Uncertainty-Aware Safety Propagation Critics for Safe Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

Safe reinforcement learning (RL) aims to optimize long-term performance while satisfying safety constraints, a requirement that is critical in many applications but difficult to guarantee when cost estimates are inaccurate or data is limited. In model-free actor-critic methods, cost critics are often unreliable in poorly explored regions, leading to constraint violations during both training and deployment. In this work, we propose a novel uncertainty-aware approach in safe RL called USPC, which constructs conservative cost surrogates using epistemic uncertainty. Our method trains an ensemble of cost critics to estimate uncertainty and uses these estimates to build an upper confidence bound on predicted costs. We then introduce a safe set network that approximates a pessimistic surrogate of the cost action-value function inspired by safe Bayesian optimization, enabling scalable safety propagation in continuous state-action spaces. Replacing standard cost critics with this surrogate in existing off-policy safe RL algorithms yields policies that are significantly less likely to violate cost constraints. We show empirically across multiple Safety Gymnasium benchmark tasks that our approach consistently reduces both the frequency and magnitude of constraint violations while maintaining competitive reward performance compared to strong baselines. Project website is available at `https://sites.google.com/view/uspcrl`.

1 Introduction

Reinforcement learning (RL) has shown promise in solving complex tasks in diverse domains, including robotic control (Kober et al., 2013), games (Mnih et al., 2015), and natural language processing (Ouyang et al., 2022). However, most of RL's achievements so far have been in simpler simulated or digital environments (Dulac-Arnold et al., 2021), while RL in real-world tasks, such as autonomous driving with a real car (Zhang et al., 2024), faces several obstacles. One major obstacle is that reinforcement learning policies often violate some desired safety constraints (such as an autonomous car with the goal of arriving at a target point should also avoid collisions), which makes collecting real-world data expensive or even outright impossible.

To address this issue, safe reinforcement learning deals with algorithms that learn a cumulative reward maximizing policy while also constraining cumulative costs to be below some predefined threshold. However, learning such a policy faces some further difficulties, especially when black box models such as deep neural networks are used. If the safety constraint is active (i.e., actually constraining the optimal solution), the gradients for reward maximization and cost reduction are often conflicting, which makes learning difficult. Furthermore, without the use of an explicit world model, the predicted future cumulative cost can be inaccurate, and thus constraint satisfaction cannot be guaranteed.

Existing works in safe learning-based control use uncertainty, particularly epistemic uncertainty, in various ways to account for poorly explored regions of the data. Early approaches to safety under uncertainty were explored in the context of safe Bayesian optimization, which leveraged Gaussian process uncertainty estimates to ensure constraint satisfaction while exploring unknown regions (Sui et al., 2015). However, these methods were primarily developed for discrete or low-dimensional settings, limiting their applicability to the continuous state and action spaces, which are typical in RL. Some recent model-based methods have

used the uncertainty of the world model to facilitate safer trajectories, for example, by applying it as an intrinsic penalty for the model predictive control planner in a soft-constrained setting (Lütjens et al., 2019). However, careful tuning is necessary with such algorithms; since the model has no explicit definition of safety, it is possible for the algorithm to yield trajectories that yield seemingly good objective values but would be considered unsafe from a human perspective. Other model-based works have proposed algorithms that use a Gaussian process world model with a reachability-based approach to satisfy hard constraints (As et al., 2025). While such methods often provide strong theoretical guarantees, they also tend to extend poorly to continuous state or action spaces, or large amounts of data. To address this issue, model-free safe RL methods have also received significant attention. In particular, since safety is often defined in terms of cost return in a constrained Markov decision process (CMDP), many methods leverage the epistemic uncertainty in value (or value-like) function estimators to improve safety (Stachowicz & Levine, 2024; Günster et al., 2024).

In this work, we propose Uncertainty-Aware Safety Propagation Critics (USPC) , a novel method for learning cost critics in an uncertainty-aware manner to account for the inaccuracies in cost return estimates. More specifically, we first estimate the cost returns using a well-calibrated ensemble. We then use the epistemic uncertainty in the ensemble to obtain an uncertainty-aware reachability-based pessimistic surrogate for the cost return. When a policy is trained with this surrogate instead of the regular cost critic, the agent is less likely to step into an unsafe trajectory due to model errors in poorly explored regions of the state-action space.

Our contributions can be summarized as follows:

- We propose USPC, a novel algorithm to train a safe set network that substitutes the cost critics in off-policy actor-critic safe RL algorithms. Our proposed safe set approximation enables scalable safety propagation in continuous state-action spaces.
- We empirically demonstrate that our method can be plugged into existing off-policy safe RL methods to obtain policies that yield significantly lower constraint violations while still maintaining competitive reward returns in several tasks.
- We provide a principled discussion of the assumptions and limitations behind ensemble-based uncertainty estimation and Lipschitz-based safety propagation, and explain how these choices lead to conservative behavior in practice.

We demonstrate our contributions with experiments on 6 Safety-Gymnasium tasks. Empirically, we show that USPC yields similar reward curves while keeping costs below constraints more consistently. More specifically, we show that USPC's training episodes are significantly safer compared to the baselines.

2 Related Work

Safe Bayesian optimization. Optimization and exploration under uncertainty has been studied for a long time in the bandit literature (Bubeck et al., 2012). These works were then used in settings where reward functions could be modeled by a Gaussian process, both enabling the usage of confidence bounds and establishing a connection to Bayesian optimization (Srinivas et al., 2009). Safe Bayesian optimization extends this by enforcing safety constraints throughout learning via uncertainty-aware confidence bounds and regularity assumptions. Sui et al. (2015) used confidence bounds and regularity assumptions to maintain a certified set of safe actions and expand it through safety propagation with probabilistic guarantees. Following works have extended this work to multiple safety constraints (Berkenkamp et al., 2023) and stagewise optimization strategies that decouple safety propagation from utility maximization (Sui et al., 2018; Wachi & Sui, 2020). Similar Gaussian process based safety propagation ideas have also been used in other areas such as safe exploration and control settings (Turchetta et al., 2016; Berkenkamp et al., 2017).

While safe Bayesian optimization provides a principled framework for uncertainty-aware safety propagation with strong theoretical guarantees, it is not directly applicable to reinforcement learning immediately due to its reliance on Gaussian processes, discrete action sets, and low-dimensional settings. Our work draws

inspiration from this line of work but replaces Gaussian process confidence bounds and explicit safe set construction with neural approximations that work over continuous state-action spaces. We embed the inspired safety propagation into actor-critic framework by learning a pessimistic cost surrogate function.

Safe model-free reinforcement learning. Safe reinforcement learning has received significant attention following the success of unsafe reinforcement learning algorithms elsewhere. Often in these works, the MDP is replaced with a CMDP, where an episode is considered safe if the episodic cost return is below some threshold. A large proportion of modern safe RL focuses on primal-dual Lagrangian methods in a model-free setting. The simplest of these methods augment standard unsafe RL algorithms with a Lagrangian objective: In the primal step, they obtain the Lagrangian for the constrained optimization problem and maximize for rewards. Then, in the dual step the dual variables are updated. Related to this approach, RCPO (Tessler et al., 2018) proposes a multi-timescale approach where the dual variables are updated less frequently to stabilize training, and PID-Lagrangian (Stooke et al., 2020) extends the basic Lagrangian dual update with a PID controller to improve stability and reduce oscillations. Another seminal Lagrangian-based work is CPO (Achiam et al., 2017), which formulates each policy update as a constrained second-order trust-region optimization problem, ensuring that the new policy remains close (in KL-divergence) to the old one while keeping the expected cumulative constraint costs below specified limits. This design provides near-constraint satisfaction guarantees throughout training, but is computationally costly due to its second-order updates. Future CPO-based work has then improved on it in various way such as PCPO decoupling reward optimization and constraint satisfaction with a projection step (Yang et al., 2020), and FOCOPS replacing the second-order updates with a better scalable first-order step (Zhang et al., 2020). However, these primal-dual methods often suffer from unstable learning dynamics and slow convergence due to coupled training of model parameters and dual variables, hence a variety of primal-only methods have been proposed to avoid those drawbacks. CRPO (Xu et al., 2021) proposes a mechanism to switch gradients between reward improvement when the policy is safe and cost reduction when the policy is unsafe, which PCRPO (Gu et al., 2024) then expands to reduce performance degradations and oscillations near the constraint boundary with a projection approach. IPO (Liu et al., 2020) transforms the constrained problem into an unconstrained one by adding a logarithmic barrier term to the primal objective. Other approaches to safe RL include Lyapunov-based methods (Chow et al., 2018; 2019), action projection and shielding (Dalal et al., 2018; Wagener et al., 2021), and safe set/reachability-based methods using model-free approximations (Wachi & Sui, 2020).

In contrast to these methods, USPC introduces a learned safety surrogate that directly replaces the cost critic and propagates safety information across the action space. This allows safety information obtained at a limited set of actions to generalize to nearby actions, reducing constraint violations caused by sparse or inaccurate cost estimates. As a result, our method preserves the simplicity and scalability of model-free actor–critic algorithms while addressing a core limitation shared by many existing approaches, which enforce safety only at sampled actions or along the current policy trajectory, without explicitly propagating safety to nearby actions in the continuous action space.

Epistemic uncertainty-based methods in deep reinforcement learning.

Epistemic uncertainty is central to a wide variety of problems in reinforcement learning. While most reinforcement learning methods only implicitly reduce the epistemic uncertainty, a significant amount of works explicitly quantify and use epistemic uncertainty to address various problems.

