Feasible Policy Optimization for Safe Reinforcement Learning

Yujie Yang[1,*], **Yuanxu Sun**[2,*], **Wenyu Li**[3], **Beiyan Jiang**[1], **Tao Zhang**[4], **Shengbo Eben Li**[1,†]
[1]School of Vehicle and Mobility & College of AI, Tsinghua University
[2]School of Software, BNRist, Tsinghua University
[3]College of Artificial Intelligence, Nankai University, [4]SunRisingAI Lab
[*]Equal contribution, [†]Corresponding author: lishbo@tsinghua.edu.cn

Abstract

Policy gradient methods serve as a cornerstone of reinforcement learning (RL), yet their extension to safe RL, where policies must strictly satisfy safety constraints, remains challenging. While existing methods enforce constraints in every policy update, we demonstrate that this is unnecessarily conservative. Instead, each update only needs to progressively expand the feasible region while improving the value function. Our proposed algorithm, namely feasible policy optimization (FPO), simultaneously achieves both objectives by solving a region-wise policy optimization problem. Specifically, FPO maximizes the value function inside the feasible region and minimizes the feasibility function outside it. We prove that these two sub-problems share a common optimal solution, which is obtained based on a tight bound we derive on the constraint decay function. Extensive experiments on the Safety-Gymnasium benchmark show that FPO achieves excellent constraint satisfaction while maintaining competitive task performance, striking a favorable balance between safety and return compared to state-of-the-art safe RL algorithms.

1 Introduction

Reinforcement learning (RL) has demonstrated remarkable success in domains ranging from board games [1] and racing simulations [2] to recent breakthroughs in large language models [3]. Despite these successes, a fundamental challenge persists: current methods primarily excel in simulated environments where unsafe behaviors carry no real cost, while in safety-critical applications, policy failures could lead to severe consequences. Addressing this challenge requires considering a constrained optimal control problem, where policies must strictly satisfy safety constraints at all times, also known as state-wise constraints [4], while maximizing expected returns [5].

Policy gradient (PG) is a foundational method in RL [6], which formulates RL as an optimization problem and applies gradient-based methods to solve it. This framework has given rise to powerful modern deep RL algorithms such as proximal policy gradient (PPO) [7] and group relative policy optimization (GRPO) [8]. However, a critical limitation of standard PG methods is that they are not directly applicable to safe RL because of their unconstrained problem formulation. Despite many well-established constrained optimization techniques [9], integrating them with PG while maintaining high training efficiency remains an open challenge.

Existing safe RL methods fall into two categories. A prominent class is called iterative unconstrained RL, which reformulates safe RL as a sequence of unconstrained optimization problems, typically via the method of Lagrange multipliers, and solves them using standard RL algorithms [10]. While theoretically sound, these methods suffer from slow convergence and training instability. Slow convergence arises from the need to solve an RL problem in each iteration, resulting in convergence rates approximately an order of magnitude slower than standard RL algorithms. Training instability

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

stems from the characteristic of the Lagrange multiplier, manifesting as persistent oscillations in return and constraint violation throughout training [11].

Another class of methods, called constrained policy optimization, aligns more closely with PG, or more generally, policy optimization, which employs more advanced optimization techniques than pure gradient ascent. These methods impose the safety constraint on the sub-problem in each iteration, requiring every intermediate policy to be strictly safe. While more efficient than unconstrained iterative RL, these methods suffer from the infeasibility issue: they often fail to find a constraint-satisfying solution to the sub-problems, especially during early training stages. This is because the constraint is too stringent for policies that have not been sufficiently trained after random initialization. In such cases, these methods must resort to pure constraint minimization without reward optimization [12], resulting in overly conservative updates and inefficient training.

In this paper, we challenge the conventional practice of enforcing the original constraint in every iteration of policy optimization. Instead, we demonstrate that this stringent constraint can be replaced by a much milder one: each iteration only needs to produce a policy that is safer than the previous iteration in the sense that its feasible region is larger. Building on this foundation, we propose feasible policy optimization (FPO), which features a region-wise policy update rule: maximizing the value function inside the feasible region and minimizing the feasibility function outside it. We prove that these two objectives, originally expressed by two separate optimization problems, can be simultaneously achieved with a shared optimal solution. We further derive a tight bound on the constraint decay function (CDF), enabling more accurate feasible region estimation compared to the conventional cost value function (CVF). Extensive evaluation on the Safety-Gymnasium benchmark demonstrates FPO's excellent balance between safety and return.

2 Related work

Iterative unconstrained RL Most iterative unconstrained RL methods use the method of Lagrange multipliers and solve the dual problem using dual ascent, where the minimization step solves an unconstrained RL problem [10]. For example, Chow et al. [13] constrain the conditional value-at-risk of the CVF in a constrained Markov decision process [14], forming a probabilistic constraint. Tessler et al. [15] incorporate the cost signal into the reward function, treating the integrated discounted sum as a new value function. The Lagrange multiplier framework is also adaptable to other kinds of feasibility functions, including Hamilton-Jacobi reachability [16, 17], control barrier function [18, 19], and safety index [20]. As a special case, when the multiplier is fixed as a constant, the algorithm reduces to a penalty function method [21].

Constrained policy optimization The most representative example of this class is the constrained policy optimization (CPO) algorithm [12], which builds on the trust region policy optimization (TRPO) [22] and further adds a linearized safety constraint. To avoid the computationally expensive line search in CPO, Yang et al. [23] propose to first perform a reward improvement update and then project the policy back onto the constrained set. Zhang et al. [24] propose to first solve for the optimal policy in a non-parameterized policy space and then project it back into the parametric space. Following the projection method, Yang et al. [25] propose generalized advantage estimation (GAE) for the surrogate function to further improve performance. Inspired by techniques from constrained optimization, the interior-point method [26] and the augmented Lagrange method [27] are also explored to solve the policy optimization problem in each iteration. For finite-horizon problems, Zhao et al. [28, 29] convert state-wise constraints to cumulative constraints through cost reconstruction and bound the worst-case violation.

3 Preliminaries

3.1 Problem statement

Safe RL addresses control problems in which an agent aims to maximize long-term rewards while strictly adhering to safety constraints at every step. We consider a Markov decision process (MDP) $(\mathcal{X}, \mathcal{U}, d_{\text{init}}, P, r, \gamma)$, where $\mathcal{X} \subseteq \mathbb{R}^n$ is the state space, $\mathcal{U} \subseteq \mathbb{R}^m$ is the action space, $d_{\text{init}} \in \Delta\mathcal{X}$ is the initial state distribution, $P : \mathcal{X} \times \mathcal{U} \to \Delta\mathcal{X}$ is the transition probability, $r : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is the reward function, and $0 < \gamma < 1$ is the discount factor. We consider a stochastic policy $\pi : \mathcal{X} \to \Delta\mathcal{U}$,

whose value function is defined as:

$$V^{\pi}(x) = \mathbb{E}_{x_{t+1}\sim P(\cdot|x_t,u_t),u_t\sim\pi(\cdot|x_t)}\left[\sum_{t=0}^{\infty}\gamma^t r(x_t,u_t)\Big|x_0=x\right]. \tag{1}$$

Safety is specified through a state constraint expressed as an inequality $h(x) \leq 0$, where $h : \mathcal{X} \to \mathbb{R}$ is the constraint function. We aim to find a policy that maximizes the expected value function while satisfying the state constraint at every step over an infinite horizon:

$$\begin{aligned} &\max_{\pi} \mathbb{E}_{x\sim d_{\text{init}}}[V^{\pi}(x)] \\ &\text{s.t. } h(x_t) \leq 0, \forall t \geq 0, x_0 \in \mathcal{X}_{\text{init}}, \end{aligned} \tag{2}$$

where $\mathcal{X}_{\text{init}} = \{x \in \mathcal{X} | d_{\text{init}}(x) > 0\}$ is the support of the initial state distribution.

3.2 Feasible region and feasibility function

The constrained optimal control problem (2) is intractable because it has infinitely many constraints. A common solution is to aggregate these constraints into a single one through a feasibility function. To formally describe the concept of feasibility, we first define the reachable set.

Definition 1 (Reachable set)**.** *The reachable set of a policy π from a state $x \in \mathcal{X}$, denoted $\mathcal{R}^{\pi}(x)$, is the set of states that can be reached with non-zero probability under π in finite time:*

$$\mathcal{R}^{\pi}(x) = \{x' \in \mathcal{X} | \exists t \geq 0, \text{ s.t. } P(x_t = x' | x, \pi) > 0\}, \tag{3}$$

where $P(x_t = x'|x, \pi)$ is the probability of reaching x' at time t starting from x and following π.

We call a state feasible under a policy if all its future states satisfy the safety constraint, and the set of all feasible states under a policy is the feasible region of the policy.

Definition 2 (Feasible region)**.** *The feasible region of a policy π, denoted X^{π}, is the set of states from which every reachable set under π satisfies the safety constraint:*

$$\mathrm{X}^{\pi} = \{x \in \mathcal{X} | \forall x' \in \mathcal{R}^{\pi}(x), h(x') \leq 0\}. \tag{4}$$

The feasible region enables us to describe the long-term safety requirement compactly: the feasible region must include all possible initial states. This requirement can be expressed as a single constraint by the feasibility function.

