}$$

where $b \in \mathbb{R}^k$ is a learned representation of T_θ, and $\epsilon > 0$ is a small constant for numerical stability.

This formulation bounds the value in $(0, 1]$ by its design. A value of 1 for $Q^\pi_{\text{safe}}(s, a, \theta)$ indicates two scenarios: 1) Safety: The policy selects an action a resulting in zero long-term cost violations, i.e., $Q^\pi_C(s, a, \theta) = 0$; or 2) Exploration: The probability of selecting action a is small. In contrast, a value near 0 for $Q^\pi_{\text{safe}}(s, a, \theta)$ indicates that action a substantially contributes to the expected cumulative cost $V^\pi_C(s, \theta)$, posing a higher risk compared to other action choices at state s. See Appendix G for details on the design choice.

To integrate this safety measure into policy optimization, we define an augmented action-value function, $Q^\pi_{\text{aug}}(s, a, \theta) = Q^\pi_R(s, a, \theta) + \alpha Q^\pi_{\text{safe}}(s, a, \theta)$, where $Q^\pi_R(s, a, \theta)$ is the reward action-value function, and $\alpha \geq 0$ is a hyperparameter balancing safety and reward. Our *safety-regularized objective* (SRO) is:

$$J_{\text{safe}}(\pi) = \mathbb{E}_{\theta \sim P_\Theta, s_0 \sim \mu_0(\cdot|\theta), a_0 \sim \pi(\cdot|s_0, b)} \left[Q^\pi_{\text{aug}}(s_0, a_0, \theta) \right] . \tag{7}$$

A larger α encourages the policy to prioritize safe actions that result in zero-violation costs or to select under-explored actions with lower assigned probabilities. Next, we introduce a proposition which justifies this choice of objective.

Proposition 1. Let $\Pi_{\text{zero-violation}}$ denote the set of zero-violation policies, defined as $\{\pi \mid J_C(\pi) = 0\}$. Then, for any $\alpha \geq 0$, the optimal policy obtained by maximizing the safety-regularized objective function $J_{\text{safe}}(\pi)$ within $\Pi_{\text{zero-violation}}$ is equivalent to the optimal policy obtained by maximizing the standard reward objective $J_R(\pi) = \mathbb{E}_{\theta \sim P_\Theta, s_0 \sim \mu_0(\cdot|\theta), a_0 \sim \pi(\cdot|s_0, b)}[Q_R^\pi(s_0, a_0, \theta)]$ within the same set of policies.

Proof Sketch. For any policy within the zero-violation set, all actions sampled from the policy lead to $Q_C^\pi(s, a, \theta) = 0$. By our design of the safety term, this condition implies $Q_{\text{safe}}^\pi(s, a, \theta) = 1$. Substituting this into our regularized objective, $J_{\text{safe}}(\pi)$ simplifies to $J_R(\pi) + \alpha$. Therefore, for zero-violation policies, maximizing $J_{\text{safe}}(\pi)$ is equivalent to maximizing the standard reward objective $J_R(\pi)$, as α is a constant. □

Proposition 1 is significant because it proves that the safety regularization does not degrade performance unnecessarily when an agent already behaves safely. Specifically, it guarantees that if we focus only on the set of policies that satisfy all safety constraints, maximizing the safety-regularized objective is equivalent to maximizing the standard reward objective.

4.2 Inferring Hidden Parameters Online

To infer the underlying dynamics T_θ and predict the next state s_{t+1} based on transition samples and (s_t, a_t), we use a function encoder, denoted by $\hat{f}_{\text{FE}}$. Given observed transition samples $\{(s_i, a_i, s_{i+1})\}_{i=1}^{t-1}$ and the current state-action pair (s_t, a_t), the function encoder predicts the next state $\hat{s}_{t+1}$ as: $\hat{s}_{t+1} = \hat{f}_{\text{FE}}(s_t, a_t) = \sum_{i=1}^{k} b_i \cdot g_i(s_t, a_t)$, where $g_i(s_t, a_t)$ are pretrained basis functions, and b_i are coefficients derived from a subset of transition samples. These coefficients b_i additionally serve as a representation for T_θ. Due to the properties of basis functions, these representations are fully informative and linear Ingebrand et al. [2024b]. We concatenate them with the state to form an augmented input $(s_t, b_1, \ldots, b_k)$, which the policy uses as input. As the agent interacts with the environment, collecting new transitions (s_t, a_t, s_{t+1}), we refine the coefficients b_i by solving Equation 5 with updated transition samples. Consequently, the agent receives an online representation of the dynamics. Note that with a fixed number of basis functions k, the computation, involving the inverse of a $k \times k$ matrix, remains efficient even for large samples. We denote the coefficients $(b_1, \cdots, b_k)$ as b.

4.3 Adaptive Shielding Mechanism

To ensure safety during policy execution, we propose an adaptive shielding mechanism that dynamically intervenes based on uncertainty in model predictions. This shield wraps any underlying policy π, adjusting actions to prevent unsafe outcomes. We illustrate the shielding process at timestep t. Given that the safety-regularized objective mitigates long-term cost violations, we adopt one-step prediction to minimize runtime overhead. For complex tasks, multi-step predictions can be employed.

We first introduce the necessary settings. The cost function is defined as $C(s_t, a_t, s_{t+1}) = \mathbb{I}\{\nu(h(s_{t+1}), E_{t+1}) \leq 0\}$, where $h : S \to \mathbb{R}^{n_1}$ extracts agent-centric safety features, $E_{t+1} \in \mathbb{R}^{n_2}$ captures environment features, and $\nu : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}$ is Lipschitz continuous. By the Lipschitz property, the equation $\|h(s_{t+1}) - h(s_t)\| \leq \Delta_{\max}$ implies:

$$\nu(h(s_{t+1}), E_{t+1}) \geq \nu(h(s_t), E_t) - L_\nu \Delta_{\max}. \tag{8}$$

Thus, if $\nu(h(s_t), E_t) > L_\nu \Delta_{\max}$, then $C(s_t, a_t, s_{t+1}) = 0$ for all $a_t \in A$. Since the value $\nu\left(h\left(s_t\right), E_t\right)$ can be computed at state s_t before selecting action a_t to assess its safety, we call it as the pre-safety indicator.

1. **Pre-Safety Check**: To minimize intervention, we evaluate the pre-safety indicator:

$$\nu(h(s_t), E_t) > L_\nu \Delta, \tag{9}$$

where Δ is a predefined value larger than $\Delta_{\max}$. If this condition is violated, full safety verification is triggered; otherwise, the policy executes directly. This pre-safety check step improves computational efficiency when full safety verification is excessive.

2. **Action Generation**: The policy π generates N candidate actions $\{a_t^{(i)}\}_{i=1}^{N}$ by sampling from its action distribution $\pi(\cdot \mid s_t, b)$, where b derived by a subset of transition samples up to time step t explained in Section 4.2.

3. **Transition Prediction**: For each candidate action $a_t^{(i)}$, a function encoder $\hat{f}_{\text{FE}}$ predicts the next state: $\hat{s}_{t+1}^{(i)} = \hat{f}_{\text{FE}}(s_t, a_t^{(i)})$. Note that any pre-trained forward dynamics model $\hat{f}$ can be used for prediction. However, the function encoder enables inference of varying underlying dynamics and next-state prediction at once.

4. **Safety Verification**: Using ACP, we compute uncertainty-aware safety margins for each action:

$$\text{SafetyScore}(a_t^{(i)}) = \nu\left(h_{t+1}(\hat{s}_{t+1}^{(i)}), \hat{E}_{t+1}\right) - 2L_\nu \Gamma_t, \tag{10}$$

   where $\hat{E}_{t+1}$ represents predicted environment features and Γ_t is the adaptive conformal prediction bound for $h_{t+1}(\hat{s}_{t+1}^{(i)})$ and $\hat{E}_{t+1}$ calibrated to maintain a $1-\delta$ safety probability. Actions are ranked by their safety scores, with positive scores indicating safety compliance.

5. **Action Selection**: Define the safe action set at state s_t as $\hat{A}_{\text{safe}}(s_t) = \{a_t^{(i)} : \text{SafetyScore}(a_t^{(i)}) > 0\}$ and sampled action set as $\hat{A}_{\text{sample}} = \{a_t^{(i)}\}_{i \in [N]}$. The shield executes the following selection rule:

$$a_t^* = \begin{cases} a \sim \mathcal{U}(\text{Top}_k(\hat{A}_{\text{safe}}(s_t))), & \text{if } \hat{A}_{\text{safe}} \neq \emptyset, \\ \arg\max_{a \in \hat{A}_{\text{sample}}} \text{SafetyScore}(a), & \text{otherwise}, \end{cases} \tag{11}$$

   where $\mathcal{U}(\text{Top}_k(\cdot))$ denotes a uniform distribution over the top k actions ranked by their safety scores.

The following theorem demonstrates that an optimal policy, augmented with an adaptive shield, maximizes the expected cumulative discounted return while maintaining a tight bound on the average cost rate. See Appendix B for details.

Theorem 1. Given a Constrained Hidden Parameter MDP $\mathcal{M} = \{S, A, \Theta, T, R, C, \gamma, P_\Theta\}$ with initial state $s_0 \in S$ and failure probability $\delta \in (0,1)$, an optimal policy $\pi^* : S \times \Phi \to A$ augmented with an adaptive shield maximizes the expected cumulative discounted return $J_R(\pi^*)$ while satisfying the average cost rate constraint: for $\theta \sim P_\Theta$ and some $0 \leq \bar{\epsilon} \leq 1$,

$$\phi^{\pi^*}(s,\theta) = \lim_{H \to \infty} \frac{1}{H} \mathbb{E}_{\pi^*, T_\theta}\left[\sum_{t=0}^{H-1} C(s_t, a_t, s_{t+1}) \mid s_0 = s, \theta\right] \leq \delta + \bar{\epsilon}(1-\delta). \tag{12}$$

Given this bound, if safe actions exist at each step, this theorem proves that our algorithm achieves a low average cost rate constraint, governed by the ACP failure probability, i.e., $\phi^{\pi^*}(s,\theta) \leq \delta$.