Popularly, epistemic uncertainty-based methods use the uncertainty to guide exploration. One way to use the uncertainty is to construct an intrinsic reward that scales with uncertainty, so that the policy will be more likely to visit these underexplored regions. The intrinsic reward can be constructed in various ways, such as using visitation counts (Auer et al., 2002; 2008), disagreement in ensemble transition models (Shyam et al., 2019; Sekar et al., 2020), information gain (Sun et al., 2011; Houthooft et al., 2017). Some other works utilize epistemic uncertainty to do posterior sampling of value functions (Osband et al., 2016; 2018; Azizzadenesheli & Anandkumar, 2019; An et al., 2021) or to obtain upper confidence bound-based optimistic value function (Lee et al., 2021; Schmitt et al., 2023) to guide exploration.

Another family of works use epistemic uncertainty in safe reinforcement learning. A unique problem in safe RL compared to some other safe control methods is that since the underlying MDP is unknown, exploration is critical for obtaining a near-optimal policy, however the agent often lacks sufficient evidence to accurately

estimate future costs, resulting in unsafe behaviors. Epistemic uncertainty can be used to provide a principled signal for this uncertainty, enabling safe RL algorithms to act conservatively and explore safely. One branch of these methods quantify and use the uncertainty in the world models. In online model-based safe RL, compounding model errors can make seemingly safe predicted trajectories unsafe in reality, hence the epistemic uncertainty can be used to account for this unreliability (Berkenkamp et al., 2017; Wachi & Sui, 2020; As et al., 2025). In offline model-based settings, epistemic uncertainty can additionally help with preventing unsafe behaviors in regions insufficiently covered by the offline data (Yu et al., 2020; Kidambi et al., 2020; Diehl et al., 2023). However, all of these methods can still suffer from model inaccuracy in cases where the environment is too complex to learn dynamics accurately without prohibitively large datasets or additional training, such as high-dimensional, contact-rich or discontinuous environments. In existing model-free safe RL works, uncertainty has been used in diverse ways, including conservative constraint enforcement (Zhang et al., 2024; Stachowicz & Levine, 2024), dual variable tuning in primal-dual methods (Zhang et al., 2024), and safety filters (Eilers et al., 2023; Flögel et al., 2025).

Our work falls into the model-free safe RL category, but we use a Lipschitz safety propagation-based statistic similar to (Sui et al., 2015). To the best of our knowledge, existing model-free works often utilize uncertainty using simpler statistics such as the upper confidence bound, whereas a safety propagation/reachability-based RL approach has only been applied in model-based settings to quantify uncertainty in learned dynamics (Wachi & Sui, 2020; As et al., 2025).

3 Background

3.1 Problem Setting

Constrained Markov decision processes (CMDPs) are a common mathematical framework used for safe reinforcement learning. We define a CMDP by a tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0, C)$, where $\mathcal{S}$ and $\mathcal{A}$ denote state and action spaces, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition kernel that contains probabilities $P(s_{t+1}|s_t, a_t)$, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\gamma \in [0, 1]$ is the discount factor, $\rho_0 \in \mathcal{P}(\mathcal{S})$ is the probability distribution of initial states, and $C = \{c_i : \mathcal{S} \times \mathcal{A} \to [0, c_{max}], i = 1, \ldots, |C|\}$ is the set of cost functions that define the safety related constraints. We denote possible minimum and maximum action values as $a_{\min}$ and $a_{\max}$, respectively. A trajectory is a sequence $(s_t, a_t)_{t=0}^{T}$, where t denotes a timestep and T denotes the horizon (possibly infinite). At every timestep t, the agent observes the state $s_t \in \mathcal{S}$, samples an action $a_t \sim \pi(\cdot|s_t)$, and the environment transitions to $s_{t+1} \sim P(\cdot \mid s_t, a_t)$.

For any stationary policy π, the reward action-value function and the cost action-value function for constraint i are defined as

$$Q_r^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \, r(s_t, a_t) \,\middle|\, s_0 = s,\, a_0 = a \right] , \tag{1}$$

$$Q_{c_i}^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \, c_i(s_t, a_t) \,\middle|\, s_0 = s,\, a_0 = a \right] , \tag{2}$$

where the expectation is taken over trajectories induced by π and P. In this paper, we consider the action-values for the current learned policy by default and omit the π for notational simplicity.

Given a vector of cost thresholds $\boldsymbol{h} := [h_1, \ldots, h_M]^\top \in \mathbb{R}_{\geq 0}^M$, the goal of safe RL is to obtain a safe and optimal policy π^* that maximizes expected reward while satisying cost constraints:

$$\pi^* \in \arg\max_{\pi} \; J_r(\pi) := \mathbb{E}_{s_0 \sim \rho_0,\, a_0 \sim \pi(\cdot|s_0)} \big[Q_r(s_0, a_0) \big] \tag{3}$$

$$\text{s.t.} \;\; J_{c_i}(\pi) := \mathbb{E}_{s_0 \sim \rho_0,\, a_0 \sim \pi(\cdot|s_0)} \big[Q_{c_i}(s_0, a_0) \big] \leq h_i, \quad i = 1, \ldots, |C| \; . \tag{4}$$

In this work, we focus on a single safety constraint (i.e., $|C| = 1$) for clarity, and denote its cost function by c and its threshold by h. Our method can be extended easily to multiple constraints, given that the base safe RL algorithm is compatible.

3.2 Actor-Critic Safe Reinforcement Learning

In actor-critic-based safe RL algorithms, the value functions defined in the previous subsection are represented using neural networks and optimized from replay data. The reward critic $Q_r(s, a)$ and cost critic $Q_c(s, a)$ are trained using temporal-difference (TD) regression on single-step targets:

$$y_t^r = r(s_t, a_t) + \gamma \mathbb{E}_{a' \sim \pi_\theta(\cdot|s')}\big[Q_r(s', a')\big] \ , \quad y_t^c = c(s_t, a_t) + \gamma \mathbb{E}_{a' \sim \pi_\theta(\cdot|s')}\big[Q_c(s', a')\big] \ . \tag{5}$$

The policy is then updated using a constrained objective based on the critics. A common approach is to form a Lagrangian where a multiplier $\lambda \geq 0$ penalizes unsafe behavior:

$$\mathcal{L}(\theta, \lambda) = \mathbb{E}_{s,a \sim \pi_\theta}\big[- Q_r(s, a) + \lambda(Q_c(s, a) - h)\big] \ . \tag{6}$$

This results in alternating gradient updates on θ and dual updates on λ, and covers many primal-dual safe RL algorithms. Variants include penalty methods, trust-region formulations, and maximum-entropy actor-critic objectives.

Throughout this work, we follow this standard deep actor-critic paradigm, where the policy is represented by a neural network $\pi_\theta(a|s)$, and the action-value functions Q_r and Q_c are learned by minimizing TD losses over replayed transitions. This forms the basis for our deep safe learning setting.

3.3 Safe Bayesian Optimization

Safe Bayesian optimization provides a framework for decision making under uncertainty when function evaluations are expensive and safety constraints must be respected throughout the learning process. Rather than treating uncertainty only for exploration, safe Bayesian optimization also leverages epistemic uncertainty to prevent unsafe evaluations while gradually expanding the set of admissible decisions.

The main technique used in safe Bayesian optimization is the construction of uncertainty-aware confidence bounds. Given noisy observations of an unknown quantity, a Gaussian process with posterior mean $\mu_t(x)$ and variance $\sigma_t^2(x)$ is used. Then, uncertainty estimates are constructed using confidence intervals:

$$l_t(x) = \mu_t(x) - \beta_t^{1/2}\sigma_t^2(x), \quad u_t(x) = \mu_t(x) + \beta_t^{1/2}\sigma_t^2(x) \ , \tag{7}$$

where β_t is chosen to control the level of conservatiness. In Gaussian process works, β_t is typically chosen as a function of time to ensure that the confidence intervals contain the true function with true probability, which enables formal safety and regret guarantees under standard regularity assumptions (Srinivas et al., 2009; Sui et al., 2015; Rothfuss et al., 2023).

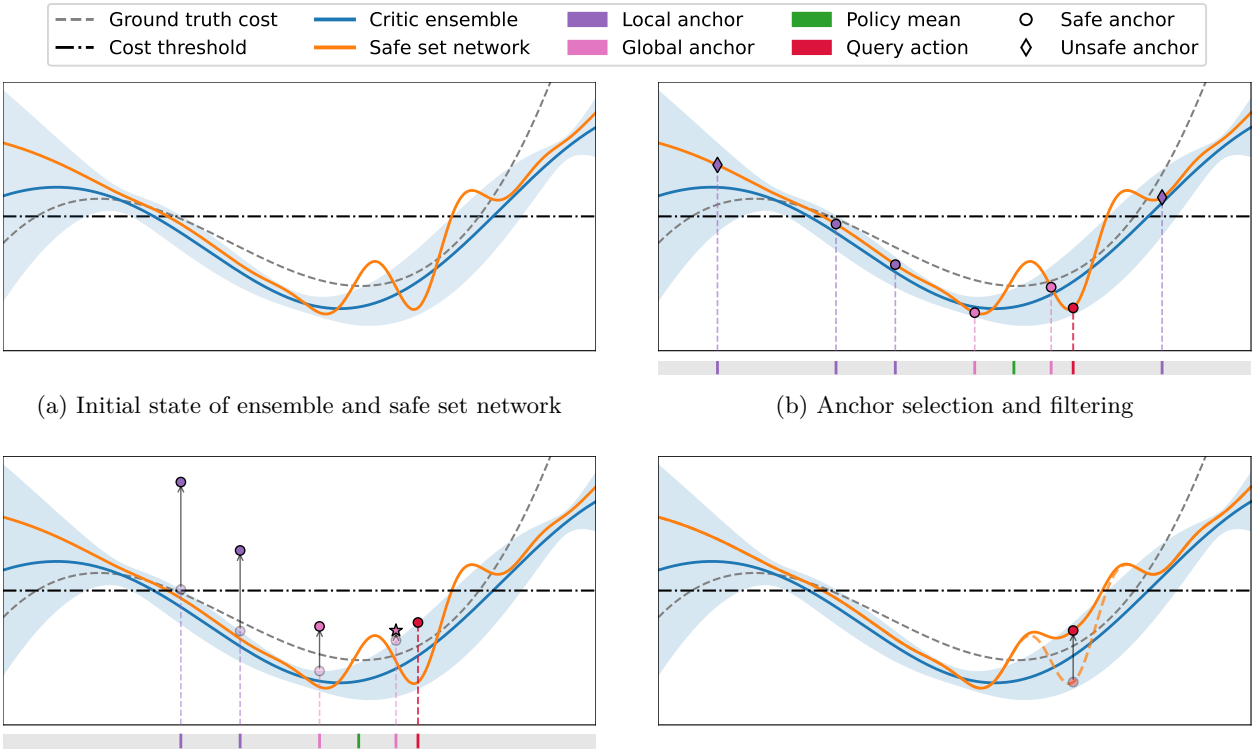
Sui et al. (2015) leverages this construction to define safety in a reachability-based manner. Rather than certifying safety only at individual decisions, they maintain a safe set that is iteratively expanded over time. At iteration t, a decision is considered safe if its upper confidence bound satisfies the safety constraint, or if it can be certified as safe by propagating safety from previously safe decisions using a regularity assumption. Assuming Lipschitz continuity with L with respect to some distance metric d, safety is propagated to decision x if:

$$\min\{u_t(x), u_t(x') + Ld(x - x')\} \leq h \ . \tag{8}$$

This allows us to conservatively infer safety at unobserved points that are sufficiently close to the known safe ones.

The use of Gaussian process models enables principled uncertainty quantification and high-probability guarantees. However, this approach relies on assumptions that limit scalability, including the need to evaluate confidence bounds over discrete or low-dimensional domains and the cubic computational cost of Gaussian process inference. These limitations make direct application of these methods infeasible in reinforcement learning settings with continuous state and action spaces.

Nevertheless, the mathematical structure underlying this line of work is relevant to safe reinforcement learning. The combination of uncertainty-aware upper confidence bounds and Lipschitz-based safety propagation motivates the construction of conservative surrogates for cost evaluation. Our method adopts this structure while replacing Gaussian processes and explicit safe set representations with scalable neural approximations that are compatible with reinforcement learning.



(a) Initial state of ensemble and safe set network

(b) Anchor selection and filtering

(c) Safe set network target computation

(d) Fitting the safe set network

Figure 1: An illustration of one step of USPC training on a simple one-dimensional example. (a) Initial state of the critic ensemble and safe set network. Shaded blue region shows the confidence interval for the ensemble. (b) A query point (for safe set network update) is selected, and local and global anchors are sampled to create the anchor set. Anchor points are then filtered according to the current safe set network and the cost threshold, and unsafe points are discarded. (c) Safe anchor points are evaluated on the UCB, then shifted upwards by the distance term to obtain candidate targets. The minimum of the candidates (denoted by star) is selected as the target for the safe set network update. (d) Safe set network is updated at the query point via regression towards the target selected in the previous step.

4 Uncertainty-Aware Safety Propagation Critics

In this section, we present USPC, **U**ncertainty-Aware **S**afety **P**ropagation **C**ritics for Safe Reinforcement Learning, which combines ensemble-based cost estimation with a scalable safety propagation mechanism that can be seamlessly integrated into off-policy actor-critic algorithms. The overall USPC mechanism is illustrated in Figure 1 on a simple one-dimensional example. A high level overview of safe RL training with USPC can be seen in Figure 2.

4.1 Ensemble Critics

Standard actor-critic RL methods train a single cost critic network Q_c. Instead, we train an ensemble of M critics, denoted $\{\hat{f}_{\theta_i}\}_{i=1}^{M}$, where each member $\hat{f}_{\theta_i} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is initialized independently and trained through the same stochastic gradient updates using the same replay batches. Intuitively, the ensemble captures epistemic uncertainty arising from limited data or poorly explored regions, which allows us to construct confidence intervals similar to those in Gaussian process based safe Bayesian optimization. To construct meaningful confidence intervals, we treat the outputs of the ensemble members as random samples from an underlying distribution, and we more formally justify this view under mild assumptions:

Assumption 1. *Let $\hat{f}$ be an ensemble consisting of members $\hat{f}_{\theta_i} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ for $i = 1, \ldots, M$ trained to fit a target function f, where the parameters θ_i are initialized independently and are identically distributed. Suppose that all $\hat{f}_{\theta_i}$ are trained in parallel using the same data batches presented in the same order. For a given (s, a), let $\hat{y}_i(s, a) = \hat{f}_{\theta_i}(s, a)$ denote the member outputs, and define their sample mean and variance as $\hat{\mu}(s, a)$ and $\hat{\sigma}^2(s, a)$, respectively. Then, conditional on the training process (i.e. on the fixed data and shared batch sequence, such that the only randomness comes from the initialization of weights), the outputs $\hat{y}_i(s, a)$ are i.i.d. random variables.*

In regions of the input space that are sufficiently covered by data, ensemble members can be treated as approximately unbiased estimators of the true function f. This, alongside Assumption 1, implies that by the Central Limit Theorem, for sufficiently large M, the sampling distribution of $\hat{\mu}(s, a)$ more specifically converges to

$$\hat{f}(s, a) := \hat{\mu}(s, a) \sim \mathcal{N}\left(f(s, a), \frac{\hat{\sigma}^2(s, a)}{M}\right).$$

Assumption 1 follows from the fact that each ensemble member is independently initialized using the same prior distribution for θ_i and trained through the same deterministic procedure. Conditional on the shared training data and batch order, the outputs are i.i.d., and the Central Limit Theorem implies that their empirical mean distribution approaches a Gaussian as the ensemble size M grows.

Using ensemble mean and variance statistics, we construct an upper confidence bound (UCB) on the cost action-value function:

$$\mathrm{UCB}(s, a) := \hat{\mu}(s, a) + \beta \hat{\sigma}(s, a) \ , \tag{9}$$

where $\beta \geq 0$ in our work is a hyperparameter controlling the level of conservativeness. We use this bound as a pessimistic surrogate for cost evaluation, increasing predicted costs in regions with high uncertainty. A visualization of the UCB is available in Figure 1a (and the following subfigures).

4.2 Safe Set Network

The critic ensemble Q_c^{ens} that we train is not directly used to enforce safety. Instead, we train a safe set network (SSN) that approximates the safety of a state-action pair, denoted by $\hat{S}_c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, as a substitute for the cost critics. SSN allows us to extend the safe set approach in reachability-based safe BO methods to continuous functions where the input space is infinite, while also scaling better than GPs used in these methods.

We start by considering the GP-based definition of the safe set (Sui et al., 2015) with the Euclidean distance metric for a simplified finite action space RL setting. For finite settings, the safe set at state s and training step t for cost function c, denoted as $S_{c,t}$ [1] , is defined as the set of actions that are deemed to be safe at s:

$$S_{c,t}(s) = \bigcup_{a \in S_{c,t-1}(s)} \{a' \in \mathcal{A} \,|\, \mathrm{UCB}(s, a) + L_{Q_c} \|a - a'\|_2 \leq h\}, \tag{10}$$

where L_{Q_c} is the assumed Lipschitz constant for the cost Q-value Q_c. which can be equivalently written as

$$S_{c,t}(s) = \{a' \in \mathcal{A} \,|\, \min_{a \in S_{c,t-1}(s)} \mathrm{UCB}(s, a) + L_{Q_c} \|a - a'\|_2 \leq h\}. \tag{11}$$

In order to extend this definition to the RL training setting in continuous action spaces, we make three changes. First, we define the safe state network $\hat{S}_c$ to fit to the left hand side of the inequality in (11), which allows us to redefine safety for continuous spaces. Specifically, we call a state-action pair (s, a) safe if we have $\hat{S}_c(s, a) \leq h$, which is defined for continuous state and action spaces. Second, analogously to how Q-functions are learned, we introduce a target network $\hat{S}_c^{\text{target}}$ and use it to select safe actions in the computation of the minimum, which allows us to avoid retraining a safe set network at every time step.

[1] In GP-based works, it is usually assumed that a "seed set" is given to initialize $S_{c,0}$. This assumption is usually needed in proofs of safety guarantees. Since we are working in a purely online setting we do not assume a seed set; however, in an offline safe RL setting this may be applicable.

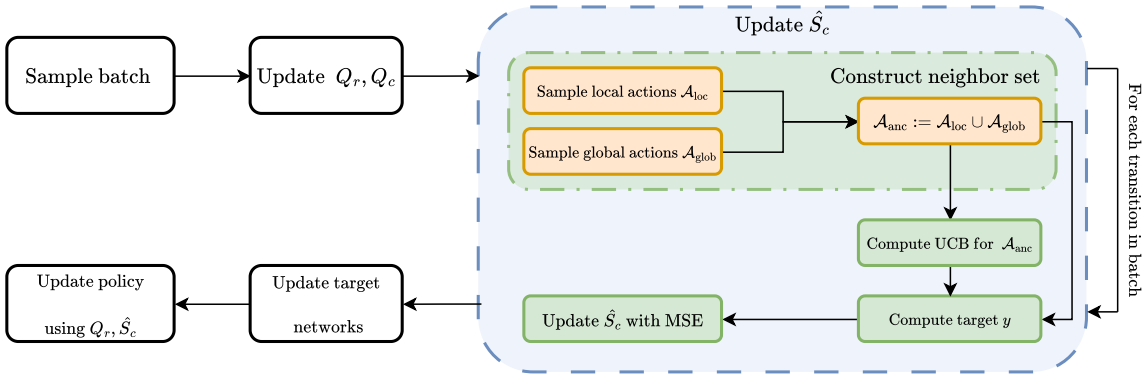


Figure 2: A high level overview of USPC.