Definition 3 (Feasibility function)**.** *Function $F^{\pi} : \mathcal{X} \to \mathbb{R}$ is a feasibility function of π if and only if its zero-sublevel set equals the feasible region of π, i.e., $\{x \in \mathcal{X} | F^{\pi}(x) \leq 0\} = \mathrm{X}^{\pi}$.*

An example of a feasibility function is the CDF [19].

Definition 4 (Constraint decay function)**.** *The CDF of a policy π is defined as*

$$F^{\pi}(x) = \mathbb{E}_{\tau\sim\pi}\left[\gamma^{N(\tau)}\big|x_0 = x\right], \tag{5}$$

where $\gamma \in (0, 1)$ is the discount factor, $\tau = \{x_0, u_0, x_1, u_1, \dots\}$ is a trajectory sampled by π, and $N(\tau) \in \mathbb{N}$ is the time step of the first constraint violation in τ.

The CDF is non-negative by definition, and thus its zero-sublevel set equals its zero-level set. Without loss of generality, we only consider non-negative feasibility functions in this paper. For feasibility functions with negative values, we can take their non-negative parts $F^{\pi}_{+} = \max\{F^{\pi}, 0\}$ without changing the feasible region. With a feasibility function, we can aggregate the infinitely many constraints in Problem (2) into a single one, obtaining the following problem:

$$\max_{\pi} \mathbb{E}_{x\sim d_{\text{init}}}[V^{\pi}(x)] \quad \text{s.t. } \mathbb{E}_{x\sim d_{\text{init}}}[F^{\pi}(x)] \leq 0. \tag{6}$$

4 Methods

Existing constrained policy optimization methods typically require that every intermediate policy satisfies the constraint in Problem (6). Instead, our algorithm only requires each policy to have a larger feasible region than the previous policy, which can be achieved through a region-wise policy optimization scheme.

4.1 Region-wise policy optimization

The core of our approach is to solve two optimization problems in each iteration. Let π_k denote the policy from the previous iteration. The first problem is to maximize the value function inside the feasible region under the constraint that the new feasible region is not smaller than the previous one:

$$\begin{aligned} &\max_{\pi} \mathbb{E}_{x\sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x) \leq 0]V^{\pi}(x)] \\ &\text{s.t. } \mathbb{E}_{x\sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x) \leq 0]F^{\pi}(x)] \leq 0. \end{aligned} \tag{7}$$

The second problem is to minimize the feasibility function outside the feasible region under the same constraint:

$$\begin{aligned} &\min_{\pi} \mathbb{E}_{x\sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x) > 0]F^{\pi}(x)] \\ &\text{s.t. } \mathbb{E}_{x\sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x) \leq 0]F^{\pi}(x)] \leq 0. \end{aligned} \tag{8}$$

This mechanism is intended to produce a sequence of monotonically improving value functions alongside expanding feasible regions, ultimately leading to the optimal solution of the original safe RL problem (6). While this principle draws inspiration from classical state-wise optimization schemes [30], we significantly extend it to expectation-based optimization that is compatible with deep RL. Crucially, while Problems (7) and (8) separately address optimality and feasibility, we prove that they admit a shared optimal solution, allowing us to combine the two seemingly divergent requirements into a single, unified policy update.

Theorem 1. *There exists a policy π_{k+1} that is the optimal solution to both Problem (7) and (8).*

Proof Sketch. Let π_{in} and π_{out} denote the optimal solutions to Problem (7) and (8), respectively. We construct the following policy:

$$\pi_{k+1}(\cdot|x) = \begin{cases} \pi_{\text{in}}(\cdot|x), & x \in \mathcal{R}^{\pi_{\text{in}}}(\mathcal{X}_{\text{init}} \cap \mathrm{X}^{\pi_k}), \\ \pi_{\text{out}}(\cdot|x), & \text{otherwise}, \end{cases} \tag{9}$$

where $\mathcal{R}^{\pi}(X) = \bigcup_{x\in X} \mathcal{R}^{\pi}(x)$ denotes the reachable set of π from a set of states $X \subseteq \mathcal{X}$. We prove that π_{k+1} is the optimal solution to both problems. The key is to observe that $\mathcal{R}^{\pi_{\text{in}}}(\mathcal{X}_{\text{init}} \cap \mathrm{X}^{\pi_k})$ is forward invariant under π_{k+1}. See Appendix A.1 for the complete proof. □

Note that Equation (9) only provides one valid choice of π_{k+1}. There may exist other valid policies, such as remaining with π_k in the overlapping part of the reachable sets. Theorem (1) allows us to merge Problem (7) and (8) into a single problem as follows:

$$\begin{aligned} &\max_{\pi} \mathbb{E}_{x\sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x) \leq 0]V^{\pi}(x) - \mathbb{I}[F^{\pi_k}(x) > 0]F^{\pi}(x)] \\ &\text{s.t. } \mathbb{E}_{x\sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x) \leq 0]F^{\pi}(x)] \leq 0. \end{aligned} \tag{10}$$

Corollary 1. *The optimal solution to Problem (10) is also the optimal solution to both Problem (7) and (8).*

This is because the objective function of Problem (10) is the sum of the objective functions of Problem (7) and (8), and they share the same constraint. Thus, π_{k+1} defined in (9) is the optimal solution to all three problems.

4.2 Feasibility function bounds

A difficulty of solving Problem (10) is that the value function and feasibility function of the new policy π cannot be directly approximated with samples collected by the old policy π_k. To solve this problem, we replace the two functions with their lower and upper bounds, which can be approximated by samples from the old policy. Achiam et al. [12] derive the bounds for functions in the form of discounted summation, which is applicable to the value function. In this section, we move a step further and derive the bounds for CDF.

We begin with a decomposition of state distribution. Given an initial state $x \in \mathcal{X}$, the discounted future state distribution under policy π is

$$d^{\pi}(x'|x) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t P(x_t = x'|x, \pi).$$

By law of total probability, we decompose each term in the summation based on whether the constraint has been violated up to that step:

$$P(x_t = x'|x, \pi) = P(x_t = x', \max_{s<t} c_s = 0|x, \pi) + P(x_t = x', \max_{s<t} c_s = 1|x, \pi),$$

where $c(x) = \mathbb{I}[h(x) > 0]$ is the binary indicator of an instantaneous constraint violation and $c_s = c(x_s)$ denotes its value at step s. Then, the future state distribution can be decomposed as $d^\pi(x'|x) = d_0^\pi(x'|x) + d_+^\pi(x'|x)$, where

$$d_0^\pi(x'|x) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t P(x_t = x', \max_{s<t} c_s = 0|x, \pi),$$
$$d_+^\pi(x'|x) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t P(x_t = x', \max_{s<t} c_s = 1|x, \pi).$$

We call d_0^π the prefix state distribution. This decomposition is critical in deriving the bounds for CDF. As we will show later, the bounds for CDF only depend on the prefix state distribution because states beyond the first violation are irrelevant to the CDF.

Theorem 2. *For any policies $\tilde{\pi}$ and π, and any state $x \in \mathcal{X}$, define*

$$A_F^\pi(x, u) = \mathbb{E}_{x' \sim P(\cdot|x,u)}[c(x) + (1 - c(x))\gamma F^\pi(x') - F^\pi(x)],$$

and $L_{\tilde{\pi}}^\pi(x) = \mathbb{E}_{x' \sim d_0^\pi(\cdot|x), u' \sim \tilde{\pi}(\cdot|x')}[A_F^\pi(x', u')]$, $\epsilon_F^{\tilde{\pi}} = \max_x |\mathbb{E}_{u \sim \tilde{\pi}(\cdot|x)}[A_F^\pi(x, u)]|$. Then,

$$F^{\tilde{\pi}}(x) - F^\pi(x) \geq \frac{L_{\tilde{\pi}}^\pi(x)}{1-\gamma} - \frac{2\gamma\epsilon_F^{\tilde{\pi}}}{(1-\gamma)^2}\mathbb{E}_{x' \sim d_0^\pi(\cdot|x)}[D_{TV}(\tilde{\pi}\|\pi)[x']],$$
$$F^{\tilde{\pi}}(x) - F^\pi(x) \leq \frac{L_{\tilde{\pi}}^\pi(x)}{1-\gamma} + \frac{2\gamma\epsilon_F^{\tilde{\pi}}}{(1-\gamma)^2}\mathbb{E}_{x' \sim d_0^\pi(\cdot|x)}[D_{TV}(\tilde{\pi}\|\pi)[x']],$$

where $D_{TV}(\tilde{\pi}\|\pi)[x'] = (1/2)\sum_u |\tilde{\pi}(u|x') - \pi(u|x')|$ is the total variational divergence between action distributions at x'. Furthermore, the bounds are tight (when $\tilde{\pi} = \pi$, the LHS and RHS are identically zero).

Here, $A_F^\pi(x, u)$ measures whether taking action u increases or decreases the predicted first-violation risk relative to $F^\pi(x)$; a positive value indicates reduced feasibility. The surrogate $L_{\tilde{\pi}}^\pi(x)$ accumulates this change over the pre-violation portion of trajectories, since states after the first violation do not affect the CDF. The constant $\epsilon_F^{\tilde{\pi}}$ uniformly bounds the expected feasibility-advantage magnitude and controls the higher-order error caused by a policy update. Finally, $D_{TV}(\tilde{\pi}\|\pi)[x]$ measures the update size between the two action distributions at state x.