Proof Sketch. The ACP provides a probabilistic guarantee on the deviation between the predicted state $\hat{s}_{t+1} = \hat{f}(s_t, a_t)$ and the true state s_{t+1}: $(\|s_{t+1} - \hat{s}_{t+1}\| \leq \Gamma_t) \geq 1 - \delta$, where Γ_t is the confidence region at time $t+1$. Since ν is Lipschitz continuous with constant L_ν, we bound the difference in the safety margin between the true and predicted states: $\nu(h(s_{t+1}), E_{t+1}) \geq \nu(h(\hat{s}_{t+1}), \hat{E}_{t+1}) - L_\nu \|h(s_{t+1}) - h(\hat{s}_{t+1})\| - L_\nu \left\|E_{t+1} - \hat{E}_{t+1}\right\|$. We assume that prediction errors are bounded by accurate next-state predictions: $\|h(s_{t+1}) - h(\hat{s}_{t+1})\| \leq \Gamma_t$ and $\left\|E_{t+1} - \hat{E}_{t+1}\right\| \leq \Gamma_t$. The set of safe actions is defined based on the predicted margin: $\hat{A}_{\text{safe}}(s_t) = \{a \in A \mid \nu(h(\hat{f}(s_t, a)), \hat{E}_{t+1}) > 2L_\nu\Gamma_t\}$. If an action is selected from $\hat{A}_{\text{safe}}$, we guarantee $\nu(h(s_{t+1}), E_{t+1}) > 0$, ensuring a safe state at $t+1$. The final bound depends on the failure probability of state prediction and the probability of selecting safe actions. □

5 Experiments

We empirically evaluate our approach to assess its safety, generalization, and efficiency across diverse RL tasks [1]. We compare against established safe RL baselines and analyze three variants of our method: using only the safety-regularized objective, only the adaptive shield, and their combination. Our experiments are guided by the following research questions:

[1] Our code is available on Code Link

- **RQ1:** How does our approach balance safety and task performance during training?
- **RQ2:** How well does our approach generalize to out-of-distribution test environments?
- **RQ3:** What is the runtime overhead of our approach at execution?

5.1 Experimental Setup

Environments. We conduct experiments using the Safe-Gym benchmark [Ji et al., 2023] for safe RL, with two robot types: *Point* and *Car*. Each robot performs four tasks: (1) *Goal*: navigate to a target while avoiding obstacles; (2) *Button*: activate a button while avoiding hazards; (3) *Push*: push an object to a goal under contact constraints; (4) *Circle*: follow a circular path while staying within safe boundaries. Robot-task combinations are denoted as robot-task (e.g., Point-Goal, Car-Circle). Each task includes a safety constraint (e.g., obstacle avoidance or region adherence). Episode-level randomness is introduced by sampling gravity (α_g) and hidden dynamics parameters (α_d, α_ρ).

Baselines. We compare our approach to four established safe RL algorithms:

- **TRPO-Lag**: Trust Region Policy Optimization with second-order reward updates and first-order constraint updates [Schulman et al., 2015].
- **PPO-Lag**: Proximal Policy Optimization with Lagrangian updates for both reward and constraint [Schulman et al., 2017].
- **CPO**: Constrained Policy Optimization with joint second-order updates to enforce linearized cost constraints [Achiam et al., 2017].
- **USL**: Unrolling Safety Layer, which re-weights the policy loss for safety and projects unsafe actions into a feasible set at execution [Zhang et al., 2023].

Baselines are given access to hidden parameters to support dynamics adaptation, as they do not perform inference. In contrast, our approach uses $\hat{f}_{\text{FE}}$ to infer hidden parameters online and is evaluated *without this privileged information*, demonstrating robustness under limited parameter awareness.

Hyperparameters. All methods use the default hyperparameters provided by their respective implementations: Omni-Safe [Ji et al., 2024] for TRPO-Lag, PPO-Lag, and CPO, and Safe-RL-Kit [Zhang et al., 2023] for USL. When evaluating our approach on top of each base algorithm (e.g., TRPO or PPO), we adopt the same hyperparameters as the corresponding baseline to ensure a fair comparison. Each method is trained for 2 million environment steps using 3 random seeds. Each trained policy is evaluated over 100 episodes at test time.

We set the pre-safety distance to 0.275 and the ACP failure probability to 2%. The function encoder $\hat{f}_{\text{FE}}$ is pre-trained on 1000 episodes (1000 steps each) and remains fixed during policy training, introducing realistic prediction error that is managed by ACP. All agents are trained under a strict safety constraint, with a cost limit of zero.

For in-distribution training, environment parameters (α_g, α_d, α_ρ) are sampled uniformly from $[0.25, 1.75]$. For out-of-distribution evaluation, parameters are sampled from $[0.1, 0.25] \cup [1.75, 2.5]$, and the number of obstacles is increased to stress generalization.

Metrics. We evaluate each method using per-episode averages for the following metrics, each capturing a different aspect of performance: (1) *Return*, measuring task performance as the cumulative reward per episode; (2) *Cost Rate*, reflecting safety by measuring the frequency of constraint violations per timestep; (3) *Runtime*, quantifying execution-time efficiency as the wall-clock time per episode; and (4) *Shielding Rate*, indicating how often the shield is triggered during an episode (i.e., when the pre-safety check fails).

5.2 Results Analysis

RQ1: Trade-offs Between Safety and Return. Figure 1 shows training curves of episodic return and cost rate for the Car robot across four tasks. Baseline methods exhibit a range of trade-offs. TRPO-Lag and PPO-Lag tend to achieve high returns but incur higher cost rates. CPO enforces strict safety through a zero-violation constraint, often at the expense of reward learning, resulting in ineffective policies on several tasks. USL, which relies on cost-Q-value estimation, performs well on the Circle task but struggles on Goal, Button, and Push—likely due to sensitivity to environment stochasticity, such as randomly reset obstacle positions, which disrupt cost estimation. These results

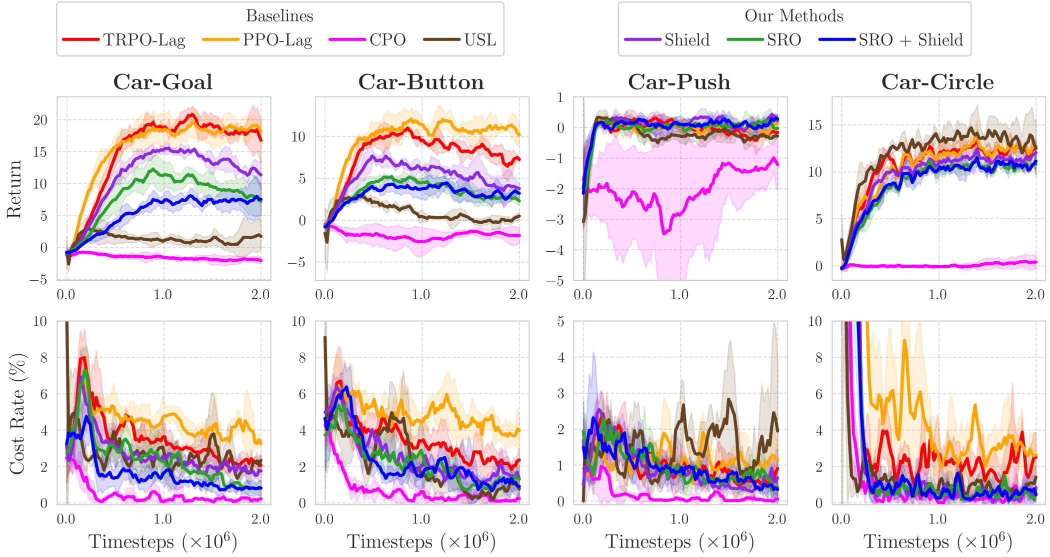

Figure 1: Training curves of episodic return and cost rate for the Car robot across four tasks. Solid lines show means over three seeds; shaded regions indicate the 95% range as ±2 standard deviations.

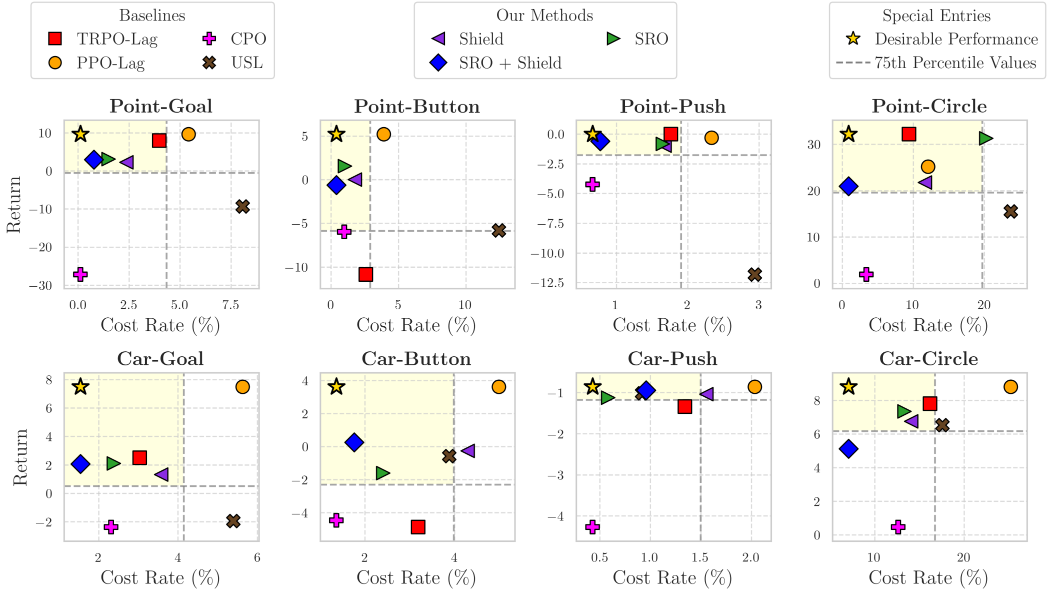

Figure 2: Trade-off between average episodic return and cost rate in out-of-distribution domains.

highlight the challenge of achieving both safety and return without mechanisms for adapting to latent dynamics and managing uncertainty.

In contrast, our methods (using TRPO as the base RL algorithm) consistently achieve lower cost rates while maintaining competitive returns, demonstrating their ability to balance safety and task performance during training. Variants using only the safety-regularized objective (SRO) or only the adaptive shield also reduce cost violations compared to baselines, but are less effective than the combined method. We observe similar results for the Point robot and when using PPO as the base RL algorithm for our methods (see results in Appendix C and D).

Takeaway: our combined method (SRO + shield) achieves the best safety outcomes during training, with only a modest impact on task return.

RQ2: Generalization to Out-of-Distribution Environments. Figure 2 illustrates the trade-off between average episodic return and cost rate in out-of-distribution test environments across all tasks.

Robots	Methods & Metrics	Goal	Button	Push	Circle
Car	TRPO-Lag Runtime (s)	2.52±0.03	2.97±0.02	2.85±0.01	1.74±0.00
	Ours Runtime (s)	2.80±0.04	3.10±0.06	3.11±0.02	1.59±0.08
	Ours Shielding Rate (%)	22.15±2.88	16.68±1.80	17.05±1.52	15.82±0.29
Point	TRPO-Lag Runtime (s)	2.26±0.00	2.62±0.01	2.58±0.03	1.63±0.02
	Ours Runtime (s)	2.57±0.15	2.70±0.00	2.69±0.06	1.71±0.07
	Ours Shielding Rate (%)	22.55±5.23	14.09±0.50	13.66±3.00	5.93±5.23

Table 1: Runtime (in seconds) and shielding rate (in percent) across tasks for each robot type. “Ours” refers to the combined method using SRO and the adaptive shield.