Third, since we cannot evaluate every action in the space for minimization, we instead check only the actions in a finite "anchor set" $\mathcal{A}_{\text{anc}}$. Consequently, our training target for $\hat{S}_c(s, a')$ becomes

$$y(s, a') \leftarrow \min_{\substack{a \in \mathcal{A}_{\text{anc}} \\ \hat{S}_c^{\text{target}}(s,a) \leq h}} \text{UCB}(s, a) + L_{\hat{Q}_c} \|a - a'\|_2 \tag{12}$$

where $\hat{Q}_c$ is the Lipschitz constant for the learned cost critic [2] , and $\alpha \in [0, 1]$ is the hyperparameter for the exponentially moving average update. We depict a simplified version of the anchor filtering, target computation and and safe set network update in Figures 1b, 1c and 1d, respectively.

4.3 Anchor Set Generation

Since the state is not directly controlled by the agent, we define propagation only in the action dimension. Thus, we train S_c by varying actions for a fixed state. In order to approximate doing a pass on the entire action space, we create an anchor set with local and global samples. Local samples, denoted by $\mathcal{A}_{\text{anc}}^{\text{loc}}$ are sampled from a Gaussian input around the policy action. For deterministic policies we use a Gaussian centered around the selected action (e.g. the exploration noise distribution), while for stochastic policies the policy's action distribution output can be directly used instead. Global actions, denoted by $\mathcal{A}_{\text{anc}}^{\text{glob}}$ are sampled uniformly from the action space (assuming that it is bounded). Finally, we also include the queried point itself, since intuitively if its own UCB is below the cost threshold then the point itself can be considered safe. The anchor set for the queried state-action pair (s, a) is then defined as $\mathcal{A}_{\text{anc}} := \{a\} \cup \mathcal{A}_{\text{anc}}^{\text{loc}} \cup \mathcal{A}_{\text{anc}}^{\text{glob}}$.

The anchor set allows us to efficiently approximate a full action space scan, which would only be tractable in discrete and bounded settings. In continuous domains, the global anchors $\mathcal{A}_{\text{anc}}^{\text{glob}}$ emulate the full scan of the action space by querying the regions that are not often visited by the policy. However, we also include local anchors $\mathcal{A}_{\text{anc}}^{\text{loc}}$. This is motivated by the observation that for most policy gradient methods, a small step size in parameter updates tends to yield small changes in actions on average (although this is not guaranteed). This is especially the case for TRPO-based methods (Schulman et al., 2017), where a KL divergence constraint is explicitly used in order to limit the change in action distributions. Thus, we include the local anchors in order to get a better estimate around the existing policy behavior. Figure 1b illustrates how anchor points are sampled both locally (with respect to the policy) and globally over the action set.

The anchor set construction and the overall USPC algorithm can be found in Algorithm 1. USPC adapted versions of off policy algorithms can be found in Appendix A.1 in detail.

[2]It is possible for the true Q-function to be discontinuous, however since in practice we always work with a neural network cost critic, we assume that we are always using a continuous estimate for the true function. See the "Lipschitz assumptions" section in discussion for more details.

Algorithm 1 Safe set net update

1: **Input:** Batch size B, particle sizes $K^{\text{loc}}, K^{\text{glob}}$, cost critic Lipschitz constant L
2: **Output:** Updated policy parameter θ_{i+1}
3: Sample B transitions $\{(s_b, a_b, r_b, c_b, s_{b+1})\}_{b=1}^{B}$ from replay buffer
4: **for** $b = 1, \ldots, B$ **do**
5: Sample K^{loc} local actions $\mathcal{A}_{\text{anc}}^{\text{loc}} := \{a_{b,j}^{\text{loc}}\}_{j=1}^{K^{\text{loc}}}$ for s_b as $a_{b,j}^{\text{loc}} \sim \mathcal{N}(\mu_{\theta_i}, \sigma_{\theta_i}^2), \quad j = 1, \ldots, K^{\text{loc}}$.
6: Sample K^{glob} global actions $\mathcal{A}_{\text{anc}}^{\text{glob}} := \{a_{b,j}^{\text{glob}}\}_{j=1}^{K^{\text{glob}}}$ for s_b as $a_{b,j}^{\text{glob}} \sim \text{Uniform}(\mathcal{A}), \quad j = 1, \ldots, K^{\text{glob}}$.
7: Set anchor set as $\mathcal{A}_{\text{anc}} := \{a_b\} \cup \mathcal{A}_{\text{anc}}^{\text{loc}} \cup \mathcal{A}_{\text{anc}}^{\text{glob}}$
8: Clamp actions in $\mathcal{A}_{\text{anc}}$ to be in $[a_{\min}, a_{\max}]$
9: Compute target

$$y(s_b, a_b) = \min_{\substack{a' \in \mathcal{A}_{\text{anc}} \\ S_c^{\text{target}}(s_b, a') \leq h}} \text{UCB}(s_b, a') + L_{\hat{Q}_c} \|a_b - a'\|_2$$

10: **end for**
11: Update $\hat{S}_c$ with loss $\frac{1}{B} \sum_{i=1}^{B} \left(S_c(s_i, a_i) - y(s_i, a_i)\right)^2$
12: Update $\hat{S}_c^{\text{target}}$ via EMA towards S_c

4.4 Exploration and Conservativeness

At each iteration t, the ensemble of cost critics provides multiple predictions for the cost action-value function at a given state-action pair. Motivated by optimism-based methods for exploration (Sui et al., 2015; Lee et al., 2021) we construct an upper confidence bound on the cost action-value function as in (9) using Assumption 1, where β is a hyperparameter controlling the degree of conservativeness. Intuitively, this bound increases the cost estimates in regions where ensemble disagreement is high, corresponding to areas where the learned critics are less reliable.

Unlike classical UCB methods, which use optimism to encourage exploration, we use the upper confidence bound as a pessimistic surrogate for safety evaluation. In explored regions where ensemble members agree, the confidence bound closely matches the mean estimate. In contrast, in poorly explored regions, the bound increases and discourages unsafe actions that rely on uncertain cost predictions. This ensemble-based confidence construction is simple to compute and scalable to continuous domains.

To discuss the conservativeness introduced by this construction, we first consider mild regularity assumptions on the true cost action-value function. In particular, we assume that for each fixed state, the true cost action-value function $Q_c(s, \cdot)$ is L_{Q_c}-Lipschitz with respect to the action:

$$|Q_c(s, a) - Q_c(s, a')| \leq L_{Q_c} \left\|a - a'\right\|_2, \qquad s \in \mathcal{S},\ a, a' \in \mathcal{A}\ . \tag{13}$$

Under this regularity assumption and assuming that the true cost action-value function resides under the ensemble confidence bound, then the true cost of an action can be upper bounded by the confidence bound of a nearby action plus a Lipschitz correction term:

$$Q_c(s, a) \leq Q_c(s, a') + L_{Q_c} \left\|a - a'\right\|_2 \leq u_t(s, a') + L_{Q_c} \left\|a - a'\right\|_2, \qquad s \in \mathcal{S},\ a, a' \in \mathcal{A}\ . \tag{14}$$

This observation implies that the targets used to train the safe set network provide upper bounds on the true cost values whenever the conservativeness hyperparameter is picked properly. As a result, when the safe set network predicts that a state-action pair is safe, the true cost is unlikely to exceed the specified threshold, even in the presence of approximation error or limited data.

5 Experiment Results

5.1 Setup

We use FSRL (Liu et al., 2024) and OmniSafe (Ji et al., 2024) frameworks for our implementation. OmniSafe is used for TD3, CPO, TD3Lag and USPC-TD3Lag experiments, whereas for CVPO and USPC-CVPO we

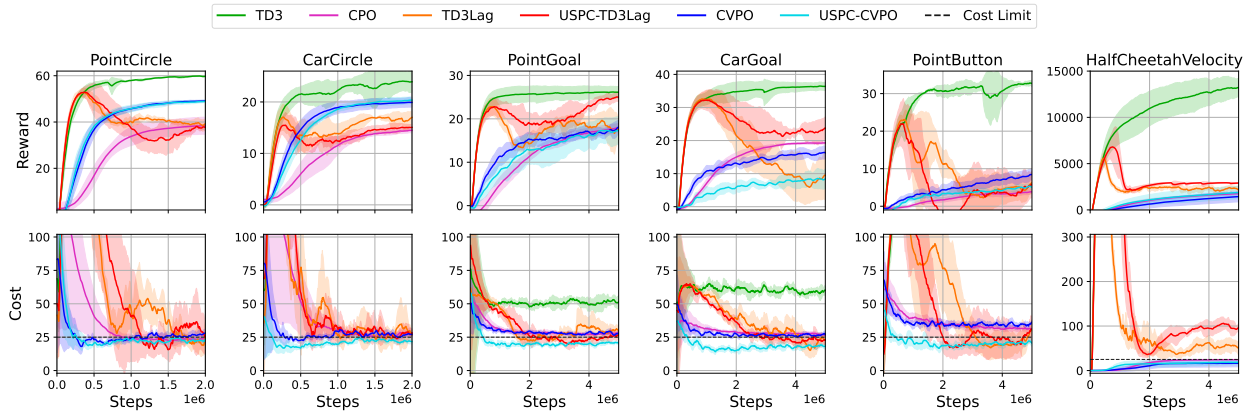


Figure 3: Training reward and cost curves for baselines and our method, each column representing a task. Solid lines represents the mean and shaded regions represent the 95% confidence interval, across 5 random seeds.

use FSRL. For our runs, we used the default applicable hyperparameters provided with these frameworks, and tuned the ones introduced by USPC (see Appendix B). For each algorithm-task pair, we report our results on 5 random seeds. Our experiments were run on a computer with an NVIDIA GeForce RTX 3090 GPU.