Proof Sketch. We construct an auxiliary MDP $\tilde{M}$, which is identical to M except for its transition probability. In $\tilde{M}$, once the constraint is violated, the state is fixed at the one that violates the constraint for all future steps. We prove that the CDF and prefix state distribution are identical in M and $\tilde{M}$, and the result to prove holds in $\tilde{M}$. Therefore, the result also holds in M. See Appendix A.2 for the complete proof. □

One may ask why not use the CVF as the feasibility function, which is a discounted summation so that the bound from CPO would still apply. The reason is that CDF yields more accurate estimates than CVF in practice. In safe RL, feasibility functions are typically estimated using bootstrapping methods like TD(λ), which suffer from approximation bias of the feasibility function itself. While this bias affects both CDF and CVF, CVF suffers more severely because it is unbound and requires infinite-horizon trajectories. In contrast, CDF is bounded within $[0, 1]$, allowing the bootstrapping target to be explicitly clipped, and its shorter trajectories (truncated at first violation) also decrease variance.

4.3 Feasible policy optimization

With the CDF bounds, we are ready to solve Problem (10). Substituting the upper bound of CDF from Theorem 2 and the lower bound of value function from Corollary 1 in the CPO paper [12], and

following the practice of trust region methods, we obtain the following problem:

$$\begin{aligned}
&\max_{\pi} \mathbb{E}_{x\sim d^{\pi_k},u\sim\pi}\left[\mathbb{I}[F^{\pi_k}(x)\leq 0]A^{\pi_k}(x,u)\right] - \mathbb{E}_{x\sim d_0^{\pi_k},u\sim\pi}\left[\mathbb{I}[F^{\pi_k}(x)> 0]A_F^{\pi_k}(x,u)\right]\\
&\text{s.t. } \mathbb{E}_{x\sim d^{\pi_k},u\sim\pi}\left[\mathbb{I}[F^{\pi_k}(x)\leq 0](F^{\pi_k}(x)+A_F^{\pi_k}(x,u)/(1-\gamma))_+\right]\leq 0\\
&\quad \mathbb{E}_{x\sim d^{\pi_k}}\left[\mathbb{I}[F^{\pi_k}(x)\leq 0]D_{KL}(\pi\|\pi_k)[x]\right]\leq \delta/2\\
&\quad \mathbb{E}_{x\sim d^{\pi_k}}\left[\mathbb{I}[F^{\pi_k}(x)> 0]D_{KL}(\pi\|\pi_k)[x]\right]\leq \delta/2.
\end{aligned} \tag{11}$$

Here, $A^{\pi_k}(x,u) = Q^{\pi_k}(x,u) - V^{\pi_k}(x)$ is the standard advantage function in RL. In the three constraints above, we replace the prefix state distribution d_0^π with the full state distribution d^π. This replacement is valid because $d^\pi \geq d_0^\pi$ pointwise and the constraint integrands are non-negative; it therefore strengthens the inequalities rather than weakening them. Our algorithm, called feasible policy optimization (FPO), iteratively solves Problem (11) to update the policy. This update rule provides the following guarantees on the safety and performance of the new policy.

Corollary 2. *The optimal solution to Problem (11), denoted π_{k+1}, satisfies the following two properties:*

1. (Feasibility enhancement) Let $\epsilon_F^{\pi_{k+1}} = \max_x |\mathbb{E}_{u\sim\pi_{k+1}(\cdot|x)}[A_F^\pi(x,u)]|$, we have

$$\mathbb{E}_{x\sim d_{init}}\left[\mathbb{I}[F^{\pi_k}(x)\leq 0]F^{\pi_{k+1}}(x)\right] \leq \frac{\sqrt{\delta}\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}, \tag{12a}$$

$$\mathbb{E}_{x\sim d_{init}}\left[\mathbb{I}[F^{\pi_k}(x)> 0]\left(F^{\pi_{k+1}}(x)-F^{\pi_k}(x)\right)\right] \leq \frac{\sqrt{\delta}\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}. \tag{12b}$$

2. (Value improvement) Let $\epsilon^{\pi_{k+1}} = \max_x |\mathbb{E}_{u\sim\pi_{k+1}(\cdot|x)}[A^\pi(x,u)]|$, we have

$$\mathbb{E}_{x\sim d_{init}}\left[\mathbb{I}[F^{\pi_k}(x)\leq 0]\left(V^{\pi_{k+1}}(x)-V^{\pi_k}(x)\right)\right] \geq -\frac{\sqrt{\delta}\gamma\epsilon^{\pi_{k+1}}}{(1-\gamma)^2}. \tag{13}$$

Proof Sketch. Split (11) into two problems similar to Section 4.1 and prove that they share the same optimal solution. The rest follows by Theorem 2 and Corollary 1 in the CPO paper [12]. See Appendix A.3 for the complete proof. □

This corollary tells us that the safety and performance degradation of the new policy is controlled. Specifically, its feasibility function will not exceed zero too much inside the feasible region or increase too much outside the feasible region, and its value function will not decrease too much inside the feasible region. As the step size δ decreases, the policy sequence obtained by FPO approaches a monotonically improving sequence in both safety and performance.

4.4 Practical implementation

We adopt the method from PPO to solve Problem (11), which applies a first-order method with the KL divergence constraints replaced by a clipped importance sampling (IS) ratio. FPO learns a feasibility network F_ϕ, a value network V_ω, and a policy network π_θ, where ϕ, ω, and θ denote their parameters. In theory, the feasible region is defined as the zero-level set of the CDF. However, in practice, using neural networks to identify this exact boundary is challenging. Since the learning target for $F_\phi(x)$ is inherently non-negative, the estimated CDF is often strictly positive almost everywhere, making the zero-level set numerically unstable. To address this, we follow a common practice in safe RL by approximating the feasible region with an ϵ-sublevel set: $\hat{\mathcal{X}}^\pi = \{x\in\mathcal{X} \mid F_\phi(x)\leq\epsilon\}$, where $\epsilon > 0$ is a small tolerance threshold. This approximation is theoretically justified under the assumption that the step to violation is uniformly bounded [21]. Empirically, we find that a fixed value of $\epsilon = 0.1$ works robustly across all evaluated environments. A comprehensive sensitivity analysis of ϵ is provided in Appendix C.4.

We deal with constraints inside the feasible region by penalizing the reward advantage $A(x,u)$ with the feasibility advantage $A_F(x,u)$:

$$\bar{A}(x,u) = \mathbb{I}[F_\phi(x)\leq\epsilon](A(x,u) - w(x)A_F(x,u)) - \mathbb{I}[F_\phi(x)>\epsilon]w_{\max}A_F(x,u).$$

The state-dependent weight is defined as a barrier-like function $w(x) = C(1/(\alpha - r(x)) - 1/\alpha)$, where $r(x) = \min(F_\phi(x)/\epsilon, 1)$ is the feasibility ratio, $\alpha > 1$ is a fixed margin parameter, and the scaling factor $C = w_{\max}\alpha(\alpha - 1)$ ensures the weight is bounded by $w_{\max}$. The weight acts as a penalty coefficient for the region constraint: states near the estimated boundary receive larger weights because a policy update is more likely to move them outside the feasible region, while the smooth form avoids a discontinuity across the state space. In practice, ϵ should be small but above the estimator's noise floor, $w_{\max}$ should be large enough to enforce feasibility without stalling return learning, and α should be slightly larger than one to provide a smooth boundary margin. Appendix C.5 compares this design with a linear alternative. To compute the feasibility advantage, we extend the GAE of the value function to the CDF (see Appendix B.1):

$$A_F(x,u) = \sum_{t=0}^{\infty}(\lambda\gamma)^t \prod_{s=0}^{t-1}(1-c_s)\left(c_t + (1-c_t)\gamma F_\phi(x_{t+1}) - F_\phi(x_t)\right). \tag{14}$$

The loss function for the feasibility network is

$$L_F(\phi) = \mathbb{E}\left[\left(F_\phi(x) - (F_{\phi_k}(x) + A_F(x,u))\right)^2\right]. \tag{15}$$

The loss function for the value network is

$$L_V(\omega) = \mathbb{E}\left[\left(V_\omega(x) - (V_{\omega_k}(x) + A(x,u))\right)^2\right]. \tag{16}$$

The loss function for the policy network is

$$L_\pi(\theta) = -\mathbb{E}\left[\min\left\{\frac{\pi_\theta(u|x)}{\pi_{\theta_k}(u|x)}\bar{A}(x,u), \text{clip}\left(\frac{\pi_\theta(u|x)}{\pi_{\theta_k}(u|x)}, 1-\xi, 1+\xi\right)\bar{A}(x,u)\right\}\right], \tag{17}$$

where $\xi > 0$ is a constant for clipping the IS ratio. In the policy-loss objective, we also replace d_0^π with d^π as a practical approximation rather than an exact equality between the distributions. Using d_0^π would require discarding every sample after the first violation; retaining these samples substantially improves sample efficiency. The pseudocode of FPO is in Appendix B.2.

5 Experiments

We aim to answer the following questions through our experiments:

Q1 How does FPO perform in terms of safety and return compared to existing algorithms?

Q2 Does FPO maintain monotonic expansion of the feasible region throughout training?

Q3 What specific behaviors does the agent learn with FPO that enable it to achieve both safety and high performance?