Each marker corresponds to the mean performance across three trained policies (one per seed) for a given method, evaluated separately on Point and Car robots.

Our full method (SRO + Shield) consistently appears near or within the desirable yellow region (high return and low cost rate) across all tasks, indicating strong generalization to previously unseen dynamics. The SRO-only and Shield-only variants also perform well but tend to deviate more from the Pareto front, especially in complex environments. This trend highlights the complementary effect of combining proactive (SRO) and reactive (Shield) safety mechanisms.

Among the baselines, CPO maintains low cost rates but sacrifices return, often placing it outside the desirable region. PPO-Lag and TRPO-Lag yield high returns but suffer from high violation rates. USL exhibits inconsistent behavior, particularly on the Point robot, suggesting poor generalization under dynamic, partially-observed conditions.

Takeaway: our approach generalizes effectively to out-of-distribution settings, consistently achieving a favorable balance between return and safety across robot types and task variations.

RQ3: Execution-Time Efficiency. Table 1 compares the average runtime per episode between the baseline method (TRPO-Lag) and our full approach (SRO + Shield). Across all tasks and both robot types, our method introduces only a modest runtime overhead, demonstrating practical efficiency during execution. Additionally, the shield trigger rate remains moderate, indicating that safety shielding is invoked selectively and does not dominate execution time.

Takeaway: our approach achieves safe execution with limited computational overhead, making it suitable for real-time deployment.

Ablation Studies. We conduct additional ablation studies, presented in the appendix. These include an analysis of key hyperparameters, such as sampling size and safety bonus (Appendix E), and an evaluation of the impact of the function encoder’s representation on safety and performance (Appendix F). Our ablations show that performance remains stable across different sampling sizes and safety bonus values. Additionally, our function encoder-based inference effectively adapts to varying dynamics T, achieving performance comparable to that of oracle representations.

6 Conclusion

We presented a novel approach for safe and generalizable reinforcement learning in settings with dynamically varying hidden parameters. Our approach comprises three key components: (1) function encoder-based inference of hidden dynamics, (2) a safety-regularized objective that promotes low-violation behavior during training, and (3) an adaptive runtime shield that uses conformal prediction to filter unsafe actions based on uncertainty at execution time. Experimental results demonstrate that our approach consistently outperforms strong baselines in reducing safety violations while maintaining competitive task performance, and generalizes effectively across diverse tasks and out-of-distribution environments.

Despite its effectiveness, our approach has several limitations. First, the safety guarantees rely on assumptions about the structure of the cost function, although these apply to a broad range of practical scenarios. Second, the method depends on an offline dataset to train the function encoder, which may limit applicability in settings without prior data. Third, our evaluation has so far been limited to simulated environments. Future work will aim to address these limitations by relaxing modeling assumptions, reducing reliance on offline data, and extending evaluations to physical robotic platforms to assess scalability and real-world applicability.

References

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

Mohammed Alshiekh, R. Bloem, Rüdiger Ehlers, Bettina Könighofer, S. Niekum, and U. Topcu. Safe reinforcement learning via shielding. In *AAAI Conference on Artificial Intelligence*, 2017.

Qinbo Bai, Amrit Singh Bedi, and Vaneet Aggarwal. Achieving zero constraint violation for constrained reinforcement learning via conservative natural policy gradient primal-dual algorithm. In *AAAI*, 2023.

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.

Michael Beukman, Devon Jarvis, Richard Klein, Steven James, and Benjamin Rosman. Dynamics generalisation in reinforcement learning via adaptive context-aware policies. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

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.

Onur Celik, Aleksandar Taranovic, and Gerhard Neumann. Acquiring diverse skills using curriculum reinforcement learning with mixture of experts. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

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.

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.

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

Ian Gibbs and Emmanuel Candès. Adaptive conformal inference under distribution shift. In *Advances in Neural Information Processing Systems*, 2021.

Ian Gibbs and Emmanuel J Candès. Conformal inference for online prediction with arbitrary distribution shift. *Journal of Machine Learning Research*, 2024.

Yingtao Hu, Jianye Hao, Junyou Li, Yujing Hu, Chongjie Zhang, Changjie Fan, and Yang Gao. Autocost: Evolving intrinsic cost for zero-violation reinforcement learning. In *AAAI*, 2023.

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

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

Tyler Ingebrand, Adam J. Thorpe, and Ufuk Topcu. Function encoders: A principled approach to transfer learning in hilbert spaces. *arXiv preprint arXiv:2501.18373*, 2025.

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 *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Jiaming Ji, Jiayi Zhou, Borong Zhang, Juntao Dai, Xuehai Pan, Ruiyang Sun, Weidong Huang, Yiran Geng, Mickel Liu, and Yaodong Yang. Omnisafe: An infrastructure for accelerating safe reinforcement learning research. *Journal of Machine Learning Research*, 2024.

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

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *J. Artif. Int. Res.*, 2023.

George Konidaris and Finale Doshi-Velez. Hidden parameter markov decision processes: An emerging paradigm for modeling families of related tasks. In *Knowledge, Skill, and Behavior Transfer in Autonomous Robots: Papers from the 2014 AAAI Fall Symposium*, 2014.

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.

Tao Liu, Ruida Zhou, Dileep Kalathil, Panganamala R. Kumar, and Chao Tian. Learning policies with zero or bounded constraint violation for constrained mdps. In *Advances in Neural Information Processing Systems*, 2021.

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.

Haitong Ma, Yuhong Liu, Jianye Hao, Zhaopeng Meng, Chongjie Zhang, and Yang Gao. Learn zero-constraint-violation safe policy in model-free constrained reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

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.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

Sahand Rezaei-Shoshtari, Charlotte Morissette, Francois R. Hogan, Gregory Dudek, and David Meger. Hypernetworks for zero-shot transfer in reinforcement learning. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, 2023.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

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

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 2008.

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

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*, 2024b.

Brijen Thananjeyan, A. Balakrishna, Suraj Nair, Michael Luo, K. Srinivasan, M. Hwang, Joseph E. Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. In *IEEE Robotics and Automation Letters*, 2020.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems*, 2019.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

Akifumi Wachi and Yanan Sui. Safe reinforcement learning in constrained markov decision processes. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Jiachen Yang, Brenden Petersen, Hongyuan Zha, and Daniel Faissol. Single episode policy transfer in reinforcement learning. In *International Conference on Learning Representations*, 2019.

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

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.

Linrui Zhang, Qin Zhang, Li Shen, Bo Yuan, Xueqian Wang, and Dacheng Tao. Evaluating model-free reinforcement learning toward safety-critical tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Weiye Zhao, Tairan He, and Changliu Liu. Model-free safe control for zero-violation reinforcement learning. In *5th Conference on Robot Learning*, 2021.

A Broader Impacts

This paper discusses potential positive societal impacts of our safe RL framework for robotics. We do not identify specific negative societal impacts from our research, as it focuses on improving safety and robustness in controlled settings. Our work enhances safety in autonomous systems, which could benefit applications like autonomous driving and robotic manipulation by reducing accidents.

B Proofs

This section presents our main theoretical results, including proofs. We first introduce the necessary notations.

Notations. We introduce the notation for dimensions and sets as follows. For $n, m, k_1, k_2 \in \mathbb{N}$, let n denote the state dimension, m the action dimension, k_1 the hidden parameter dimension, and k_2 the dimension of the function encoder's learned representation. Note that the dimension k_2 of the learned representation may differ from k_1, depending on the number of chosen basis functions. The state space is $S \subseteq \mathbb{R}^n$, the action space is $A \subseteq \mathbb{R}^m$, the hidden parameter space is $\Theta \subseteq \mathbb{R}^{k_1}$, and the learned representation space is $\Phi \subseteq \mathbb{R}^{k_2}$, where Φ is the coefficient space of basis functions for the function encoder.

- $\mathcal{M}$: Constrained Hidden Parameter Markov Decision Process.
- $s_t \in \mathbb{R}^n$: State at time step t.
- $\hat{s}_t \in \mathbb{R}^n$: Predicted State for time step t.
- $X_{i,t}$ $(X_t) \in \mathbb{R}^2$: Position of the i-th obstacle at time step t. The index i is omitted when referring to a single obstacle without ambiguity.
- $\hat{X}_{i,t}$ $(\hat{X}_t) \in \mathbb{R}^2$: Predicted position of the i-th obstacle at time step t. The index i is omitted when referring to a single obstacle without ambiguity.
- $a_t \in \mathbb{R}^m$: Action at time step t.
- $\hat{f} : S \times A \to S$: Continuous transition dynamics predictor.
- $\theta \in \mathbb{R}^{k_1}$: hidden parameter.
- $b \in \mathbb{R}^{k_2}$: learned representation to T_θ.
- $s' \sim T(\cdot \mid s, a, \theta)$: Transition dynamics given parameter θ. When s, a, s' are unspecified, we denote this by T_θ.
- $C : S \times A \times S \to [0, 1]$: Cost function bounded in $[0, 1]$.
- $\mathcal{S}_{\text{safe}}(s_t, a_t)$: Safe state set, defined as $\{s_t \in \mathcal{S} \mid C(s_t, a_t, s_{t+1}) = 0 \text{ s.t } T(s_{t+1} \mid s_t, a_t, \theta) > 0\}$ where θ is a parameter sampled per episode.