In our experiments, we use a subset of tasks found in Safety Gymnasium (Ji et al., 2023). Our Safety Gymnasium tasks have the following agents:

- Point: A 2-dof spherical robot with two joints; one for forward-backward motion and one for rotation. Actions contain the force in the forward-backward direction and the angular velocity for rotation.
- Car: A 3-dof robot with two parallel front wheels and one rear wheel. Actions contain the tuple of forces (in newtons) in each front wheel joint.
- HalfCheetah: A 8-dof robot consisting of 9 body parts (two legs and a body connecting them).

We test these agents on the following tasks:

- Circle: Rewards are given for following a predefined circle. Costs are incurred if the agent moves outside the boundaries (upper and lower limits on the x-axis).
- Goal: Rewards are given for moving towards a goal position. Costs are incurred for going through circular hazard zones scattered around the environment.
- Button: Rewards are given for pressing the correct goal button. Costs are incurred for going through circular hazard zones or colliding with dynamic moving obstacles in the environment.
- Push: Rewards are given for pushing a target object to the goal position. Costs are incurred for passing through circular hazard zones. Unlike the Goal task, the environment also contains pillars that may block the movement of the agent and the target object.
- Velocity: Rewards are given for moving in the +x direction. Costs are incurred if the velocity for the current time step exceeds a velocity threshold.

For all tasks, we use a common cost threshold of 25 as our safety criterion.

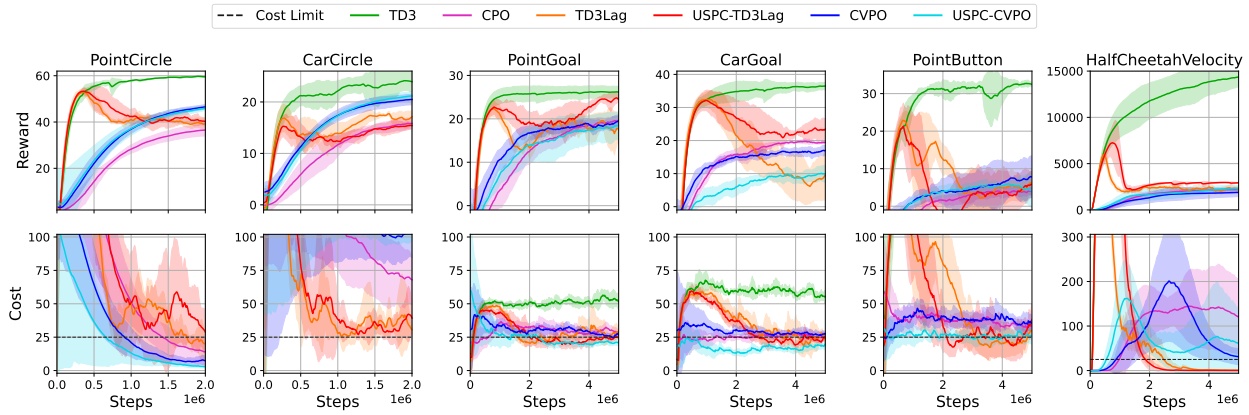


Figure 4: Evaluation reward and cost curves for baselines and our method, each column representing a task. Solid lines represents the mean and shaded regions represent the 95% confidence interval, across 5 random seeds.

5.2 Baselines

We run our experiments with two USPC variants alongside four model-free RL baselines. As an unsafe RL baseline, we use TD3 (Fujimoto et al., 2018), a popular off-policy actor-critic algorithm. Similarly, to be able to make a comparison with on-policy safe RL methods, we use CPO (Achiam et al., 2017) which is a trust region-based second-order algorithm. Finally, we use TD3Lag, a standard primal-dual method, and CVPO (Liu et al., 2022), a variational inference-based method, as our off-policy safe RL baselines. For a direct comparison, we then implement USPC on TD3Lag and CVPO.

5.3 Results

Reward and cost performances. Figures 3 and 4 show the performance of all methods in various environments in training and evaluation episodes, respectively. Note that for some tasks, TD3's cost curves are not visible because it yields extremely high costs compared to safe RL algorithms (see Appendix C for full plots). We see that for most tasks, USPC variants yield lower or equal costs compared to the base algorithms, while still yielding similar rewards. Interestingly, in some tasks (PointGoal and CarGoal) USPC variants achieve higher reward returns while exhibiting similar cost performance compared to their base algorithms. This behavior is consistent with our design: by using uncertainty-aware safety propagation, USPC reduces excessive conservativeness induced by noisy cost critics in well-explored regions while still encouraging safety. As a result, the policy can exploit genuinely safe, high-reward behaviors that are incorrectly penalized by standard cost critics, leading to improved reward without increasing constraint violations. We also note that the largest drop in reward performance occurs in HalfCheetahVelocity. This is

Training and evaluation cost discrepancy. It can be observed that while for most tasks CVPO and USPC-CVPO have lower costs in training than validation, the opposite occurs for HalfCheetahVelocity. We believe that one reason for this may be that the substitution of USPC morphs the estimated safe action region so that the mean action falls outside the safe region. Thus, although the training policy (which has most of its probability mass in the safe region) is safe, the evaluation policy (which simply uses the mean action) is not.

Constraint violation rates. Figure 5 shows the percentage of episodes that violate the cost constraint for all methods. Note that we omit TD3 since it violates constraints in almost all episodes on all tasks (see Appendix C). In nearly all tasks, USPC methods exhibit lower median and mean violation rates than the corresponding baselines. This reduction is particularly observed in many tasks where USPC-CVPO achieves the lowest violation percentages among all methods by a clear margin.

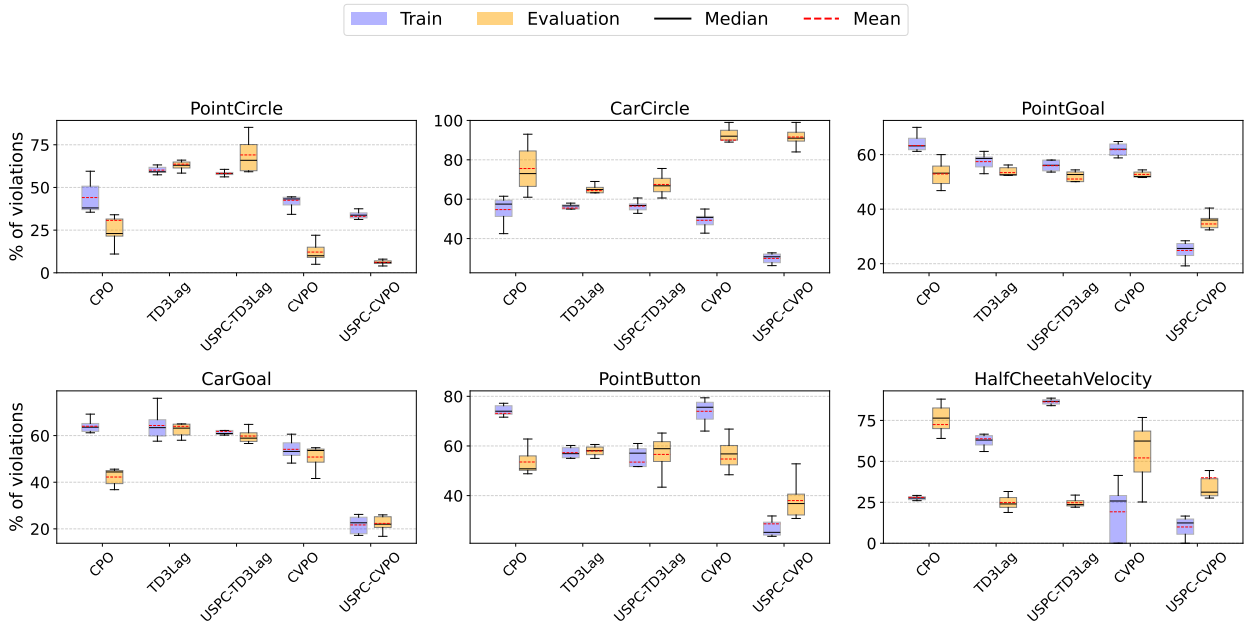


Figure 5: Box plots representing violation rates over episodes during training and evaluation episodes, for experiment runs across 5 random seeds. Shaded boxes represent a 50% confidence interval whereas the lower and upper extremes represent the minimum and maximum, respectively.

Effectiveness of USPC variants. We observe that USPC yields larger reductions in cost (and consequently lower constraint violation rates) when combined with CVPO than with TD3Lag. We believe this is because CVPO explicitly relies on accurate action-level cost estimates during its constrained E-step optimization, whereas TD3Lag only uses cost estimates through a scalar Lagrangian penalty. SSN targets are computed over an anchor set rather than a single action, which yields a quantity that is smoother and less sensitive to small changes or noise in the actions, and hence CVPO benefits more from USPC.

5.4 Ablation Study

We ablate our choice of Lipschitz constant L and confidence bound parameter β in the PointGoal task with USPC-CVPO, shown in Figure 6a and 6b, respectively.

Ablating Lipschitz constant. In the L ablation, we test three values of L: $L = 0$ which corresponds to simply using the UCB with the anchor set, $L = 2$ which is our default, and $L = 5$ as a more conservative estimate. Our results show that although all three values of L eventually converge to similar cost returns, increasing L allows the policy to become safe earlier, at the cost of lower reward returns. This result aligns with our intuition: A lower L makes it more likely for the SSN target to select an anchor point that yields a lower result than the queried point as the minimizer, which reduces the conservativeness of the policy and allows it to pursue higher rewards at the cost of higher costs. On the other hand, a higher value of L hinders exploration but allows faster convergence within the (now smaller) feasible region.