5.1 Experiment Setups

Environments Our experiments cover 14 environments in the Safety-Gymnasium benchmark [31], including navigation and locomotion. The navigation environments include two robots, i.e., Point and Car, and four tasks, i.e., Goal, Push, Button, and Circle, with all difficulty levels set as 1 and constraints set as default. The locomotion environments include six classic robots from Gymnasium's MuJoCo environments, i.e., HalfCheetah, Hopper, Swimmer, Walker2d, Ant, and Humanoid, with maximum velocity constraints.

Baselines We compare FPO with a wide variety of mainstream safe RL algorithms implemented in the Omnisafe toolbox [32], including iterative unconstrained RL methods RCPO [15], PPO-Lag [33], and TRPO-PID [11], and constrained policy optimization methods CPO [12], PCPO [23], FOCOPS [24], and P3O [34]. Hyperparameters for all algorithms are detailed in Appendix C.1. We use the default hyperparameters in Omnisafe for all baselines except that we set the cost limit to zero for all algorithms. The default values of other hyperparameters have been tuned for good performance as stated by Ji et al. [32].

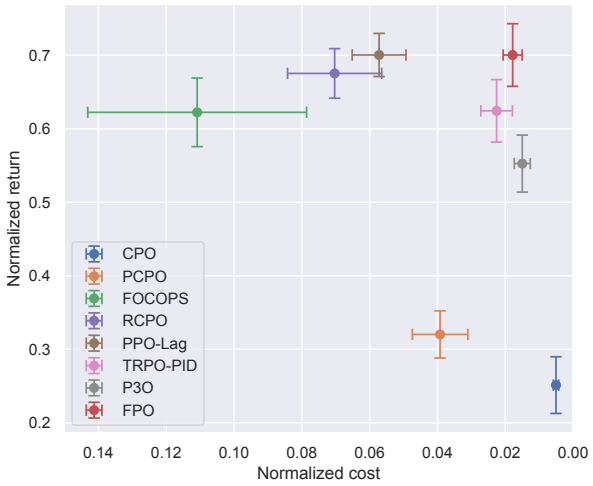


Figure 1: Normalized cost and return averaged over 14 Safety-Gymnasium environments. Points are means, and error bars show 95% confidence intervals.

5.2 Experiment results

Cost-return evaluation In safe RL, we evaluate algorithms by two metrics: (1) episode cost, representing the average number of constraint-violating steps per episode, and (2) episode return, representing the average cumulative rewards per episode. To perform a comprehensive evaluation, we place the scores of all algorithms in a cost-return plot in Figure 1. The scores are first normalized by those of PPO and then averaged on all 14 environments. The results demonstrate FPO's excellent performance in balancing safety and return: it reduces violation to less than 2% of PPO's level while maintaining 70% of its return. In contrast, other algorithms exhibit less favorable trade-offs. CPO and PCPO significantly sacrifice return due to their strict requirements on constraint satisfaction in every iteration. Lagrangian and penalty-based methods (PPO-Lag, RCPO, TRPO-PID, and P3O) explicitly trade off cost and return by adjusting penalty coefficients, forming a Pareto front. Among these, TRPO-PID adaptively controls the Lagrange multiplier to achieve a more balanced average performance, while FPO occupies a more favorable point in the normalized cost-return plot. These results answer **Q1**.

Training curves Figure 2 shows the training curves of all algorithms across eight environments. Training curves on all 14 environments, along with final cost and return scores, are provided in Appendix C.2 and C.3. FPO achieves a favorable cost-return trade-off in most environments, although it does not uniformly dominate every baseline. Notably, FPO is the only algorithm that finds a high-return and safe policy in SwimmerVelocity, while all other algorithms fail to solve this task. Constrained optimization methods like CPO and PCPO are overly conservative in most environments. Lagrangian-based methods like RCPO and PPO-Lag exhibit severe oscillations during training in several environments. HalfCheetahVelocity is a clear weaker case for FPO: its complex multi-joint dynamics are strongly coupled with the feasible-region boundary, making the first-violation CDF more demanding to estimate and FPO more conservative under the fixed interaction budget. These results provide further empirical evidence to answer **Q1**.

Feasible region visualization We visualize the feasible regions learned by FPO during training in Figure 3(a) to check whether they are monotonically expanding as required by the constraint of our algorithm. While the training lasts 500 epochs, we find that the feasible regions after 100 epochs remain almost the same. The red circles in the figure are where the hazards are located. By epoch 5, FPO demonstrates preliminary capability to identify unsafe areas, but no state is identified as feasible. With continued learning, the feasible region emerges and gradually expands. By epoch 50, FPO already achieves complete distinguishability between feasible and infeasible regions. These results demonstrate that the monotonic expansion constraint of the feasible region is satisfied throughout training, answering **Q2**. The same behavior is observed in the more challenging CarGoal environment with a different robot, as shown in Figure 3(b): the learned contours progressively tighten around the

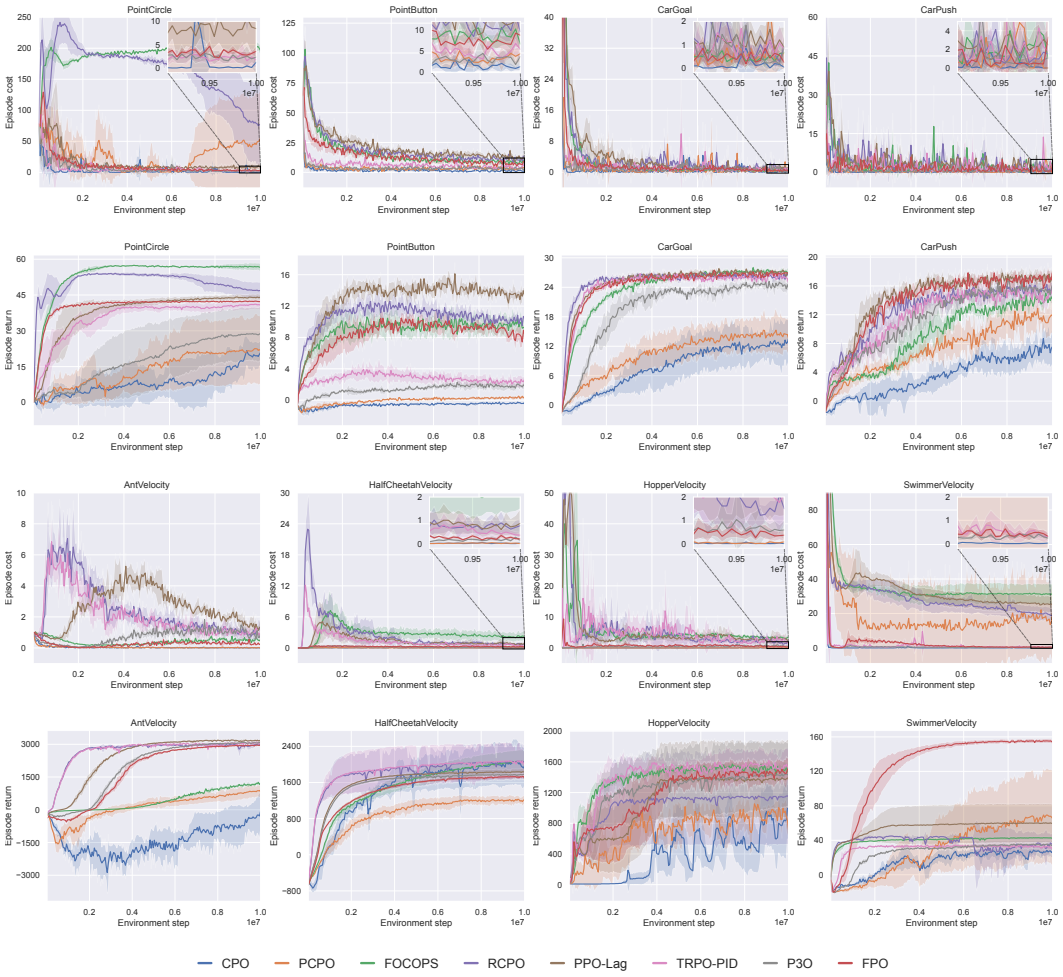
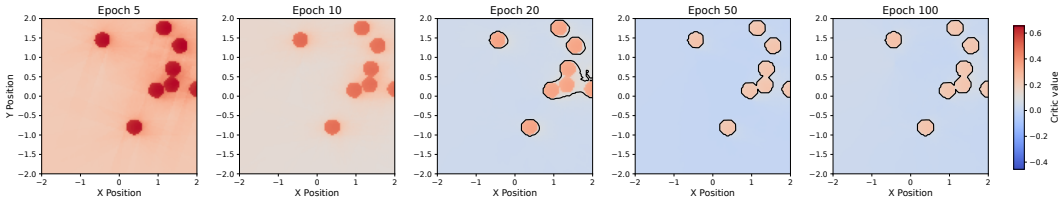
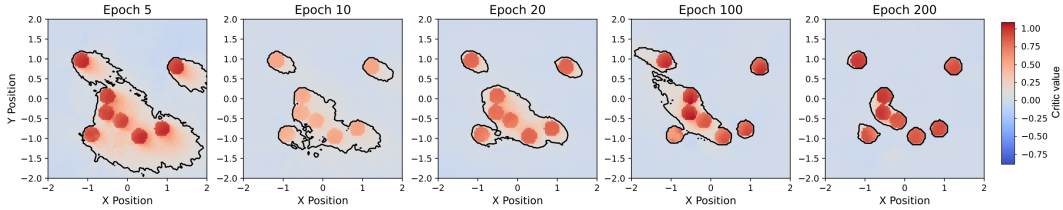


Figure 2: Training curves on eight Safety-Gymnasium environments. Curves show the mean over five seeds, and shaded areas show 95% confidence intervals.