- $Q_R^\pi(s,a,\theta)$: State-action value function for reward under policy π defined as
$$\mathbb{E}_{\pi,T_\theta}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t,s_{t+1}) \mid s_0=s,a_0=a,\theta\right].$$
We aim to maximize this value.
- $Q_C^\pi(s,a,\theta)$: State-action value function for cost under policy π defined as
$$\mathbb{E}_{\pi,T_\theta}\left[\sum_{t=0}^{\infty}\gamma^t C(s_t,a_t,s_{t+1}) \mid s_0=s,a_0=a,\theta\right].$$
We aim to minimize this value.
- $Q_{\text{aug}}^\pi(s,a,\theta)$: Safety-regularized state-action value function defined as $Q_R^\pi(s,a,\theta)+\alpha Q_{\text{safe}}^\pi(s,a,\theta)$ where α is a positive constant and
$$Q_{\text{safe}}^\pi(s,a,\theta)=1-\frac{\pi(a\mid s,\theta)Q_C^\pi(s,a,\theta)}{V_C^\pi(s,\theta)+\epsilon}$$
with $\epsilon>0$ is a small constant for numerical stability. We aim to maximize this value.
- $J_{\text{safe}}(\pi)$: Safety-regularized objective function for policy π defined as
$$\mathbb{E}_{\theta\sim P_\Theta,s_0\sim\mu_0(\cdot|\theta),a_0\sim\pi(\cdot|s_0,b)}\left[Q_{\text{aug}}^\pi(s_0,a_0,\theta)\right].$$
The policy π aims to maximize this value.
- $J_R(\pi)$: Standard reward objective function for policy π defined as
$$\mathbb{E}_{\theta\sim P_\Theta,s_0\sim\mu_0(\cdot|\theta),a_0\sim\pi(\cdot|s_0,b)}\left[Q_R^\pi(s_0,a_0,\theta)\right].$$
The policy π aims to maximize this value.
- $J_C(\pi)$: Standard cost objective function for policy π defined as
$$\mathbb{E}_{\theta\sim P_\Theta,s_0\sim\mu_0(\cdot|\theta),a_0\sim\pi(\cdot|s_0,b)}\left[Q_C^\pi(s_0,a_0,\theta)\right].$$
The policy π aims to minimize this value.
- $\Pi_{\text{zero-violation}}$: Set of zero-violation policies defined as $\{\pi \mid J_C(\pi)=0\}$.
- $\delta\in(0,1)$: Failure probability.
- $\Gamma_t\in\mathbb{R}^+$: Adaptive Conformal Prediction (ACP) threshold at time step t.
- $\phi^\pi(s,\theta)$: Average cost under policy π defined as
$$\phi^{\pi^*}(s,\theta)=\lim_{H\to\infty}\frac{1}{H}\mathbb{E}_{\pi^*,T_\theta}\left[\sum_{t=0}^{H-1}C(s_t,a_t,s_{t+1})\mid s_0=s,\theta\right],$$
starting from state s and parameter θ.
- π^*: Optimal policy satisfying constraint on the average cost rate.

We restate our proposition and theorems, then provide detailed proofs.

Proposition 1. Let $\Pi_{\text{zero-violation}}$ be the set of zero-violation policies. Then, for any $\alpha\geq 0$, the optimal policy obtained by maximizing the safety-regularized objective function $J_{\text{safe}}(\pi)$ within $\Pi_{\text{zero-violation}}$ is equivalent to the optimal policy obtained by maximizing the standard reward objective $J_R(\pi)$ within the same set of policies.

Proof. By definition, if $\pi\in\Pi_{\text{zero-violation}}$, then for any state s, parameter θ, and action a with $\pi(a|s,\theta)>0$, the state-action value function for the cost is zero: $Q_C^{\pi_\theta}(s,a,\theta)=0$. This implies that for such policies, the safety term $Q_{\text{safe}}^\pi(s,a,\theta)$ in the regularized objective is a constant value of 1.

Therefore, for any policy $\pi\in\Pi_{\text{zero-violation}}$, the safety-regularized objective function becomes:
$$\begin{aligned}J_{\text{safe}}(\pi)&=\mathbb{E}_{\theta\sim P_\Theta,s_0\sim\mu_0(\cdot|\theta),a_0\sim\pi(\cdot|s_0,b)}\left[Q_R^\pi(s_0,a_0,\theta)+\alpha\cdot Q_{\text{safe}}^\pi(s_0,a_0,\theta)\right]\\&=\mathbb{E}_{\theta\sim P_\Theta,s_0\sim\mu_0(\cdot|\theta),a_0\sim\pi(\cdot|s_0,b)}\left[Q_R^\pi(s_0,a_0,\theta)+\alpha\cdot 1\right]\\&=J_R(\pi)+\alpha\end{aligned}$$

Since α is a constant, maximizing $J_{\text{safe}}(\pi)$ within $\Pi_{\text{zero-violation}}$ is equivalent to maximizing $J_R(\pi)$ within the same set. □

Next, we will prove our main theorem. To prove the main theorem, we first establish the necessary notation and settings.

We define the safe state set for a given state $s \in S$, action $a \in A$, and hidden parameter $\theta \in \Theta$ as: $\mathcal{S}_{\text{safe}}(s,a) = \{s' \in S \mid C(s,a,s') = 0 \text{ s.t. } T(s' \mid s,a,\theta) > 0\}$. Thus, $\mathcal{S}_{\text{safe}}(s,a)$ contains next states s' where the safety condition is satisfied. Throughout the theorem, we address a cost function defined by safe distance

$$C(s_t, a_t, s_{t+1}) = \begin{cases} 1 & \text{if } \min_{X_{t+1}} \|pos(s_{t+1}) - X_{t+1}\| \le d \\ 0 & \text{otherwise,} \end{cases}$$

where X_{t+1} denotes the positions of obstacles at time step $t+1$, and $\text{pos}(s_{t+1})$ represents the agent's position in state s_{t+1}. As our theorem applies to any norm satisfying the triangle inequality, we do not specify a particular norm.

We assume the state includes the agent's position, a natural choice for navigation tasks. Generally, the state contains critical information needed to evaluate the cost function. Hence, a function $\text{pos} : S \to \mathbb{R}^2$ is a projection mapping, which is 1-Lipschitz continuous, meaning that

$$\|\text{pos}(s) - \text{pos}(s')\| \le \|s - s'\|$$

for all $s, s' \in S$.

We assume that the obstacle position X_t can be derived from the state s_t. This assumption is reasonable, as the agent's state typically includes safety-critical information. For instance, robots in navigation tasks use sensors to detect nearby obstacles, with this information integrated into the agent's state. This setup applies to all navigation environments in Safety Gymnasium. Formally, we assume a 1-Lipschitz continuous function $\text{sensor} : S \to \mathbb{R}^{2M}$, defined as $\text{sensor}(s_t) = [X_{1,t}, X_{2,t}, \dots, X_{M,t}]$, where $X_{i,t} \in \mathbb{R}^2$ is the position of the i-th obstacle detected by the robot, and M is the number of detected obstacles. When more than M obstacles are present, the sensor typically detects the M closest ones.

We adopt a Gaussian policy π, commonly employed in training RL policies across various algorithms.

Remark 1. To ensure the validity of conformal prediction, we note that the parameter θ, sampled at the start of each episode, remains fixed in each episode. This ensures that the dataset within each episode is exchangeable with respect to the transition dynamics T_θ. For the calibration set, we collect online samples to maintain exchangeability. Specifically, during the first 100 steps, we gather samples for calibration without using the ACP region. After 100 steps, we employ the online-collected calibration set to determine the ACP region.

Our argument extends to any Lipschitz continuous cost function bounded in $[0,1]$, with the proof following a similar approach. If the cost function is bounded by a constant $D > 1$, the proof remains valid, but the final bound is scaled by D.

Lemma 1. *Let $\hat{f}$ be a transition dynamics predictor and $h(a) = \min_{\hat{X}_{t+1}} \|\text{pos}(\hat{f}(s_t,a)) - \hat{X}_{t+1}\|$. Under the adaptive shielding mechanism with sampling size N for each episode with parameter θ, one of the following conditions holds:*

1. $\mathbb{P}(s_{t+1} \in S_{\text{safe}}(s_t,a_t)) \ge 1 - \delta$, *where* $s_{t+1} \sim T(\cdot \mid s_t, a_t, \theta)$,
2. $\min_{X_{t+1}} \|\text{pos}(s_{t+1}) - X_{t+1}\| \ge \max_{a \in A} h(a) - \epsilon_N - 2\Gamma_t$,

where $\lim_{N\to\infty} \epsilon_N = 0$ and Γ_t is the ACP confidence region for the state prediction at time step $t+1$.

Proof. Note that s_{t+1} is safe if $\min_{X_{t+1}} \|\text{pos}\,(s_{t+1}) - X_{t+1}\| > d$. The ACP gives us a probabilistic bound on the deviation between the true next state s_{t+1} and the predicted state $\hat{s}_{t+1} = \hat{f}(s_t, a_t)$:

$$\mathbb{P}\left(\|\hat{s}_{t+1} - s_{t+1}\| \le \Gamma_t\right) \ge 1 - \delta$$

We connect the safety of s_{t+1} to the position of the predicted state $\hat{s}_{t+1}$, using this bound. By triangle inequality, we have

$$\min_{X_{t+1}} \|\text{pos}\,(s_{t+1}) - X_{t+1}\| \ge \min_{X_{t+1}} \|\,\text{pos}\,(\hat{s}_{t+1}) - \hat{X}_{t+1}\| - \|\hat{X}_{t+1} - X_{t+1}\| - \|\text{pos}\,(s_{t+1}) - \text{pos}\,(\hat{s}_{t+1})\| . \tag{13}$$

Since pos function and sensor function are 1-Lipschitz, we have

$$\|\text{pos}\left(s_{t+1}\right)-\text{pos}\left(\hat{s}_{t+1}\right)\| \leq \|s_{t+1}-\hat{s}_{t+1}\| \text{ and } \|X_{t+1}-\hat{X}_{t+1}\| \leq \|s_{t+1}-\hat{s}_{t+1}\|.$$

Hence, if $\|s_{t+1}-\hat{s}_{t+1}\| \leq \Gamma_t$ (which occurs with probability at least $1-\delta$), then

$$\|\text{pos}\left(s_{t+1}\right)-\text{pos}\left(\hat{s}_{t+1}\right)\| \leq \Gamma_t \text{ and } \|X_{t+1}-\hat{X}_{t+1}\| \leq \Gamma_t.$$

This implies

$$\min_{X_{t+1}} \|\text{pos}\left(s_{t+1}\right)-X_{t+1}\| \geq \min_{X_{t+1}} \|\text{pos}\left(\hat{s}_{t+1}\right)-X_{t+1}\|-2\Gamma_t. \tag{14}$$

Thus, if

$$h(a_t)=\min_{X_{t+1}} \left\|\text{pos}\left(\hat{s}_{t+1}\right)-\hat{X}_{t+1}\right\| > d+2\Gamma_t,$$

then $\min_{X_{t+1}} \|\text{pos}\left(s_{t+1}\right)-X_{t+1}\| > d$ whenever $\|s_{t+1}-\hat{s}_{t+1}\| \leq \Gamma_t$. Let us define the set of safe actions on the predicted state by

$$\hat{A}_{\text{safe}}\left(s_t\right)=\left\{a \in A \mid h(a) > d+2\Gamma_t\right\}.$$

We now consider two cases based on the feasibility of selecting an action from the set $\hat{A}_{\text{safe}}$.

Case 1: If we can select $a_t \in \hat{A}_{\text{safe}}\left(s_t\right)$ and $\|s_{t+1}-\hat{s}_{t+1}\| \leq \Gamma_t$, then s_{t+1} is safe by Equation 14. By ACP of our adaptive shielding mechanism, we guarantee

$$\mathbb{P}\left(\|\hat{s}_{t+1}-s_{t+1}\| \leq \Gamma_t\right) \geq 1-\delta$$

where δ is a failure probability of ACP. Thus, condition 1 holds.