Ablating confidence interval width. In the β ablation, we again test three values of β: $\beta = 0$ which corresponds to using the ensemble mean instead of UCB, $\beta = 2$ which is our default, and $\beta = 5$ for a more conservative UCB. We see that there is a clear trend: A higher value of β results in cost returns converging at lower values, at the cost of lower rewards. For $\beta = 0$ we see that although the converged reward is significantly higher, cost converges around the threshold, which means that the constraint is violated (albeit slightly) in most episodes. On the other hand, $\beta = 5$ yields a very safe policy but the rewards are also much lower. $\beta = 2$ provides a good balance of rewards and safety.

6 Discussion

In this section, we elaborate on our design choices and aspects of our proposed method. We also consider our method's limitations and discuss future work.

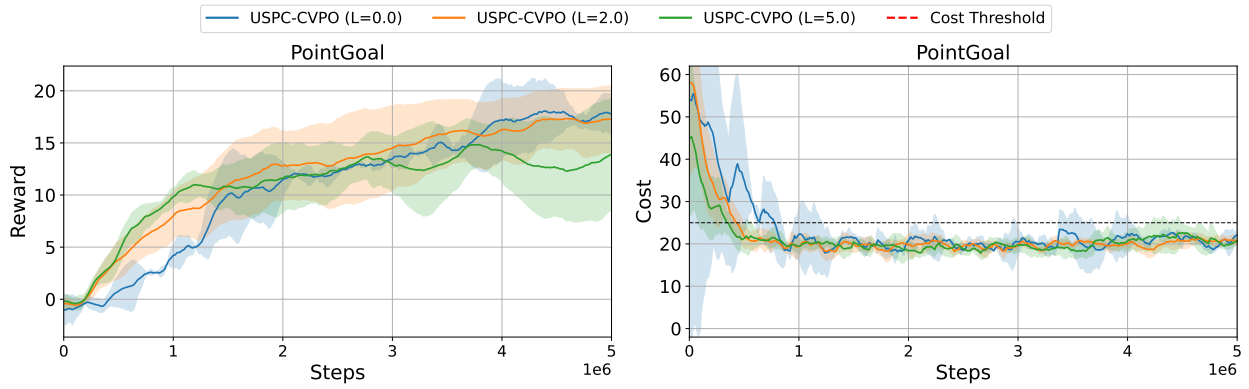
Approximating the uncertainty with neural networks. A common limitation of previous GP-based works is that for higher dimensional inputs, GPs tend to scale poorly in terms of computational cost for high dimensions and cannot be easily applied to continuous spaces, which is why we propose a neural network-based uncertainty approximation, which does not suffer from these issues. Common choices include Bayesian neural networks (Neal, 2012; Blundell et al., 2015), Monte Carlo dropout (Gal & Ghahramani, 2016), and ensemble methods (Osband et al., 2016; Lakshminarayanan et al., 2017). We chose to use an ensemble approach to estimate uncertainty in the cost returns as it usually requires the least amount of modifications in the original method, since many reinforcement learning methods already use clipped double Q-learning (Fujimoto et al., 2018) while training critics which already introduces multiple critics estimating the cost return and could easily be extended to an arbitrary number of critic networks. However, it should be noted that taking the minimum over target networks to obtain a common training target breaks the independence assumption (Assumption 1), so in our implementation we compute separate targets for each ensemble member.

Computing SSN targets. In the computation of SSN targets, an alternative approach is to perform gradient-based inner loop optimization to approximately solve the optimization problem in Equation 11 over the entire action space instead of an anchor set. In theory, this approach would yield more stable and accurate targets. However, in our early experiments we observed that this approach significantly increased the training time while not providing any noticeable benefit in terms of reward or cost returns. Thus, we use our anchor set-based approach that is computationally much cheaper.

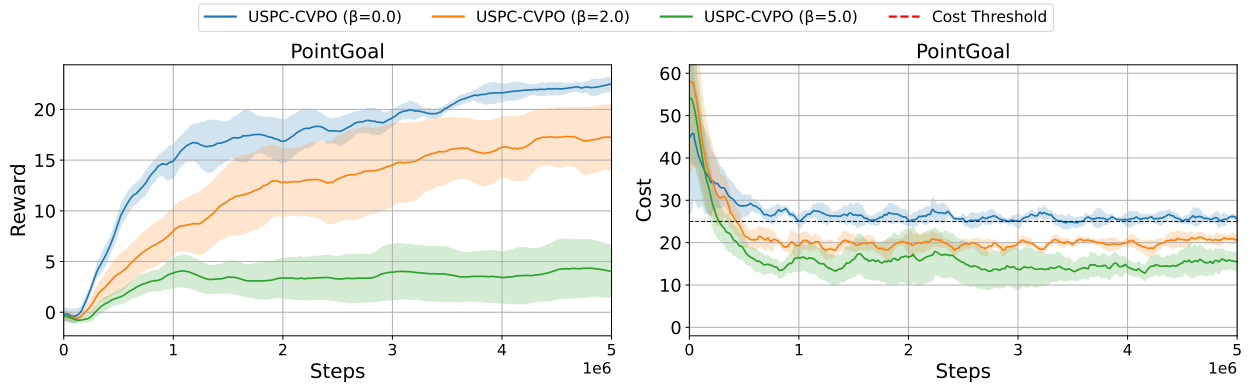
We include the queried point itself in the anchor set for our SSN calculation, which has been proposed by Sui et al. (2015). The objective at the queried point collapses to the UCB at that point (since the distance term becomes 0), which means that we may consider the queried point without "certifying" it using another known safe point. We use this approach mainly because it reduces the reliance on the assumed Lipschitz constant, which in our work is a hyperparameter and may not always be chosen accurately.

Relationship to safe Bayesian optimization. Our method is related to safe Bayesian optimization, especially methods such as Sui et al. (2015); Berkenkamp et al. (2023), which combine uncertainty-aware confidence bounds with safety propagation under Lipschitz assumption. This work can be viewed as a scalable version of this framework in reinforcement learning. The ensemble of cost critics plays the role of an uncertainty estimator and SSN approximates the safety propagation in continuous state action spaces. Unlike safe Bayesian optimization, our method must account for sequential decision-making, function approximation, and nonstationary data distributions caused by policy learning. By embedding the safe set construction directly into an actor-critic framework, we preserve the core safety principles of safe Bayesian optimization while extending them to high-dimensional reinforcement learning problems with continuous state action spaces.

Lipschitz assumptions. Our work assumes that the *learned* cost critic $\hat{Q}_c$ is globally L-Lipschitz with respect to actions for any fixed state, for some L. This assumption does not hold for the *true* Q_c in general, since the cost function c in our tasks can be discontinuous. However, it does hold for $\hat{Q}_c$ since in practice we use neural networks that have finite linear layers and weights (and therefore finite operator norms) and Lipschitz activation functions (most activation functions such as ReLU and hyperbolic tangent are Lipschitz), thus $\hat{Q}_c$ itself is also Lipschitz. However, it is difficult to get a value for L. One option is to derive a value mathematically. Although some previous literature get an estimate for L using the neural network's architecture (Virmaux & Scaman, 2018; Bjorck et al., 2021), we do not wish to make any further assumptions about the critic networks, so we consider these out of the scope of our work. Another option is to estimate the value in a data-driven fashion, such as by maximizing a finite-difference slope estimate or the gradient itself over the anchor set. However, a data-driven approach often increases computational costs further, and more importantly destabilizes the training by moving the targets for SSN. Therefore, we use the third approach, which is treating L as a hyperparameter. Since obtaining a tight value is already



(a) L ablation.



(b) β ablation.

Figure 6: Ablation studies for USPC-CVPO on PointGoal (training episodes). (a) Effect of the Lipschitz constant L. (b) Effect of the uncertainty scaling parameter β.

difficult, having L be a hyperparameter allows us to tune it for the desired balance of safety versus reward performance, and results in a more stable SSN.

Effect of substituting SSN on safety constraints. Substituting SSN in place of the cost critic improves safety by modifying the constraint in the underlying optimization problem. Specifically, as discussed in 4.4, SSN does not provide a strict upper-bound for the cost critic; it upper-bounds the cost critic with high probability instead. What this means is that substituting SSN converts the optimization problem from

$$\begin{aligned} \pi^* \in \arg\max_{\pi} \;& \mathbb{E}_{s_0\sim\rho_0,\, a_0\sim\pi(\cdot|s_0)}\big[Q_r(s_0,a_0)\big] \\ \text{s.t.} \;& \mathbb{E}_{s_0\sim\rho_0,\, a_0\sim\pi(\cdot|s_0)}\big[Q_c(s_0,a_0)\big] \leq h \end{aligned} \tag{15}$$

to

$$\begin{aligned} \pi^* \in \arg\max_{\pi} \;& \mathbb{E}_{s_0\sim\rho_0,\, a_0\sim\pi(\cdot|s_0)}\big[Q_r(s_0,a_0)\big] \\ \text{s.t.} \;& \Pr\left(\mathbb{E}_{s_0\sim\rho_0,\, a_0\sim\pi(\cdot|s_0)}\big[Q_c(s_0,a_0)\big] \leq h\right) \geq 1-\delta \end{aligned} \tag{16}$$

for some δ. In other words, the substitution of SSN changes the original expected cost constraint into a high-probability constraint on the expected cost. Intuitively, for sufficiently high β, the feasible region of (16) will be a subset of the feasible region of (15) with high probability, meaning that a policy that is feasible for (16) is not necessarily feasible for (15). However, empirical results show that the USPC variants almost always have lower cost returns and constraint violation rates (see Figures 3 and 4), so we argue this does not pose a problem in practice.