(a) PointGoal

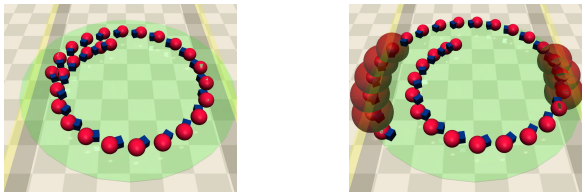


(b) CarGoal

Figure 3: Visualization of the feasible regions during training in (a) PointGoal and (b) CarGoal. The colors represent the CDF values computed by placing the agent on every point of a grid covering the space, and the contours of the 0.1-level sets are marked in black.

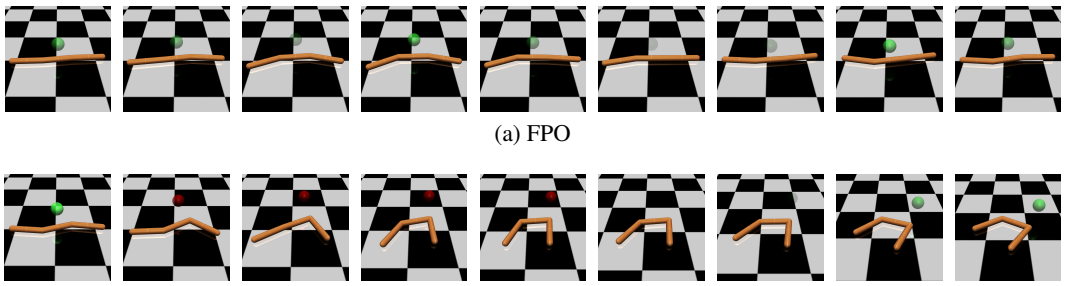
hazards and the feasible region keeps expanding. By quickly acquiring representations of the feasible region, FPO effectively focuses exploration within safe boundaries while optimizing returns.

Trajectory visualization We inspect representative policy behaviors in PointCircle and SwimmerVelocity. In Figure 4, FPO remains within the constraint boundary while PPO-Lag leaves it. In Figure 5, FPO moves within the speed limit while PPO-Lag crosses a violating pose and becomes trapped in a safe, low-return pose. These illustrative, task-dependent cases show how CDF's focus on the first violation can avoid such local optima, answering **Q3**.



(a) FPO (b) PPO-Lag

Figure 4: Trajectories of agents trained with (a) FPO and (b) PPO-Lag in PointCircle.



(a) FPO

(b) PPO-Lag

Figure 5: Trajectories of agents trained with (a) FPO and (b) PPO-Lag in SwimmerVelocity.

6 Conclusion

This paper points out that the conventional practice of enforcing the original constraint in each iteration in safe RL is unnecessarily conservative. Instead, each update only needs to find an expanded feasible region and an improved value function. We propose an algorithm called FPO that achieves both objectives by simultaneously maximizing the value function inside the feasible region and minimizing the feasibility function outside it. We prove that these two optimization problems have a shared optimal solution, supported by a tight bound we derive on the CDF, which extends the result from CPO. Extensive experiments on Safety-Gymnasium show that FPO strikes a favorable balance between safety and return compared with state-of-the-art safe RL algorithms.

A limitation of FPO is that it must learn a first-violation CDF boundary, which can require more samples than estimating a cumulative cost value in environments whose feasible region is strongly coupled with complex dynamics, as observed in HalfCheetahVelocity. Its practical implementation also relies on a handcrafted state-dependent weight, although Appendix C.5 supports the chosen form, and it may struggle when safe states are rarely encountered during initial random exploration. Replacing the handcrafted weight and improving safe exploration and CDF sample efficiency are important directions for future work.

Acknowledgements and Disclosures

This work was supported by the Beijing Natural Science Foundation (L257002) and the Shanghai Automotive Industry Science and Technology Development Foundation. The authors declare no competing interests.

References

[1] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[2] Peter R Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, et al. Outracing champion gran turismo drivers with deep reinforcement learning. *Nature*, 602(7896):223–228, 2022.

[3] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

[4] Weiye Zhao, Tairan He, Rui Chen, Tianhao Wei, and Changliu Liu. State-wise safe reinforcement learning: A survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 6814–6822, 8 2023. Survey Track.

[5] Yujie Yang, Zhilong Zheng, Masayoshi Tomizuka, Changliu Liu, and Shengbo Eben Li. The feasibility theory of constrained reinforcement learning: a tutorial study. *Foundations and Trends in Systems and Control*, 13(1):1–72, 2026.

[6] Shengbo Eben Li. *Reinforcement learning for sequential decision and optimal control*. Springer Verlag, Singapore, 2023.

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

[8] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[9] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[10] Santiago Paternain, Luiz Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. *Advances in Neural Information Processing Systems*, 32, 2019.

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

[12] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

[13] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(167):1–51, 2018.

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

[15] Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019.

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

[17] Dongjie Yu, Wenjun Zou, Yujie Yang, Haitong Ma, Shengbo Eben Li, Yuming Yin, Jianyu Chen, and Jingliang Duan. Safe model-based reinforcement learning with an uncertainty-aware reachability certificate. *IEEE Transactions on Automation Science and Engineering*, 21(3):4129–4142, 2023.

[18] Yujie Yang, Yuxuan Jiang, Yichen Liu, Jianyu Chen, and Shengbo Eben Li. Model-free safe reinforcement learning through neural barrier certificate. *IEEE Robotics and Automation Letters*, 8(3):1295–1302, 2023.

[19] Yujie Yang, Yuhang Zhang, Wenjun Zou, Jianyu Chen, Yuming Yin, and Shengbo Eben Li. Synthesizing control barrier functions with feasible region iteration for safe reinforcement learning. *IEEE Transactions on Automatic Control*, 69(4):2713–2720, 2023.

[20] Haitong Ma, Changliu Liu, Shengbo Eben Li, Sifa Zheng, and Jianyu Chen. Joint synthesis of safety certificate and safe control policy using constrained reinforcement learning. In *Learning for Dynamics and Control Conference*, pages 97–109. PMLR, 2022.

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

[22] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

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

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

[25] Long Yang, Jiaming Ji, Juntao Dai, Linrui Zhang, Binbin Zhou, Pengfei Li, Yaodong Yang, and Gang Pan. Constrained update projection approach to safe policy optimization. *Advances in Neural Information Processing Systems*, 35:9111–9124, 2022.

[26] 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, pages 4940–4947, 2020.

[27] Juntao Dai, Jiaming Ji, Long Yang, Qian Zheng, and Gang Pan. Augmented proximal policy optimization for safe reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7288–7295, 2023.

[28] Weiye Zhao, Rui Chen, Yifan Sun, Tianhao Wei, and Changliu Liu. State-wise constrained policy optimization. *arXiv preprint arXiv:2306.12594*, 2023.

[29] Weiye Zhao, Feihan Li, Yifan Sun, Yujie Wang, Rui Chen, Tianhao Wei, and Changliu Liu. Absolute state-wise constrained policy optimization: High-probability state-wise constraints satisfaction. *arXiv preprint arXiv:2410.01212*, 2024.

[30] Yujie Yang, Zhilong Zheng, Shengbo Eben Li, Jingliang Duan, Jingjing Liu, Xianyuan Zhan, and Ya-Qin Zhang. Feasible policy iteration for safe reinforcement learning. *arXiv preprint arXiv:2304.08845*, 2025.

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

[32] 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.

[33] Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking Safe Exploration in Deep Reinforcement Learning. 2019.

[34] Linrui Zhang, Li Shen, Long Yang, Shixiang Chen, Xueqian Wang, Bo Yuan, and Dacheng Tao. Penalized proximal policy optimization for safe reinforcement learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 3744–3750, 2022. Main Track.

A Proofs

A.1 Proof of shared solution theorem

Theorem 1. *There exists a policy π_{k+1} that is the optimal solution to both Problem (7) and (8).*

Proof. Let π_{in} and π_{out} denote the optimal solutions to Problem (7) and (8), respectively. We construct the policy π_{k+1} as follows:

$$\pi_{k+1}(\cdot|x) = \begin{cases} \pi_{\text{in}}(\cdot|x), & x \in \mathcal{R}^{\pi_{\text{in}}}(\mathcal{X}_{\text{init}} \cap \mathrm{X}^{\pi_k}), \\ \pi_{\text{out}}(\cdot|x), & \text{otherwise}, \end{cases} \tag{18}$$

where $\mathcal{R}^{\pi}(X) = \bigcup_{x \in X} \mathcal{R}^{\pi}(x)$ denotes the reachable set of π from a set of states $X \subseteq \mathcal{X}$.

By construction, the trajectories of π_{k+1} starting from $\mathcal{X}_{\text{init}} \cap \mathrm{X}^{\pi_k}$ coincide with those of π_{in}. Therefore,

$$\mathbb{E}_{x \sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x) \leq 0] F^{\pi_{k+1}}(x)] = \mathbb{E}_{x \sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x) \leq 0] F^{\pi_{\text{in}}}(x)] \leq 0,$$

which proves that π_{k+1} satisfies the shared constraint of both problems.