Case 2: If we cannot select $a_t \in \hat{A}_{\text{safe}}(s_t)$, our adaptive shielding mechanism samples N actions $\{a_t^{(i)}\}$ and picks the action a_t such $h(a_t)=\max_{a \in \{a_t^{(i)}\}} h(a)$. Note that $h(a)$ is continuous on a and a Gaussian policy π assigns positive probability to any subset of action space A. Hence, as sample size N goes to ∞, $\max_{a \in A} h(a)-h(a_t)=\epsilon_N$ goes to 0. Also, by Equation 14, we have

$$\|\text{pos}(s_{t+1})-X_{t+1}\| \geq h(a_t)-2\Gamma_t=\max_{a \in A} h(a)-\epsilon_N-2\Gamma_t.$$

Thus, condition 2 holds. □

To prove the theorem, we recall the function $h(a)=\min_{\hat{X}_{t+1}} \|\text{pos}(\hat{f}(s_t,a))-\hat{X}_{t+1}\|$, representing the minimum distance between the predicted state and predicted obstacles. Using this, we define the safe action set for the predicted state as:

$$\hat{A}_{\text{safe}}\left(s_t\right)=\left\{a \in A \mid h(a) > d+2\Gamma_t\right\}.$$

Lemma 1 considers two cases based on whether sampling from $\hat{A}_{\text{safe}}(s_t)$ is feasible. To derive a bound for the average cost rate constraint, we analyze both cases by defining $\epsilon_t=\mathbb{P}(\hat{A}_{\text{safe}}(s_t)=\emptyset)$. We assume that if $\hat{A}_{\text{safe}}(s_t)$ is non-empty, a large sample size N allows sampling an action from this set, as discussed in Lemma 1.

Theorem 1. Given a Constrained Hidden Parameter MDP $\mathcal{M}=\{S,A,\Theta,T,R,C,\gamma,P_\Theta\}$ with initial state $s_0 \in S$, and failure probability $\delta \in (0,1)$, an optimal policy $\pi^*: S \times \Phi \rightarrow A$, augmented with an adaptive shield, maximizes the expected cumulative discounted return $J_R(\pi^*)$, while satisfying the average cost rate constraint:

$$\phi^{\pi^*}(s,\theta)=\lim_{H \rightarrow \infty} \frac{1}{H} \mathbb{E}_{\pi^*,T_\theta}\left[\sum_{t=0}^{H-1} C(s_t,a_t,s_{t+1}) \mid s_0=s,\theta\right] \leq \delta+\bar{\epsilon}(1-\delta), \tag{15}$$

for some $0 \leq \bar{\epsilon} \leq 1$ and $\theta \sim P_\Theta$.

Proof. At each time step t, $\hat{A}_{\text{safe}}(s_t)$ is non-empty with probability $1-\epsilon_t$, allowing us to sample actions with a large sample size N. By Lemma 1, this guarantees:

$$\mathbb{P}\left(s_{t+1} \in S_{\text{safe}}\left(s_t,a_t\right)\right) \geq 1-\delta$$

where $s_{t+1} \sim T(\cdot \mid s_t, a_t, \theta)$, and the safe state set is defined as:

$$S_{\text{safe}}\left(s_t, a_t\right) = \left\{s' \in S \mid C\left(s_t, a_t, s'\right) = 0, T\left(s' \mid s_t, a_t, \theta\right) > 0\right\}$$

Thus, when $\hat{A}_{\text{safe}}(s_t)$ is non-empty with probability $1 - \epsilon_t$, the cost function satisfies:

$$C\left(s_t, a_t, s_{t+1}\right) = \begin{cases} 1 & \text{with probability at most } \delta \\ 0 & \text{with probability at least } 1 - \delta \end{cases},$$

which implies $\mathbb{P}(C = 1) \leq \delta$ and $\mathbb{P}(C = 0) \geq 1 - \delta$. In this case, the expected cost per step is bounded as follows:

$$\mathbb{E}_{\pi^*}\left[C\left(s_t, a_t, s_{t+1}\right)\right] \leq \delta(1 - \epsilon_t).$$

When $\hat{A}_{\text{safe}}(s_t)$ is empty with probability ϵ_t, the expected cost per step is bounded as follows:

$$\mathbb{E}_{\pi^*}\left[C\left(s_t, a_t, s_{t+1}\right)\right] \leq \epsilon_t.$$

Combining both cases, the expected cost per step is bounded by:

$$\mathbb{E}_{\pi^*}\left[C\left(s_t, a_t, s_{t+1}\right)\right] \leq \delta(1 - \epsilon_t) + \epsilon_t = \delta + \epsilon_t(1 - \delta).$$

By the linearity of expectation, this per-step bound extends to the long-term average cost for a fixed parameter θ:

$$\begin{aligned} \phi^{\pi^*}\left(s_0, \theta\right) &= \lim_{H \to \infty} \frac{1}{H} \sum_{t=0}^{H-1} \mathbb{E}_{\pi^*, T_\theta}\left[C\left(s_t, a_t, s_{t+1}\right)\right] && (16) \\ &\leq \limsup_{H \to \infty} \frac{1}{H} \sum_{t=0}^{H-1} \left(\delta + \epsilon_t(1 - \delta)\right) && (17) \\ &= \delta + \bar{\epsilon}(1 - \delta) && (18) \end{aligned}$$

where $\bar{\epsilon} = \limsup_{H \to \infty} \frac{1}{H} \sum_{t=0}^{H-1} \mathbb{E}\left[\epsilon_t\right]$. This satisfies Equation 15, completing the proof. Moreover, if safe actions exist at each time step t, i.e., $\epsilon_t = \mathbb{P}(\hat{A}_{\text{safe}} = \emptyset) = 0$, $\bar{\epsilon}$ becomes 0. Hence, we can bound the equation with a small failure probability δ. □

C Additional Results for RQ1

In the main text, we presented results for the Car robot. Here, we provide results for the Point robot. Consistent with our earlier observations, Figure 3 shows that our method reduces safety violations with only a small reduction in reward, demonstrating robustness across different robotic platforms.

D Adaptive Shielding with PPO-Lag

As our method is compatible with any RL policy π, we examine its impact when applied to a PPO-Lagrangian-based policy. Figures 4 and 5 demonstrate that integrating our method with the underlying policy consistently reduces safety violations.

E Ablation Study on Safety Bonus and Sampling Size

We evaluate the hyperparameter sensitivity of our method, focusing on the safety bonus and sampling size.

To assess the safety bonus, we fix the sampling size at 10 and vary the safety bonus in $0.05, 0.1, 0.5, 1$. Figures 6 and 7 demonstrate that a higher safety bonus reduces cost violations, consistent with the safety-regularized objective, as it increasingly incentivizes policies to minimize safety violations.

To investigate the impact of sampling size during training, we fix the safety bonus at 1 and vary the sampling size in $5, 10, 50, 100$. We observe that sampling size influences the early training phase, where an untrained policy benefits from larger samples to explore diverse actions. Figures 8 and 9

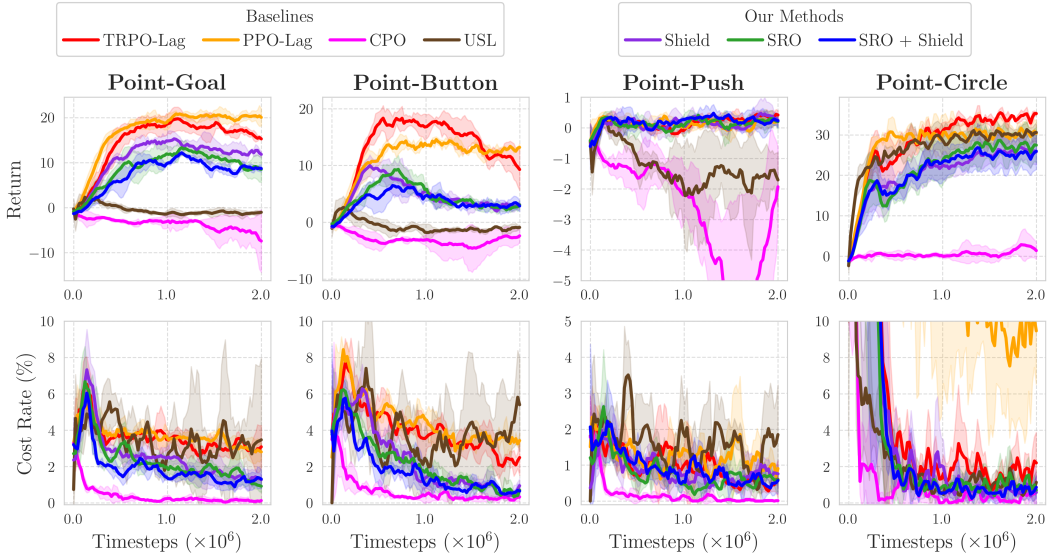

Figure 3: Training Curves for Safe Gymnasium (Point Robot, TRPO-Lag Base). **(Top)** Average episodic return versus training steps. **(Bottom)** Average episodic cost rate versus training steps.

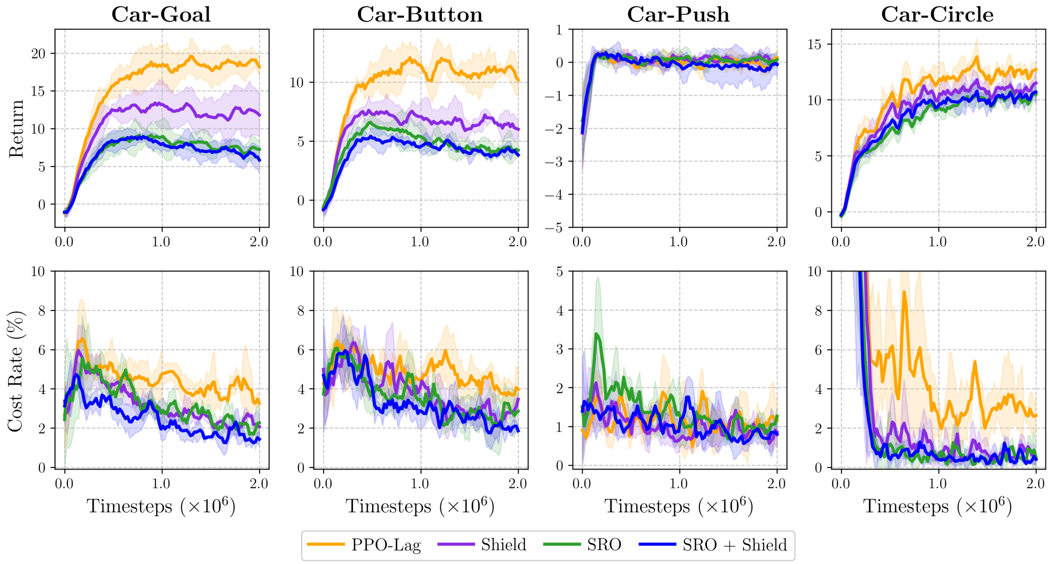

Figure 4: Training Curves for Safe Gymnasium (Car Robot, PPO-Lag Base). **(Top)** Average episodic return versus training steps. **(Bottom)** Average episodic cost rate versus training steps.

demonstrate that, as training advances, policies with varying sampling sizes converge to comparable performance in both reward and cost. These results support the robustness of our method.