Conservativeness-performance trade-off. Naturally, increasing conservativeness (i.e., shrinking the feasible region) by substituting our SSN for the cost critic results in worse reward returns. This is also reflected in the experiment results (Figs. 3-4): USPC variants of algorithms tend to have lower reward returns compared to non-USPC ones. However, the added conservativeness also results in significantly lower constraint violation rates (both in training and evaluation episodes), fewer training episodes until the policy becomes safe, and overall lower cost returns (meaning that violation magnitudes are also smaller even for unsafe episodes). Since our main goal is to handle hard constraints even during training, we believe that this is an acceptable trade-off, although for some tasks the degree of conservativeness can be tuned (via the parameter β) to obtain higher task rewards if a better reward return is desired.

Limitations and future work. Like all methods, our method introduces limitations (other than what is inherited from the base safe RL algorithm used), some of which we discuss here. First is the introduction of additional models that need to be learned: an additional SSN needs to be trained, and we also assume the cost critics are neural network ensembles rather than single or double Q-networks. This naturally increases computational costs; training with our method takes around twice as long in our setup compared to the base algorithms. Training the SSN with targets computed using the learned critics also introduces some instability. Alternative model architectures (e.g., Bayesian neural networks or MC-dropout as previously discussed) and approaches to obtaining SSN targets can help alleviate these issues.

Another limitation of our method is the fixed β in our confidence intervals. In safe BO, it is often assumed that β is a function of step t rather than a constant value, which is used to derive useful theoretical results such as a regret bound. In our case, a varying β can lead to instability issues due to moving targets for SSN, thus we treat β as a constant hyperparameter. However, we believe that a tunable β may have interesting theoretical and practical implications for deep safe RL as well, which we leave for future work.

Finally, in this work, we train our SSN for cost Q-value critics. However, a sizable portion of safe RL algorithms contain cost state-value critics ($\hat{V}_c$) instead. It is also possible to train an SSN for $\hat{V}_c$, by defining

$$\hat{S}_c^{\text{state}}(s') \leftarrow \min_{\substack{s \in \mathcal{S}_{\text{anc}} \\ \hat{S}_c^{\text{target}}(s) \leq h}} \text{UCB}(s) + L_{\hat{V}_c} \|s - s'\|_2^2. \tag{17}$$

However, $\mathcal{S}_{\text{anc}}$ cannot be obtained in the same way as $\mathcal{A}_{\text{anc}}$. While we assume that the agent can take any action in its action space, it cannot directly select the states it will be at in the next time step. As such, $\mathcal{S}_{\text{anc}}$ must be constructed using a dynamics model to make sure that only reachable states are selected. While this approach is moreso in the model-based territory which is out of our scope, it can be an interesting research direction.

7 Conclusion

In this work, we introduced Uncertainty-Aware Safety Propagation Critics (USPC), a model-free approach for safe reinforcement learning that constructs conservative cost surrogates by combining ensemble-based epistemic uncertainty estimation with Lipschitz-based safety propagation. By training a scalable safe set network that approximates a pessimistic cost action–value function, USPC enables safety information to generalize beyond sampled actions in continuous state–action spaces. Our proposed safe set approximation may also be used to extend some existing safe Bayesian optimization methods to continuous domains. Our method can be seamlessly integrated into existing off-policy safe actor–critic algorithms without altering their core optimization procedures. Empirical results on multiple Safety Gymnasium benchmarks demonstrate that, for most tasks, USPC reduces both the frequency and magnitude of constraint violations during training and evaluation while maintaining competitive reward performance. These results suggest that uncertainty-aware safety propagation is a practical and effective mechanism for improving robustness in model-free safe reinforcement learning under limited or unreliable cost estimates.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization, 2017. URL https://arxiv.org/abs/1705.10528.

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble, 2021. URL https://arxiv.org/abs/2110.01548.

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. URL https://openreview.net/forum?id=aKRADWBJ1I.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

Kamyar Azizzadenesheli and Animashree Anandkumar. Efficient exploration through bayesian deep q-networks, 2019. URL https://arxiv.org/abs/1802.04412.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/766ebcd59621e305170616ba3d3dac32-Paper.pdf.

Felix Berkenkamp, Andreas Krause, and Angela P Schoellig. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine learning*, 112(10):3713–3747, 2023.

Nils Bjorck, Carla P Gomes, and Kilian Q Weinberger. Towards deeper deep reinforcement learning with spectral normalization. *Advances in neural information processing systems*, 34:8242–8255, 2021.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks, 2015. URL https://arxiv.org/abs/1505.05424.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

Christopher Diehl, Timo Sebastian Sievernich, Martin Krüger, Frank Hoffmann, and Torsten Bertram. Uncertainty-aware model-based offline reinforcement learning for automated driving. *IEEE Robotics and Automation Letters*, 8(2):1167–1174, 2023. doi: 10.1109/LRA.2023.3236579.

Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.

Dirk Eilers, Simon Burton, Felippe Schmoeller da Roza, and Karsten Roscher. Safety assurance with ensemble-based uncertainty estimation and overlapping alternative predictions in reinforcement learning. In *SafeAI@ AAAI*, 2023.

Daniel Flögel, Marcos Gómez Villafañe, Joshua Ransiek, and Sören Hohmann. Disentangling uncertainty for safe social navigation using deep reinforcement learning, 2025. URL https://arxiv.org/abs/2409.10655.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

Shangding Gu, Laixi Shi, Yuhao Ding, Alois Knoll, Costas J Spanos, Adam Wierman, and Ming Jin. Enhancing efficiency of safe reinforcement learning via sample manipulation. *Advances in Neural Information Processing Systems*, 37:17247–17285, 2024.

Jonas Günster, Puze Liu, Jan Peters, and Davide Tateo. Handling long-term safety and uncertainty in safe reinforcement learning, 2024. URL https://arxiv.org/abs/2409.12045.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration, 2017. URL https://arxiv.org/abs/1605.09674.

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. URL https://openreview.net/forum?id=WZmlxIuIGR.

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*, 25(285):1–6, 2024. URL http://jmlr.org/papers/v25/23-0681.html.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21810–21823. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f7efa4f864ae9b88d43527f4b14f750f-Paper.pdf.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017. URL https://arxiv.org/abs/1612.01474.

Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning, 2021. URL https://arxiv.org/abs/2007.04938.

Yongshuai Liu, Jiaxin Ding, and Xin Liu. Ipo: Interior-point policy optimization under constraints. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 4940–4947, 2020.

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

Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, and Ding Zhao. Datasets and benchmarks for offline safe reinforcement learning. *Journal of Data-centric Machine Learning Research*, 2024.

Björn Lütjens, Michael Everett, and Jonathan P. How. Safe reinforcement learning with model uncertainty estimates. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8662–8668, 2019. doi: 10.1109/ICRA.2019.8793611.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn, 2016. URL https://arxiv.org/abs/1602.04621.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning, 2018. URL https://arxiv.org/abs/1806.03335.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Jonas Rothfuss, Bhavya Sukhija, Tobias Birchler, Parnian Kassraie, and Andreas Krause. Hallucinated adversarial control for conservative offline policy evaluation. In *Uncertainty in Artificial Intelligence*, pp. 1774–1784. PMLR, 2023.

Simon Schmitt, John Shawe-Taylor, and Hado van Hasselt. Exploration via epistemic value estimation, 2023. URL https://arxiv.org/abs/2303.04012.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017. URL https://arxiv.org/abs/1502.05477.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models, 2020. URL https://arxiv.org/abs/2005.05960.

Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. Model-based active exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5779–5788. PMLR, 09–15 Jun 2019.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Kyle Stachowicz and Sergey Levine. Racer: Epistemic risk-sensitive rl enables fast driving with fewer crashes, 2024. URL https://arxiv.org/abs/2405.04714.

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

Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 997–1005, Lille, France, 07–09 Jul 2015.

Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise safe bayesian optimization with gaussian processes. In *International conference on machine learning*, pp. 4781–4789. PMLR, 2018.

Yi Sun, Faustino Gomez, and Juergen Schmidhuber. Planning to be surprised: Optimal bayesian exploration in dynamic environments, 2011. URL https://arxiv.org/abs/1103.5708.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite markov decision processes with gaussian processes. *Advances in neural information processing systems*, 29, 2016.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

Akifumi Wachi and Yanan Sui. Safe reinforcement learning in constrained markov decision processes. In *International Conference on Machine Learning*, pp. 9797–9806. PMLR, 2020.

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

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

Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Projection-based constrained policy optimization. *arXiv preprint arXiv:2010.03152*, 2020.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 14129–14142. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/a322852ce0df73e204b7e67cbbef0d0a-Paper.pdf`.

Yiming Zhang, Quan Vuong, and Keith Ross. First order constrained optimization in policy space. *Advances in Neural Information Processing Systems*, 33:15338–15349, 2020.

Zheng Zhang, Qi Liu, Yanjie Li, Ke Lin, and Linyu Li. Safe reinforcement learning in autonomous driving with epistemic uncertainty estimation. *IEEE Transactions on Intelligent Transportation Systems*, 25(10): 13653–13666, 2024. doi: 10.1109/TITS.2024.3397700.

A Appendix

A.1 Complete USPC Algorithms

In this section, we show the USPC algorithms that were used in our experiments. Blue parts denote the changes made by USPC over the original algorithm.