Since π_{in} is optimal for Problem (7), and π_{k+1} achieves the same value function as π_{in} for all $x \in \mathcal{X}_{\text{init}} \cap \mathrm{X}^{\pi_k}$, it follows that

$$\mathbb{E}_{x \sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x) \leq 0] V^{\pi_{k+1}}(x)] = \mathbb{E}_{x \sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x) \leq 0] V^{\pi_{\text{in}}}(x)].$$

Thus, π_{k+1} is also optimal for Problem (7).

For any $x \in \mathcal{X}_{\text{init}} \setminus \mathrm{X}^{\pi_k}$, we analyze two cases: (1) No future state enters $\mathcal{R}^{\pi_{\text{in}}}(\mathcal{X}_{\text{init}} \cap \mathrm{X}^{\pi_k})$. In this case, $\pi_{k+1} = \pi_{\text{out}}$ for all future states, thus $F^{\pi_{k+1}}(x) = F^{\pi_{\text{out}}}(x)$. (2) There exists a future state that enters $\mathcal{R}^{\pi_{\text{in}}}(\mathcal{X}_{\text{init}} \cap \mathrm{X}^{\pi_k})$ in finite time. In this case, π_{k+1} switches to π_{in} once entered, ensuring no future constraint violation. Thus, $F^{\pi_{k+1}}(x) \leq F^{\pi_{\text{out}}}(x)$. Combining these two cases, we have

$$\forall x \in \mathcal{X}_{\text{init}} \setminus \mathrm{X}^{\pi_k}, F^{\pi_{k+1}}(x) \leq F^{\pi_{\text{out}}}(x),$$

which implies

$$\mathbb{E}_{x \sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x) > 0] F^{\pi_{k+1}}(x)] \leq \mathbb{E}_{x \sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x) > 0] F^{\pi_{\text{out}}}(x)].$$

Since π_{out} is optimal for Problem (8), π_{k+1} is also optimal. Therefore, we conclude that π_{k+1} is the optimal solution to both Problem (7) and (8). □

A.2 Proof of CDF bounds

Lemma 1. *For any policies $\tilde{\pi}$ and π, and for any state $x \in \mathcal{X}$,*

$$F^{\tilde{\pi}}(x) - F^{\pi}(x) = \mathbb{E}_{\tau \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \prod_{s=0}^{t-1} (1 - c_s) A_F^{\pi}(x_t, u_t) \Big| x_0 = x \right].$$

Proof. By definition of F^{π}, we have

$$\begin{aligned} F^{\pi}(x) &= \mathbb{E}_{\tau \sim \pi}[c_0 + (1 - c_0)\gamma(c_1 + (1 - c_1)\gamma(\dots))|x_0 = x] \\ &= \mathbb{E}_{\tau \sim \pi}[c_0 + \gamma(1 - c_0)c_1 + \gamma^2(1 - c_0)(1 - c_1)c_2 + \dots |x_0 = x] \\ &= \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \prod_{s=0}^{t-1} (1 - c_s) c_t \Big| x_0 = x \right]. \end{aligned}$$

Thus,

$$\begin{aligned} F^{\tilde{\pi}}(x) - F^{\pi}(x) &= \mathbb{E}_{\tau \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \prod_{s=0}^{t-1} (1 - c_s) c_t \Big| x_0 = x \right] - F^{\pi}(x) \\ &= \mathbb{E}_{\tau \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \prod_{s=0}^{t-1} (1 - c_s)(c_t + (1 - c_t)\gamma F^{\pi}(x_{t+1}) - F^{\pi}(x_t)) \Big| x_0 = x \right] \\ &= \mathbb{E}_{\tau \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \prod_{s=0}^{t-1} (1 - c_s) A_F^{\pi}(x_t, u_t) \Big| x_0 = x \right]. \end{aligned}$$

□

Definition 5 (Constraint-absorbing counterpart)**.** *Let M be an MDP with transition probability P. The constraint-absorbing counterpart of M, denoted $\tilde{M}$, is an MDP with all elements equal those of M except the transition probability, which is defined as:*

$$\tilde{P}(x'|x,u) = \begin{cases} P(x'|x,u), & c(x) = 0, \\ 1, & c(x) = 1 \text{ and } x' = x, \\ 0, & c(x) = 1 \text{ and } x' \neq x, \end{cases}$$

We also call such $\tilde{M}$ a constraint-absorbing MDP.

Lemma 2. *In a constraint-absorbing MDP $\tilde{M}$, for any policies $\tilde{\pi}$ and π, and any state $x \in \mathcal{X}$,*

$$\tilde{F}^{\tilde{\pi}}(x) - \tilde{F}^{\pi}(x) = \mathbb{E}_{\tau \sim (\tilde{\pi}, \tilde{P})} \left[\sum_{t=0}^{\infty} \gamma^t \tilde{A}_F^{\pi}(x_t, u_t) \Big| x_0 = x \right].$$

Proof. According to Lemma 1, we have

$$\tilde{F}^{\tilde{\pi}}(x) - \tilde{F}^{\pi}(x) = \mathbb{E}_{\tau \sim (\tilde{\pi}, \tilde{P})} \left[\sum_{t=0}^{\infty} \gamma^t \prod_{s=0}^{t-1} (1 - c_s) \tilde{A}_F^{\pi}(x_t, u_t) \Big| x_0 = x \right].$$

We split the summation into two parts:

$$\begin{aligned} \tilde{F}^{\tilde{\pi}}(x) - \tilde{F}^{\pi}(x) = \mathbb{E}_{\tau \sim (\tilde{\pi}, \tilde{P})} \Bigg[& \sum_{t=0}^{N(\tau)} \gamma^t \prod_{s=0}^{t-1} (1 - c_s) \tilde{A}_F^{\pi}(x_t, u_t) \\ & + \sum_{t=N(\tau)+1}^{\infty} \gamma^t \prod_{s=0}^{t-1} (1 - c_s) \tilde{A}_F^{\pi}(x_t, u_t) \Big| x_0 = x \Bigg]. \end{aligned}$$

For any trajectory $\tau \sim (\tilde{\pi}, \tilde{P})$, for all $t \leq N(\tau)$, we have $c_{t-1} = 0$. Thus,

$$\sum_{t=0}^{N(\tau)} \gamma^t \prod_{s=0}^{t-1} (1 - c_s) \tilde{A}_F^{\pi}(x_t, u_t) = \sum_{t=0}^{N(\tau)} \gamma^t \tilde{A}_F^{\pi}(x_t, u_t).$$

For all $t > N(\tau)$, since $c_{N(\tau)} = 1$, we have

$$\sum_{t=N(\tau)+1}^{\infty} \gamma^t \prod_{s=0}^{t-1} (1 - c_s) \tilde{A}_F^{\pi}(x_t, u_t) = 0.$$

By definition of $\tilde{P}$, for all $t > N(\tau)$, we have $x_t = x_{N(\tau)}$, $\tilde{F}^{\pi}(x_t) = c_t = 1$, and it follows that

$$\tilde{A}_F^{\pi}(x_t, u_t) = \mathbb{E}_{x_{t+1} \sim \tilde{P}(\cdot|x_t, u_t)} [c_t + (1 - c_t)\gamma \tilde{F}^{\pi}(x_{t+1}) - \tilde{F}^{\pi}(x_t)] = 0.$$

Thus, we can equivalently write the second half of the summation as follows:

$$\sum_{t=N(\tau)+1}^{\infty} \gamma^t \prod_{s=0}^{t-1} (1 - c_s) \tilde{A}_F^{\pi}(x_t, u_t) = \sum_{t=N(\tau)+1}^{\infty} \gamma^t \tilde{A}_F^{\pi}(x_t, u_t).$$

Therefore, we conclude that

$$\tilde{F}^{\tilde{\pi}}(x) - \tilde{F}^{\pi}(x) = \mathbb{E}_{\tau \sim (\tilde{\pi}, \tilde{P})} \left[\sum_{t=0}^{\infty} \gamma^t \tilde{A}_F^{\pi}(x_t, u_t) \Big| x_0 = x \right].$$

□

Lemma 3 (CDF equivalence)**.** *Let F^{π} be the CDF in an MDP M, and $\tilde{F}^{\pi}$ be the CDF in $\tilde{M}$. For any policy π and state $x \in \mathcal{X}$, we have*

$$\tilde{F}^{\pi}(x) = F^{\pi}(x).$$

Proof. Consider a trajectory where the first constraint violation happens at time step t, and we denote it as τ_t, i.e., $\tau_t = \{x_0, u_0, x_1, u_1, \dots\}$, where $c(x_t) = 1$ and $c(x_s) = 0, \forall s < t$. We split τ_t into two parts:

$$\tau_{\leq t} = \{x_0, u_0, x_1, u_1, \dots, x_t\} \text{ and } \tau_{>t} = \{u_t, x_{t+1}, u_{t+1}, \dots\}.$$

The probability of τ_t under the original MDP M can be decomposed as follows:

$$p(\tau_t) = p(\tau_{\leq t}) p(\tau_{>t} | \tau_{\leq t}),$$

where

$$p(\tau_{\leq t}) = \mathbb{I}[x_0 = x] \prod_{s=0}^{t-1} \pi(u_s|x_s) P(x_{s+1}|x_s, u_s),$$

$$p(\tau_{>t}|\tau_{\leq t}) = \prod_{s=t}^{\infty} \pi(u_s|x_s) P(x_{s+1}|x_s, u_s).$$