We evaluate the impact of sampling size during testing, varying it across $5, 10, 50, 100$. Figures 10 and 11 show that sampling size affects performance. For instance, the Car robot exhibits performance variation with sampling size, whereas the Point robot is relatively robust to changes in sampling size. However, the figures demonstrate that our method, with different sampling sizes, remains on the Pareto frontier.

F Ablation on Representation

We investigate the function encoder's representation of varying underlying dynamics T_θ. We evaluate two representations for handling hidden parameters in our safe RL framework: the Oracle representa-

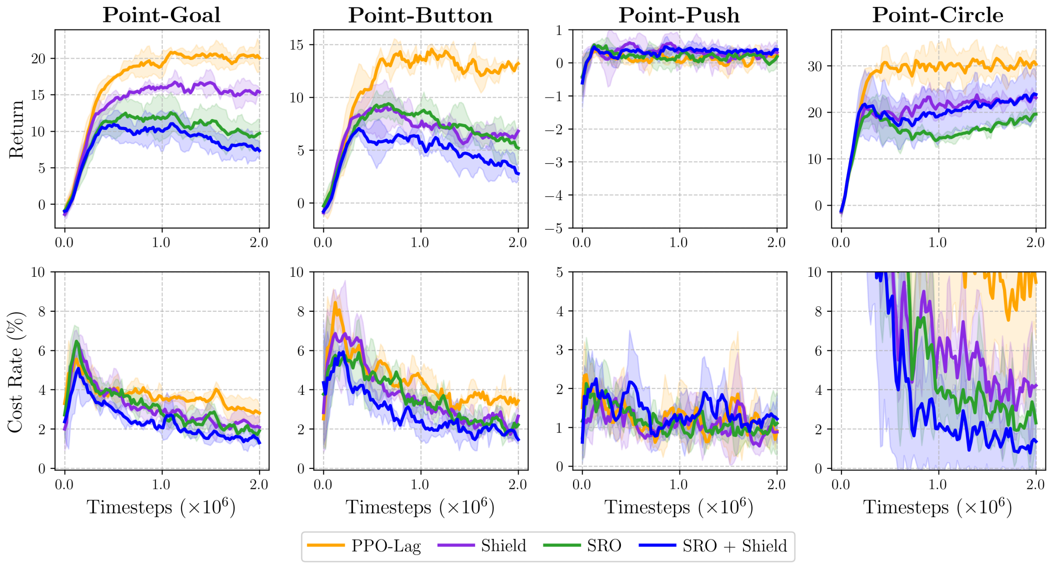

Figure 5: Training Curves for Safe Gymnasium (Point Robot, PPO-Lag Base). **(Top)** Average episodic return versus training steps. **(Bottom)** Average episodic cost rate versus training steps.

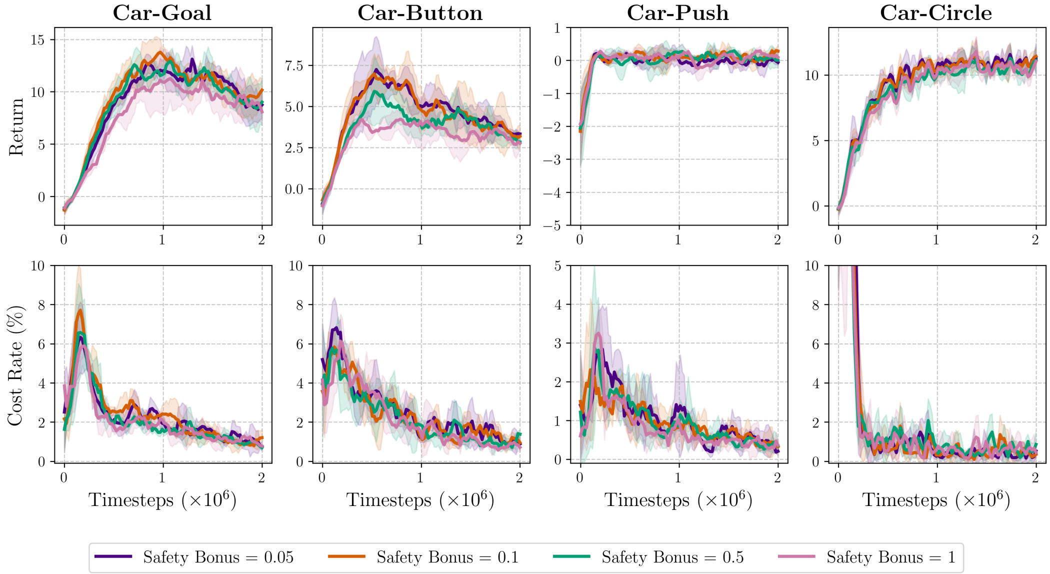

Figure 6: Ablation study on safety bonus (Car Robot, TRPO-Lag Base): **(Top)** Average episodic return over training steps, with curves for safety bonus values $\{0.05, 0.1, 0.5, 1\}$. **(Bottom)** Violation rate over training steps, decreasing with higher safety bonuses (0.5, 1). Sampling number fixed at 10.

tion, where the hidden parameter θ (a scaling factor for environmental dynamics such as density and damping) is directly provided to the policy by concatenating it with the state input, and the Function Encoder (FE) representation, which uses a function encoder $\hat{f}_{\text{FE}}$ to infer the underlying dynamics T_θ, with coefficients of pretrained basis functions serving as the representation. These representations are tested to assess the function encoder's ability to adapt to varying dynamics in Safety Gymnasium tasks.

Figures 12 and 13 show that the function encoder's representation is often comparable to the oracle representation and, in some cases, outperforms it. For instance, in the Car-Goal task, the function encoder's representation surpasses the oracle representation.

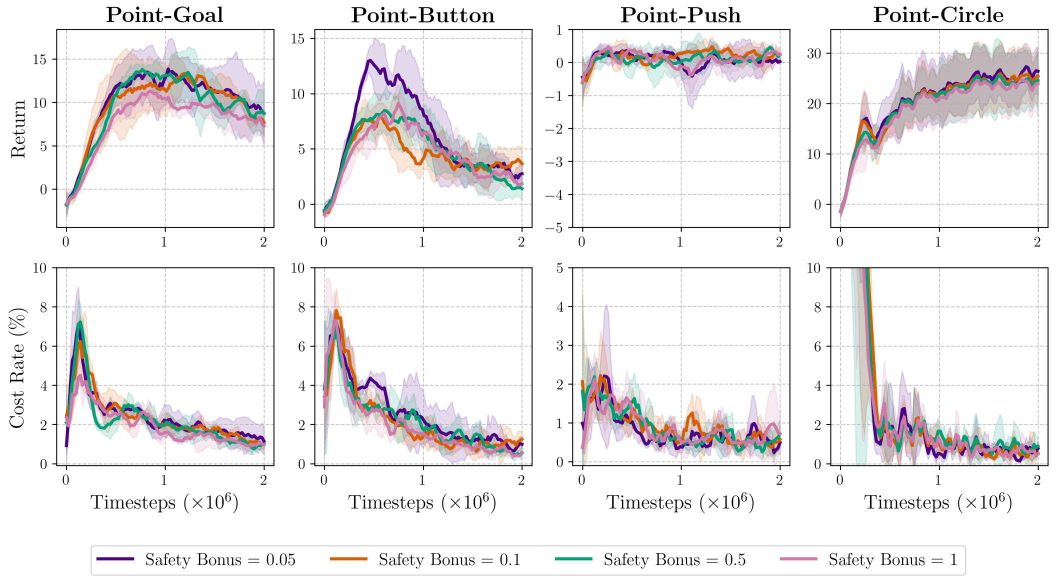

Figure 7: Ablation study on safety bonus (Point Robot, TRPO-Lag Base): **(Top)** Average episodic return over training steps, with curves for safety bonus values $\{0.05, 0.1, 0.5, 1\}$. **(Bottom)** Violation rate over training steps, decreasing with higher safety bonuses (0.5, 1). Sampling number fixed at 10.

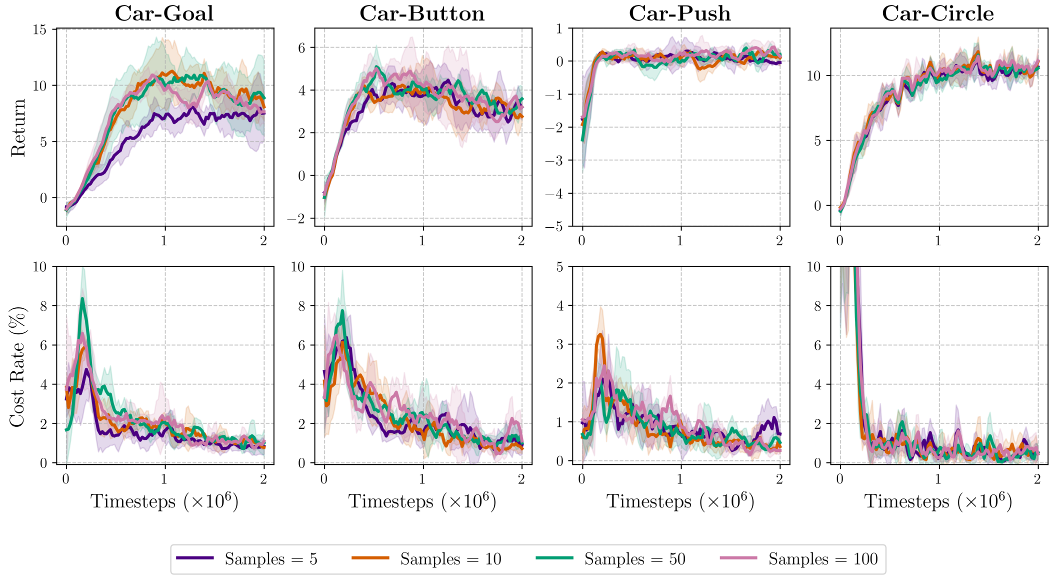

Figure 8: Ablation study on sampling size (Car Robot, TRPO-Lag Base): **(Top)** Average episodic return over training steps, with curves for sampling numbers $\{5, 10, 50, 100\}$. **(Bottom)** Violation rate over training steps, stable at low levels across sampling numbers. Safety bonus fixed at 1.

The function encoder leverages neural basis functions to represent the space of varying dynamics $\{T_\theta\}_{\theta\in\Theta}$. For instance, just as the $\mathbb{R}^2$ plane is spanned by linear combinations of basis vectors $(0, 1)$ and $(1, 0)$, the dynamics space is captured by neural basis functions, making their coefficients highly informative. This representation often transitions smoothly, as shown in [Ingebrand et al., 2024b], promoting policy effective adaptation to dynamic changes. Consequently, our function encoder's representation frequently matches or surpasses oracle representation performance.