A.1.1 USPC-TD3Lag

TD3Lag (Ji et al., 2024) is a primal-dual safe RL extension to TD3 (Fujimoto et al., 2018). Instead of simply training the policy to maximize rewards, TD3Lag introduces a dual variable λ then constructs a Lagrangian as the new training objective. After each policy update, λ is also updated according to episodic mean cost $\bar{C}$ and constraint threshold h. For USPC, we simply train $\hat{S}_c$ alongside the critics then replace the cost critic in the Lagrangian definition with $\hat{S}_c$, as shown in Algorithm 2. The parts modified by USPC are shown in blue.

Algorithm 2 USPC-TD3Lag training for one epoch

Require: Batch size B, particle size K, policy parameter θ_i

Sample B transitions from replay buffer
Update Q_r, Q_c via Bellman backup
Update $\hat{S}_c$ with Alg. 1
Update policy π_θ with TD3 update (Fujimoto et al., 2018) using loss $\mathcal{L}_\pi = \frac{1}{1+\lambda}\frac{1}{B}\sum_{i=1}^{B}\left(-\hat{Q}_r(s_i,a_i) + \lambda\hat{S}_c(s_i,a_i)\right)$
Update dual variable λ using loss $\mathcal{L}_\lambda = -\lambda(\bar{C} - h)$

A.1.2 USPC-CVPO

CVPO (Liu et al., 2022) is a variational safe RL algorithm that consists of an expectation step to get an optimal variational policy distribution for the constrained problem and updates the dual variables, and a maximization step that fits the parametrized policy to the variational distribution with supervised learning. Since the maximization step does not depend on cost critics given the optimal variational distribution, we substitute $\hat{S}_c$ for Q_c in the expectation step, as shown in Alg. 3. The parts modified by USPC are shown in blue.

Algorithm 3 USPC-CVPO training for one epoch

Require: Batch size B, particle size K, policy parameter θ_i

Sample B transitions from replay buffer
Update Q_r, Q_c via Bellman backup
Update $\hat{S}_c$ with Alg. 1
for $b = 1, \ldots, B$ **do** do
    Sample K actions $\{a_1, \ldots, a_K\}$ for s_b
    Compute $\{Q_r(s_b, a_k), \hat{S}_c(s_b, a_k); k = 1, ..., K\}$
end for
Compute optimal dual variables η^*, λ^* by solving the convex optimization problem (9) in Liu et al. (2022), substituting $\hat{S}_c$ in place of Q_c in the original algorithm
Compute the optimal variational distribution for each state $\{q^*(\cdot|s_b); b = 1, ..., B\}$ by (8) in Liu et al. (2022), substituting $\hat{S}_c$ in place of Q_c in the original algorithm
Update policy from π_{θ_i} to $\pi_{\theta_{i+1}}$ via supervised learning objective (12) in Liu et al. (2022)

B Hyperparameters

The values we use for the hyperparameters introduced by USPC are shown in Table **??**. For other hyperparameters, we use the values provided by OmniSafe (Ji et al., 2024) (for TD3, CPO, TD3Lag and USPC-TD3Lag) and FSRL (Liu et al., 2024) (for CVPO and USPC-CVPO).

Hyperparameter	**Value**	**Description**
M	6	Number of cost critics in the ensemble
β	2.0	Uncertainty scaling factor for UCB construction
L	2.0 (other tasks) 10.0 (Velocity)	Lipschitz constant assumed for $\hat{Q}_c$
K_{loc}	64	Number of local anchor actions per batch sample
K_{glob}	64	Number of global anchor actions per batch sample
α	0.005	EMA/Polyak update rate for the SSN target network
lr_{SSN}	3×10^{-4}	Learning rate of the safe set network
batch size	256	Batch size used for SSN updates

Table 1: USPC-specific hyperparameters.

C Complete Experiment Results

We present full experiment results in this section.

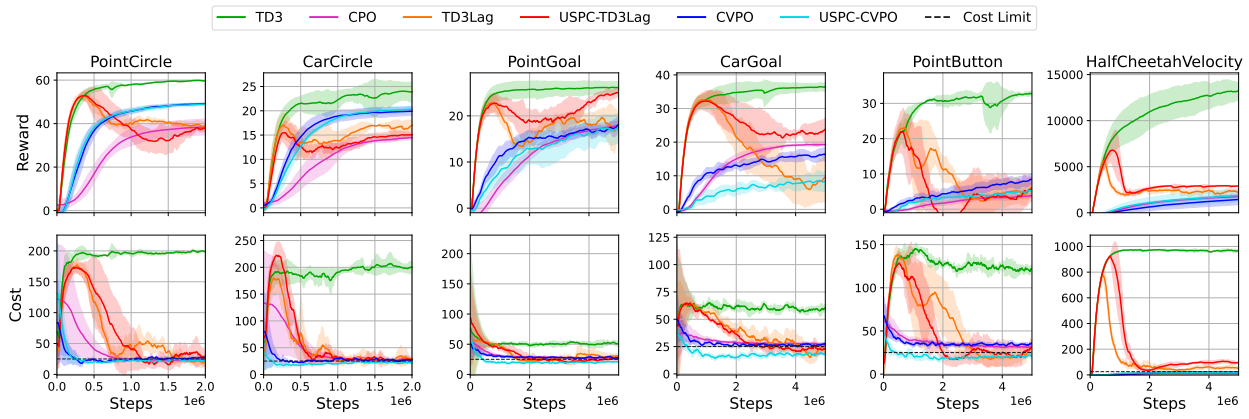


Figure 7: Reward and cost curves in training episodes, with y-axis unscaled.

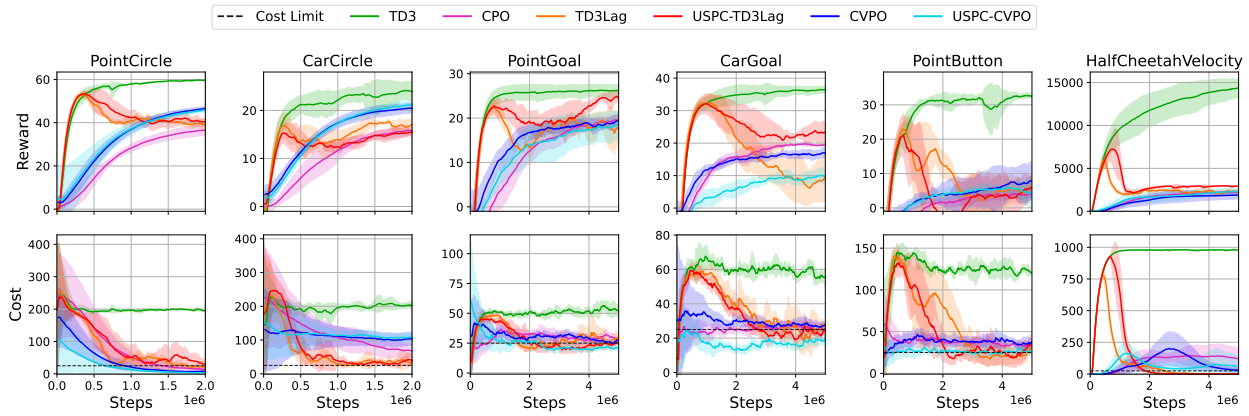


Figure 8: Reward and cost curves in evaluation episodes, with y-axis unscaled.

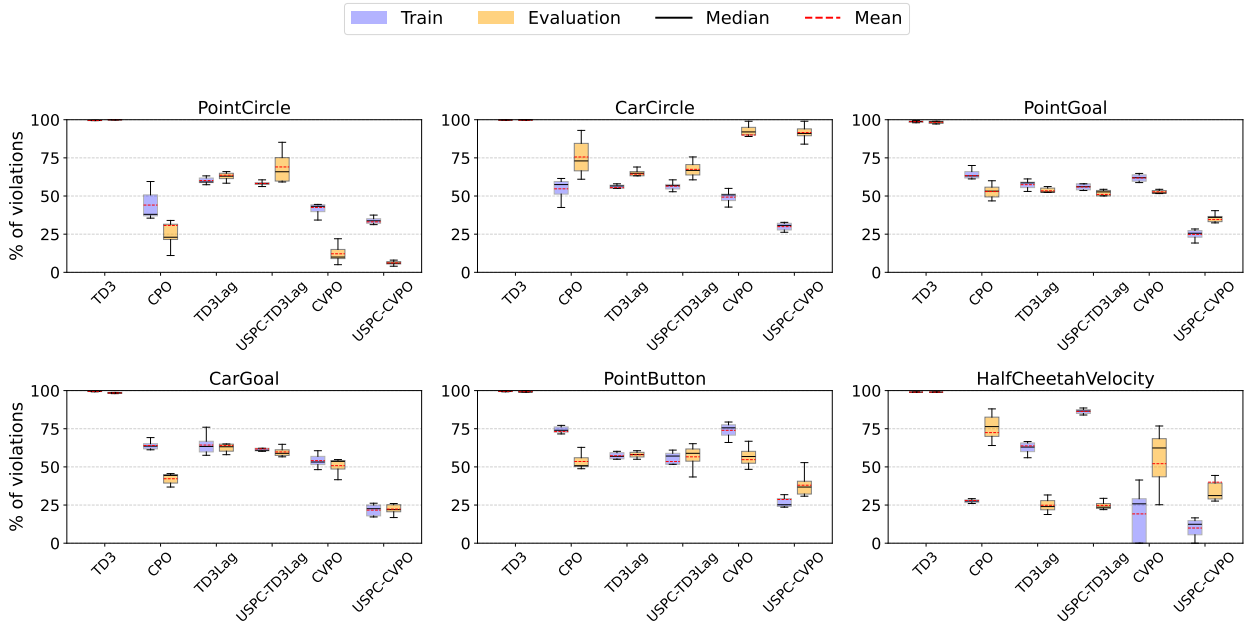


Figure 9: Constraint violation rates (including TD3), with y-axis scaled.