Using the decomposed probability, the CDF can be expressed as:

$$\begin{aligned}
F^\pi(x) &= \sum_\tau p(\tau) \gamma^{N(\tau)} \\
&= \sum_{t=0}^{\infty} \sum_{\tau_t} p(\tau_t) \gamma^t \\
&= \sum_{t=0}^{\infty} \sum_{\tau_{\leq t}, \tau_{>t}} p(\tau_{\leq t}) p(\tau_{>t}|\tau_{\leq t}) \gamma^t \\
&= \sum_{t=0}^{\infty} \sum_{\tau_{\leq t}} p(\tau_{\leq t}) \underbrace{\sum_{\tau_{>t}} p(\tau_{>t}|\tau_{\leq t})}_{=1} \gamma^t \\
&= \sum_{t=0}^{\infty} \sum_{\tau_{\leq t}} p(\tau_{\leq t}) \gamma^t.
\end{aligned}$$

Similarly, the CDF in $\tilde{M}$ can be expressed as

$$\tilde{F}^\pi(x) = \sum_{t=0}^{\infty} \sum_{\tau_{\leq t}} \tilde{p}(\tau_{\leq t}) \gamma^t,$$

where

$$\tilde{p}(\tau_{\leq t}) = \mathbb{I}[x_0 = x] \prod_{s=0}^{t-1} \pi(u_s|x_s) \tilde{P}(x_{s+1}|x_s, u_s).$$

Since the transition probability $\tilde{P}$ is identical to P up to the first constraint violation, we have $\tilde{p}(\tau_{\leq t}) = p(\tau_{\leq t})$, and thus $\tilde{F}^\pi(x) = F^\pi(x)$. □

Lemma 4 (Feasibility advantage equivalence)**.** *Let A_F^π be the feasibility advantage in an MDP M, and $\tilde{A}_F^\pi$ be the feasibility advantage in $\tilde{M}$. For any policy π, state $x \in \mathcal{X}$, and action $u \in \mathcal{U}$, we have*

$$\tilde{A}_F^\pi(x, u) = A_F^\pi(x, u).$$

Proof. By definition of the feasibility advantage,

$$\tilde{A}_F^\pi(x, u) = \mathbb{E}_{x' \sim \tilde{P}(\cdot|x,u)}[c(x) + (1 - c(x))\gamma \tilde{F}^\pi(x') - \tilde{F}^\pi(x)].$$

By Lemma 3, we can replace $\tilde{F}^\pi$ with F^π:

$$\tilde{A}_F^\pi(x, u) = \mathbb{E}_{x' \sim \tilde{P}(\cdot|x,u)}[c(x) + (1 - c(x))\gamma F^\pi(x') - F^\pi(x)].$$

Now, the only difference between $\tilde{A}_F^\pi(x, u)$ and $A_F^\pi(x, u)$ lies in the transition probability. We analyze two cases: whether state x violates the constraint or not. If $c(x) = 0$, we have $\tilde{P}(\cdot|x, u) = P(\cdot|x, u)$. In this case, $\tilde{A}_F^\pi(x, u) = A_F^\pi(x, u)$. If $c(x) = 1$, we have $F^\pi(x) = 1$. In this case,

$$c(x) + (1 - c(x))\gamma F^\pi(x') - F^\pi(x) = 0,$$

and thus $\tilde{A}_F^\pi(x, u) = A_F^\pi(x, u) = 0$. Therefore, $\tilde{A}_F^\pi(x, u) = A_F^\pi(x, u)$ holds for all $x \in \mathcal{X}$. □

Lemma 5 (Prefix state distribution equivalence)**.** *Let d_0^π be the prefix state distribution in an MDP M, and $\tilde{d}_0^\pi$ be the prefix state distribution in $\tilde{M}$. For any policy π, initial state $x \in \mathcal{X}$, and future state $x' \in \mathcal{X}$, we have*

$$\tilde{d}_0^\pi(x'|x) = d_0^\pi(x'|x).$$

Proof. Expand the probability in each term of the summation,

$$P(x_t = x', \max_{s<t} c_s = 0|x, \pi) = \sum_{\substack{x_1, x_2, \ldots, x_{t-1} \in \mathrm{X}_{\text{cstr}} \\ u_0, u_1, \ldots, u_{t-1} \in \mathcal{U}}} \pi(u_0|x) P(x_1|x, u_0) \pi(u_1|x_1) \cdots P(x'|x_{t-1}, u_{t-1}).$$

Since $c_s = 0, \forall s < t$, by definition of $\tilde{P}$, we have

$$\tilde{P}(x_{s+1}|x_s, u_s) = P(x_{s+1}|x_s, u_s), \forall s < t.$$

Thus, it follows that

$$\tilde{P}(x_t = x', \max_{s<t} c_s = 0|x, \pi) = P(x_t = x', \max_{s<t} c_s = 0|x, \pi),$$

which implies that $\tilde{d}_0^\pi(x'|x) = d_0^\pi(x'|x)$. □

Lemma 6. *For any policies $\tilde{\pi}$ and π, and state $x \in \mathcal{X}$,*

$$F^{\tilde{\pi}}(x) - F^\pi(x) = \frac{1}{1-\gamma} \mathbb{E}_{x' \sim d_0^{\tilde{\pi}}(\cdot|x), u' \sim \tilde{\pi}(\cdot|x')}[A_F^\pi(x', u')].$$

Proof. By Lemma 2, we have

$$\begin{aligned}
\tilde{F}^{\tilde{\pi}}(x) - \tilde{F}^\pi(x) &= \mathbb{E}_{\tau \sim (\tilde{\pi}, \tilde{P})} \left[\sum_{t=0}^\infty \gamma^t \tilde{A}_F^\pi(x_t, u_t) \Big| x_0 = x \right] \\
&= \sum_{t=0}^\infty \sum_{x'} \tilde{P}(x_t = x'|x, \tilde{\pi}) \sum_{u'} \tilde{\pi}(u'|x') \gamma^t \tilde{A}_F^\pi(x', u').
\end{aligned}$$

For any $t \geq 0$, if $\max_{s<t} c_s > 0$, the state will be fixed at the constraint-violating one in the constraint absorbing MDP. Thus, only those x' that violate the constraint yield $\tilde{P}(x_t = x'|x, \tilde{\pi}) > 0$. For these x', we have $\tilde{A}_F^\pi(x', u') = 0$. Therefore, we only need to consider the terms with $\max_{s<t} c_s = 0$ in the summation, i.e.,

$$\begin{aligned}
\tilde{F}^{\tilde{\pi}}(x) - \tilde{F}^\pi(x) &= \sum_{t=0}^\infty \sum_{x'} \tilde{P}(x_t = x'|x, \tilde{\pi}, \max_{s<t} c_s = 0) \sum_{u'} \tilde{\pi}(u'|x') \gamma^t \tilde{A}_F^\pi(x', u') \\
&= \sum_{x'} \sum_{t=0}^\infty \gamma^t \tilde{P}(x_t = x'|x, \tilde{\pi}, \max_{s<t} c_s = 0) \sum_{u'} \tilde{\pi}(u'|x') \tilde{A}_F^\pi(x', u') \\
&= \sum_{x'} \frac{1}{1-\gamma} \tilde{d}_0^{\tilde{\pi}}(x'|x) \sum_{u'} \tilde{\pi}(u'|x') \tilde{A}_F^\pi(x', u') \\
&= \frac{1}{1-\gamma} \mathbb{E}_{x' \sim \tilde{d}_0^{\tilde{\pi}}(\cdot|x), u' \sim \tilde{\pi}(\cdot|x')}[\tilde{A}_F^\pi(x', u')].
\end{aligned}$$

Substitute in the result from Lemma 3, 4, and 5, we have

$$F^{\tilde{\pi}}(x) - F^\pi(x) = \frac{1}{1-\gamma} \mathbb{E}_{x' \sim d_0^{\tilde{\pi}}(\cdot|x), u' \sim \tilde{\pi}(\cdot|x')}[A_F^\pi(x', u')].$$

□

Lemma 7. *For any policies $\tilde{\pi}$ and π, and any state $x \in \mathcal{X}$, define*

$$L_{\tilde{\pi}}^\pi(x) = \mathbb{E}_{x' \sim d_0^\pi(\cdot|x), u' \sim \pi(\cdot|x')} \left[\frac{\tilde{\pi}(u'|x')}{\pi(u'|x')} A_F^\pi(x', u') \right],$$

and $\epsilon_F^{\tilde{\pi}} = \max_{x'} |\mathbb{E}_{u' \sim \tilde{\pi}(\cdot|x')}[A_F^{\pi}(x', u')]|$. *The following bounds hold:*

$$F^{\tilde{\pi}}(x) - F^{\pi}(x) \geq \frac{1}{1-\gamma} \left(L_{\tilde{\pi}}^{\pi}(x) - 2\epsilon_F^{\tilde{\pi}} D_{TV}(d_0^{\tilde{\pi}}(\cdot|x) \| d_0^{\pi}(\cdot|x)) \right),$$
$$F^{\tilde{\pi}}(x) - F^{\pi}(x) \leq \frac{1}{1-\gamma} \left(L_{\tilde{\pi}}^{\pi}(x) + 2\epsilon_F^{\tilde{\pi}} D_{TV}(d_0^{\tilde{\pi}}(\cdot|x) \| d_0^{\pi}(\cdot|x)) \right),$$

where D_{TV} *is the total variational divergence. Furthermore, the bounds are tight (when* $\tilde{\pi} = \pi$*, the LHS and RHS are identically zero).*

Proof. This proof is largely borrowed from Lemma 2 in CPO [12].