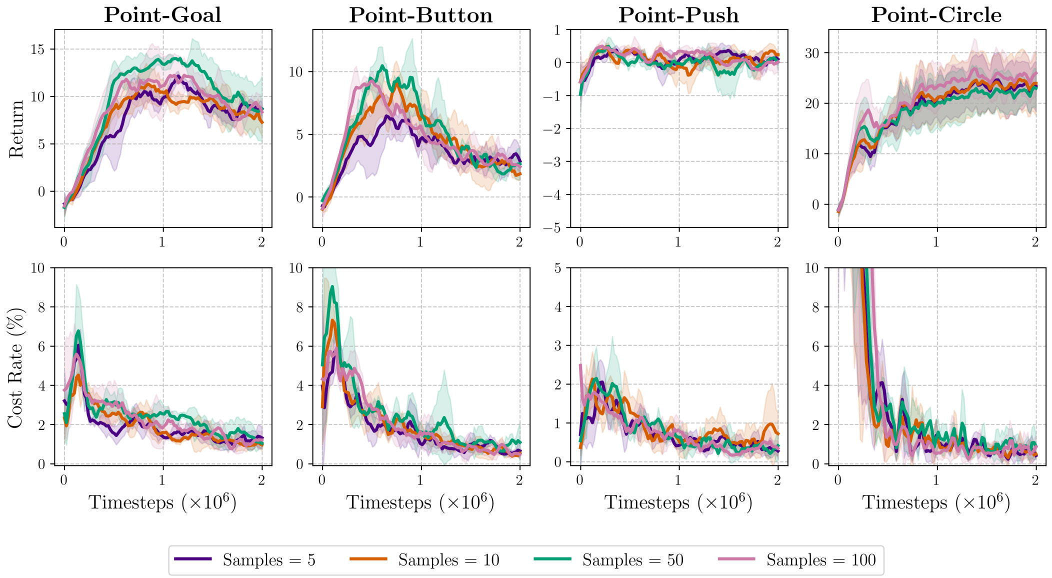

Figure 9: Ablation study on sampling size (Point Robot, TRPO-Lag Base): **(Top)** Average episodic return over training steps, with curves for sampling numbers $\{5, 10, 50, 100\}$. **(Bottom)** Violation rate over training steps, stable at low levels across sampling numbers. Safety bonus fixed at 1.

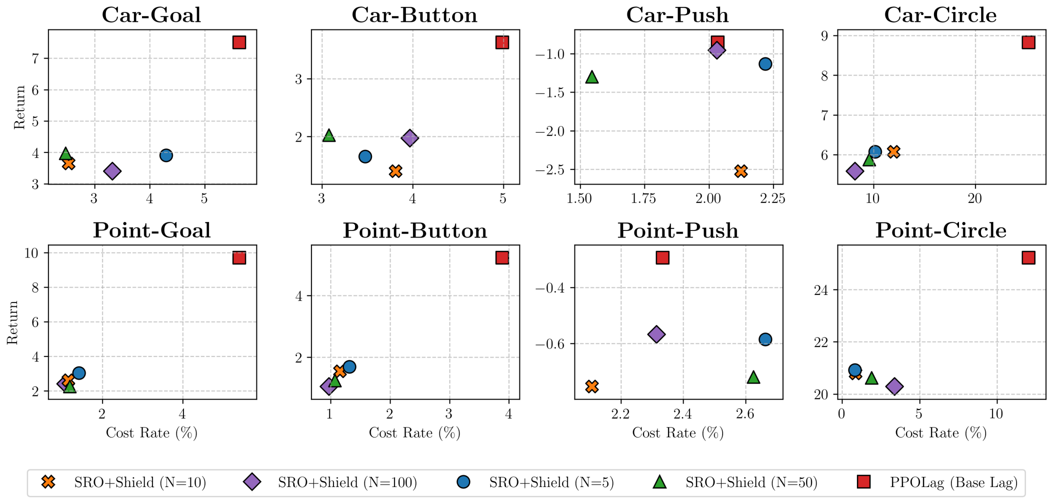

Figure 10: Ablation study on sampling size in OOD environments (PPO-Lag Base): x-axis denotes cost, y-axis denotes reward. **(Top)** Car Robot. **(Bottom)** Point Robot.

G Why Not Q_C, But Q_{safe}?

To effectively guide the policy towards safe behavior, we propose a safety-regularized objective Q_{safe}. A natural alternative is to augment the reward value function Q_R with the cost value function Q_C, which estimates the expected cost of violating safety constraints, forming $Q_{\text{aug}} = Q_R - \alpha Q_C$. However, this formulation can be transformed into Lagrangian-based safe RL methods, optimizing policies with $Q_R - \lambda Q_C$, where λ is a Lagrangian multiplier dynamically adjusted during training. In particular, the Lagrangian multiplier λ is updated using a learning rate lr. A higher learning rate accelerates the increase of λ, assigning stronger penalties on the policy for cost violations. λ is adjusted by $Q_C \times lr$; larger Q_C or learning rate values lead to faster λ growth, which increases the penalty term in the optimization objective ($Q_R - \lambda Q_C$).

However, Lagrangian-based methods are sensitive to the choice of lr, often leading to unstable optimization or suboptimal safety-performance trade-offs as shown in Figures 14 and 15. This is

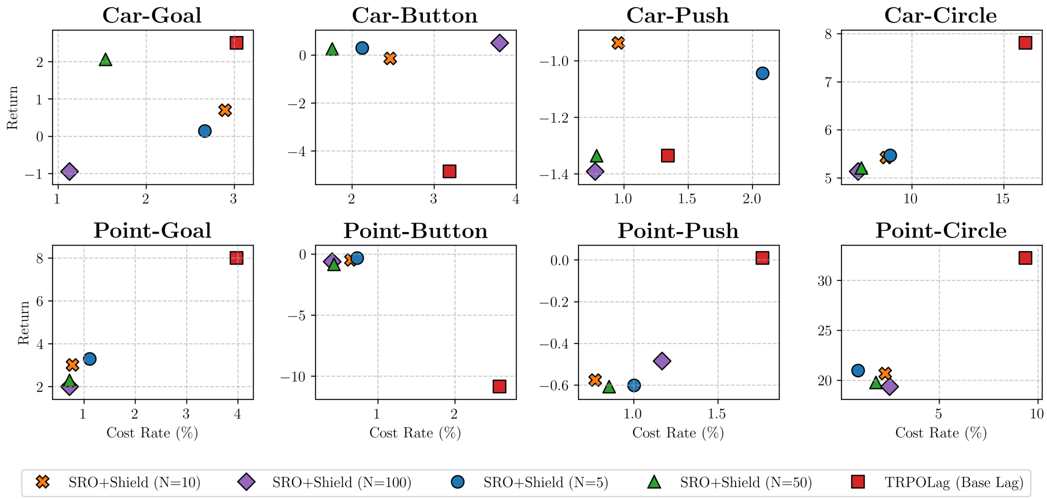

Figure 11: Ablation study on sampling size in OOD environments (TRPO-Lag Base): x-axis denotes cost, y-axis denotes reward. **(Top)** Car Robot. **(Bottom)** Point Robot.

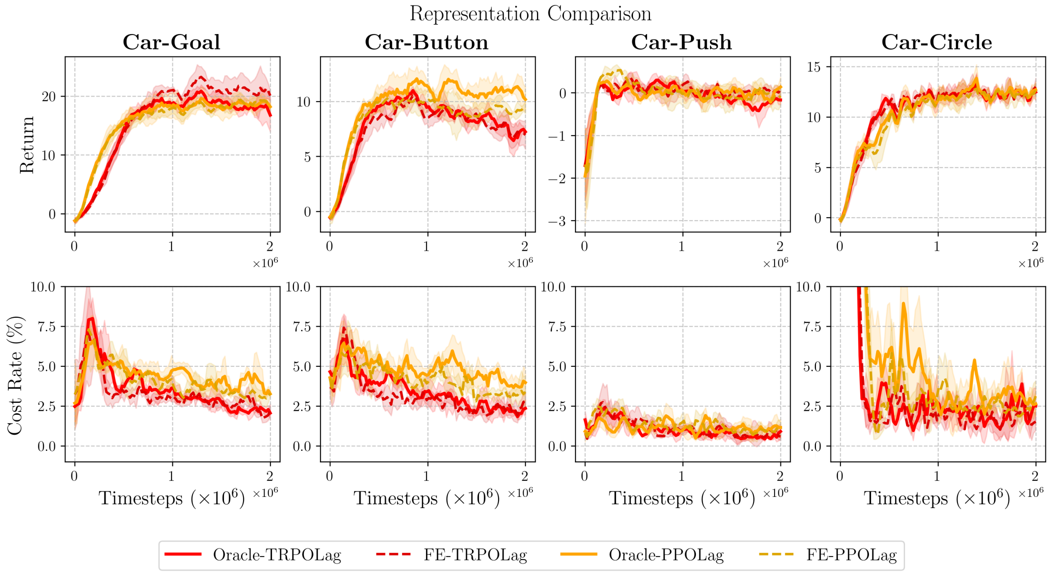

Figure 12: Ablation study on Representation (Car robot). "Oracle-" refers to a policy directly informed of hidden parameters, while "FE-" denotes the function encoder's representation derived from observations.

because the value of Q_C is highly environment-dependent, varying with the magnitude of costs and the dynamics induced by hidden parameters. In contrast, our safety-regularized objective Q_{safe} incorporates a normalized term, constrained to $(0, 1]$. This normalization simplifies controlling the safety bonus by ensuring it remains bounded. Figures 6 and 7 demonstrate that Q_{safe} maintains consistent performance regardless of Q_C's scale, achieving reliable policy optimization in hidden-parameter settings.

H Cost Functions

In this section, we present two cost functions used in our experiments. Each cost function conforms to the form:

$$C(s_t, a_t, s_{t+1}) = \mathbb{I}\left\{\nu(h(s_{t+1}), E_{t+1}) \leq 0\right\},$$

as defined in Section 4.3, where:

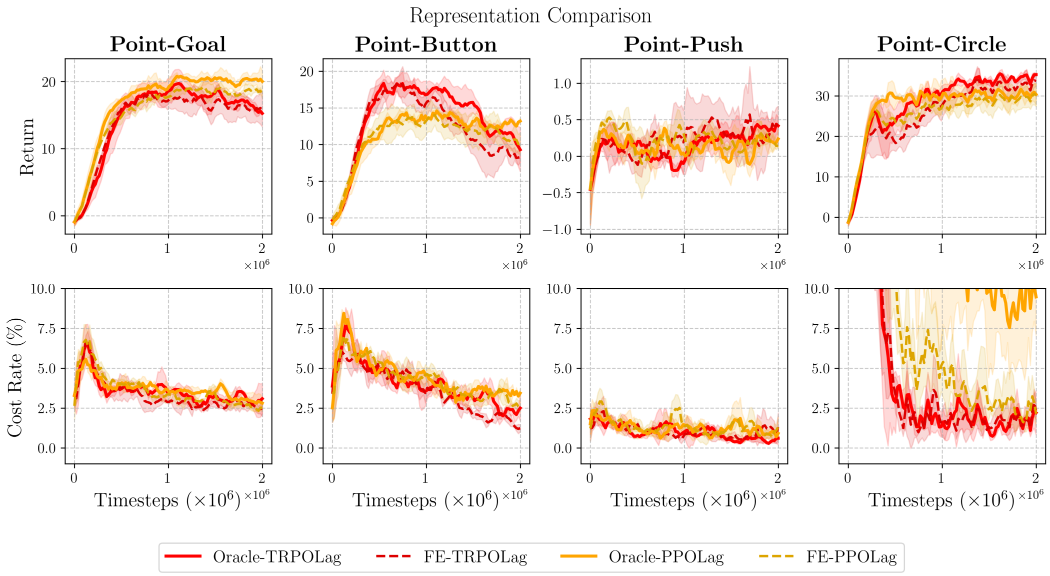

Figure 13: Ablation study on Representation (Point robot). "Oracle-" refers to a policy directly informed of hidden parameters, while "FE-" denotes the function encoder's representation derived from observations.

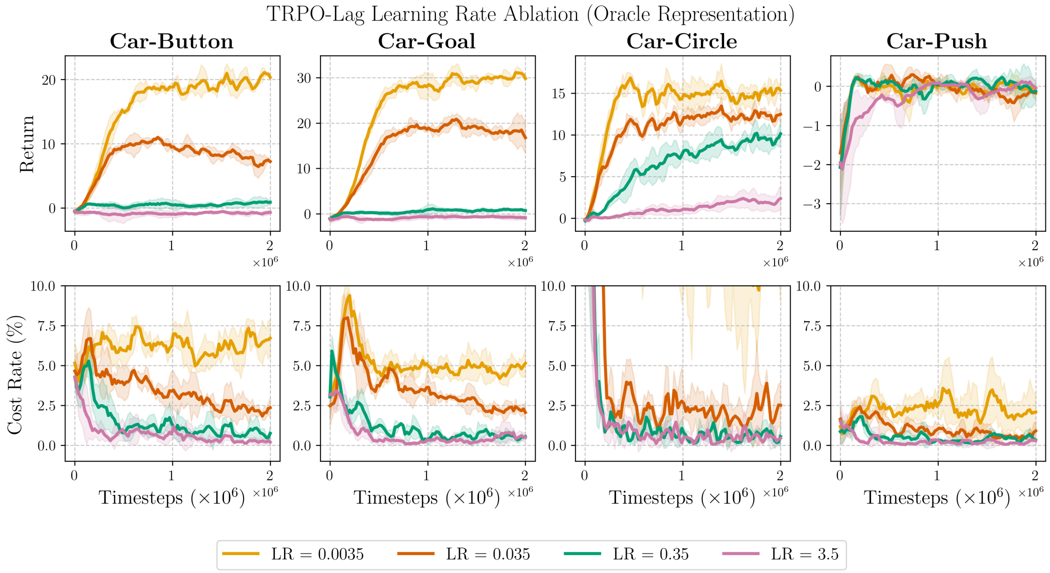

Figure 14: Training Curves for Safe Gymnasium with Car Robot. **(Top)** Average episodic return versus training steps. **(Bottom)** Average episodic cost rate versus training steps.

- $h : S \to \mathbb{R}^{n_1}$ extracts agent-centered safety features from the next state s_{t+1},
- $E_{t+1} \in \mathbb{R}^{n_2}$ captures environment features (e.g., obstacle positions, safe region boundaries),
- $\nu : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}$ is a Lipschitz continuous function, with $\nu > 0$ indicating safety and $\nu \leq 0$ indicating a violation.

Collision Avoidance. Given the robot's position $\mathrm{pos}(s) \in \mathbb{R}^3$ and the set of obstacle positions $\{X_i\}_{i=1}^{M} \subset \mathbb{R}^3$ encoded in the state s, we mark a transition unsafe whenever the robot comes closer than a safety margin $d > 0$ to any obstacle:

$$C_d\left(s, a, s'\right) = \mathbb{I}\left[\min_i \|\mathrm{pos}(s) - X_i\| < d\right]$$

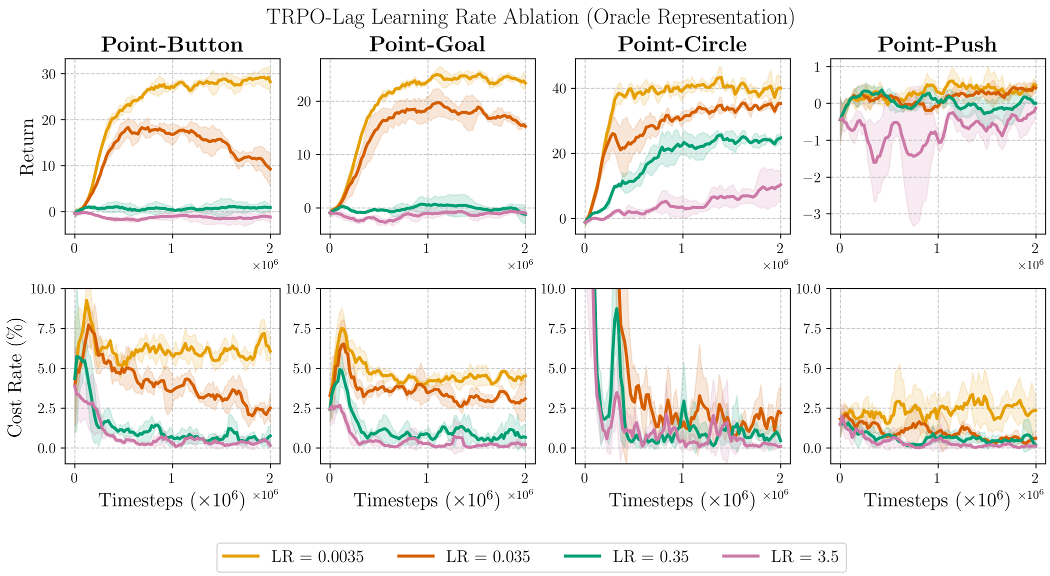

Figure 15: Training Curves for Safe Gymnasium with Point Robot. **(Top)** Average episodic return versus training steps. **(Bottom)** Average episodic cost rate versus training steps.

Task	$h(s)$	E_t	$\nu(h, E)$
Collision avoidance	$\text{pos}(s) \in \mathbb{R}^3$	$\{X_i\}_{i=1}^M \subset \mathbb{R}^3$	$\min_i \lVert h - X_i \rVert_2 - d_{\text{safe}}$
Safety-region compliance	$\text{pos}(s) \in \mathbb{R}^2$	$\mathcal{S}_{\text{safe}} \subset \mathbb{R}^2$	$\text{dist}\big(h, \mathbb{R}^2 \setminus \mathcal{S}_{\text{safe}}\big) - \varepsilon$

Table 2: Examples of function ν for different safety tasks.

Thus $C_d = 1$ whenever the robot violates the distance constraint, encouraging policies that keep a safe distance to obstacles. I

Safety Region Compliance To ensure the robot remains within a designated safety region, we evaluate its position in the next state, $\text{pos}(s') = (x, y) \in \mathbb{R}^2$, against a predefined safe region $\text{safe\_region} \subseteq \mathbb{R}^2$. A penalty is incurred if the position lies outside this region:

$$C_d(s, a, s') = \mathbb{I}[\text{pos}(s') \notin \text{safe\_region}].$$

This cost function assigns a value of 1 when the robot deviates from the safety region, indicating a safety violation.

I Average Cost Minimization and Cost Value Function

Our problem formulation targets minimizing the average cost per time step, distinct from the cumulative discounted cost over an infinite horizon typically addressed by Lagrangian-based methods like TRPO-Lag and PPO-Lag. The connection between cumulative discounted cost and average cost is well-established [Puterman, 2014]:

$$\lim_{\gamma \to 1^-} (1 - \gamma) V_C^\pi(s_0, \theta_0) = \phi^\pi(s_0, \theta_0),$$

where $V_C^\pi(s_0, \theta_0)$ denotes the value function for the cost under policy π starting from state s, parameter θ, and $\phi^\pi(s, \theta) = \lim_{H \to \infty} \frac{1}{H} \mathbb{E}_{\pi^*, T_\theta} \left[\sum_{t=0}^{H-1} C_d(s_t, a_t, s_{t+1}) \mid s_0 = s, \theta_0 \right]$ represents the expected average cost for parameter θ. Thus, by setting the constraint limit to zero in Lagrangian optimization, we ensure the average cost approaches zero, aligning with our safety objectives.

J Experimental Details

For out-of-distribution (OOD) evaluation, we modify Safety Gymnasium task environments—Goal, Button, and Push—by adding two additional hazard locations to increase complexity. For Circle

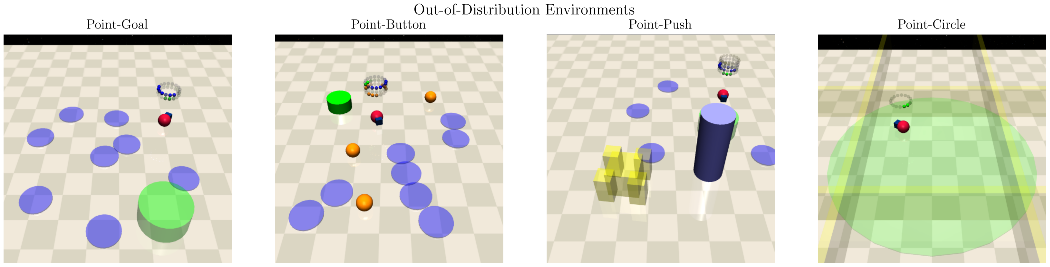

Figure 16: Four out-of-distribution environments for evaluation.

task, we reduce the safety region's x-limit from 1.125 to 1.0 to further challenge the agent's safety mechanisms.

To introduce varying hidden parameters, each episode independently samples density, damping, and gravity multipliers by randomly selecting one of two intervals, $[0.1, 0.25]$ or $[1.75, 2.5]$, with equal probability and uniformly sampling a value from the chosen interval, ensuring diverse environmental conditions.

Training is conducted on an Ubuntu 22.04 server using a Slurm job scheduler, which dynamically allocates computational resources. As resource allocations vary across runs, we do not report runtime comparisons for training.

For runtime execution comparisons, we standardize the hardware to ensure fairness. Experiments are run on a server with 256 GB of memory, dual AMD CPUs (56 cores, 224 threads), and Nvidia RTX A4500 GPUs (20 GB RAM each).