Let $\bar{A}_F^{\pi} \in \mathbb{R}^{|\mathcal{X}|}$ denote the vector of components $\bar{A}_F^{\pi}(x') = \mathbb{E}_{u' \sim \tilde{\pi}(\cdot|x')}[A_F^{\pi}(x', u')]$. With an abuse of notation, we view $d_0^{\pi}(\cdot|x)$ as a vector in $\mathbb{R}^{|\mathcal{X}|}$ when necessary. Beginning with the result in Lemma 6, we have

$$\begin{aligned}(1-\gamma)(F^{\tilde{\pi}}(x) - F^{\pi}(x)) &= \mathbb{E}_{x' \sim d_0^{\tilde{\pi}}(\cdot|x), u' \sim \tilde{\pi}(\cdot|x')}[A_F^{\pi}(x', u')] \\ &= \left\langle d_0^{\tilde{\pi}}(\cdot|x), \bar{A}_F^{\pi} \right\rangle \\ &= \left\langle d_0^{\pi}(\cdot|x), \bar{A}_F^{\pi} \right\rangle + \left\langle d_0^{\tilde{\pi}}(\cdot|x) - d_0^{\pi}(\cdot|x), \bar{A}_F^{\pi} \right\rangle.\end{aligned}$$

This term can be bounded by Holder's inequality: for any $p, q \in [1, \infty]$ such that $1/p + 1/q = 1$, we have

$$(1-\gamma)(F^{\tilde{\pi}}(x) - F^{\pi}(x)) \geq \left\langle d_0^{\pi}(\cdot|x), \bar{A}_F^{\pi} \right\rangle - \left\| d_0^{\tilde{\pi}}(\cdot|x) - d_0^{\pi}(\cdot|x) \right\|_p \left\| \bar{A}_F^{\pi} \right\|_q,$$
$$(1-\gamma)(F^{\tilde{\pi}}(x) - F^{\pi}(x)) \leq \left\langle d_0^{\pi}(\cdot|x), \bar{A}_F^{\pi} \right\rangle + \left\| d_0^{\tilde{\pi}}(\cdot|x) - d_0^{\pi}(\cdot|x) \right\|_p \left\| \bar{A}_F^{\pi} \right\|_q.$$

Choose $p = 1, q = \infty$, we have $\|d_0^{\tilde{\pi}}(\cdot|x) - d_0^{\pi}(\cdot|x)\|_1 = 2D_{TV}(d_0^{\tilde{\pi}}(\cdot|x) \| d_0^{\pi}(\cdot|x))$ and $\|\bar{A}_F^{\pi}\|_{\infty} = \epsilon_F^{\tilde{\pi}}$. Observe that by importance sampling,

$$\begin{aligned}\left\langle d_0^{\pi}(\cdot|x), \bar{A}_F^{\pi} \right\rangle &= \mathbb{E}_{x' \sim d_0^{\pi}(\cdot|x), u' \sim \tilde{\pi}(\cdot|x')}[A_F^{\pi}(x', u')] \\ &= \mathbb{E}_{x' \sim d_0^{\pi}(\cdot|x), u' \sim \pi(\cdot|x')} \left[\frac{\tilde{\pi}(u'|x')}{\pi(u'|x')} A_F^{\pi}(x', u') \right] \\ &= L_{\tilde{\pi}}^{\pi}(x).\end{aligned}$$

After rearranging terms, the bounds are obtained. □

Lemma 8. *For any policies* $\tilde{\pi}$ *and* π*, and state* $x \in \mathcal{X}$*,*

$$\left\| d_0^{\tilde{\pi}}(\cdot|x) - d_0^{\pi}(\cdot|x) \right\|_1 \leq \frac{2\gamma}{1-\gamma} \mathbb{E}_{x' \sim d_0^{\pi}(\cdot|x)} \left[D_{TV}(\tilde{\pi} \| \pi)[x'] \right],$$

where $D_{TV}(\tilde{\pi} \| \pi)[x'] = (1/2) \sum_u |\tilde{\pi}(u|x') - \pi(u|x')|$.

Proof. We prove that the result holds for the prefix state distribution in a constraint-absorbing MDP, i.e., $\tilde{d}_0^{\tilde{\pi}}$ and $\tilde{d}_0^{\pi}$. Since $\tilde{d}_0^{\pi} = d_0^{\pi}$ for any π, the result to prove directly follows.

Let $\tilde{P}^{\pi}(x'|x) = \sum_u \tilde{P}(x'|x, u)\pi(u|x)$. We view $\tilde{P}^{\pi}$ as a matrix in $\mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$, where the element on the ith row and jth column, $\tilde{P}_{ij}^{\pi}$, denotes the transition probability from the jth state to the ith state. We rearrange the order of the states in $\tilde{P}^{\pi}$ so that all constraint-violating states are located on the last rows and columns:

$$\tilde{P}^{\pi} = \begin{bmatrix} \tilde{P}_{\text{s}}^{\pi} & O \\ \tilde{P}_{\text{v}}^{\pi} & I \end{bmatrix},$$

where $\tilde{P}_{\text{s}}^{\pi}$ denotes the transition probability between constraint-satisfying states, $\tilde{P}_{\text{v}}^{\pi}$ denotes the transition probability from constraint-satisfying states to constraint-violating states, O denotes the zero matrix, and I denotes the identity matrix, which implies that a constraint-violating state will no longer transfer to other states. We construct another matrix by setting the identity matrix in $\tilde{P}^{\pi}$ to zero:

$$\tilde{P}_0^{\pi} = \begin{bmatrix} \tilde{P}_{\text{s}}^{\pi} & O \\ \tilde{P}_{\text{v}}^{\pi} & O \end{bmatrix}.$$

By definition of the prefix state distribution,

$$\tilde{d}_0^{\pi}(\cdot|x) = (1-\gamma)\sum_{t=0}^{\infty}\left(\gamma\tilde{P}_0^{\pi}\right)^t e_x = (1-\gamma)\left(I-\gamma\tilde{P}_0^{\pi}\right)^{-1} e_x,$$

where e_x is a one-hot vector where the element at the position of state x is one, and all other elements are zero, which implies that the initial state is fixed at x.

Define matrices $G = (I-\gamma\tilde{P}_0^{\pi})^{-1}$, $\tilde{G} = (I-\gamma\tilde{P}_0^{\tilde{\pi}})^{-1}$, and $\Delta = \tilde{P}_0^{\tilde{\pi}} - \tilde{P}_0^{\pi}$. Then,

$$G^{-1} - \tilde{G}^{-1} = \left(I-\gamma\tilde{P}_0^{\pi}\right) - \left(I-\gamma\tilde{P}_0^{\tilde{\pi}}\right) = \gamma\Delta.$$

Left-multiplying by $\tilde{G}$ and right-multiplying by G, we obtain

$$\tilde{G} - G = \gamma\tilde{G}\Delta G.$$

Thus,

$$\begin{aligned}\tilde{d}_0^{\tilde{\pi}}(\cdot|x) - \tilde{d}_0^{\pi}(\cdot|x) &= (1-\gamma)\left(\tilde{G}-G\right)e_x \\ &= \gamma(1-\gamma)\tilde{G}\Delta G e_x \\ &= \gamma\tilde{G}\Delta\tilde{d}_0^{\pi}(\cdot|x).\end{aligned}$$

Taking the L1 norm on both sides, we obtain

$$\left\|\tilde{d}_0^{\tilde{\pi}}(\cdot|x) - \tilde{d}_0^{\pi}(\cdot|x)\right\|_1 = \gamma\left\|\tilde{G}\Delta\tilde{d}_0^{\pi}(\cdot|x)\right\|_1 \le \gamma\left\|\tilde{G}\right\|_1\left\|\Delta\tilde{d}_0^{\pi}(\cdot|x)\right\|_1.$$

$\|\tilde{G}\|_1$ is bounded by

$$\left\|\tilde{G}\right\|_1 = \left\|\left(I-\gamma\tilde{P}_0^{\tilde{\pi}}\right)^{-1}\right\|_1 \le \sum_{t=0}^{\infty}\gamma^t\left\|\tilde{P}_0^{\tilde{\pi}}\right\|_1^t \le \sum_{t=0}^{\infty}\gamma^t = (1-\gamma)^{-1}.$$

$\|\Delta\tilde{d}_0^{\pi}(\cdot|x)\|_1$ is bounded by

$$\begin{aligned}\left\|\Delta\tilde{d}_0^{\pi}(\cdot|x)\right\|_1 &= \sum_{x''}\left|\sum_{x'}\Delta(x''|x')\tilde{d}_0^{\pi}(x'|x)\right| \\ &\le \sum_{x',x''}\left|\Delta(x''|x')\right|\tilde{d}_0^{\pi}(x'|x) \\ &= \sum_{x',x''}\left|\sum_{u'}\tilde{P}(x''|x',u')(\tilde{\pi}(u'|x') - \pi(u'|x'))\right|\tilde{d}_0^{\pi}(x'|x) \\ &\le \sum_{x',u',x''}P(x''|x',u')\left|\tilde{\pi}(u'|x') - \pi(u'|x')\right|\tilde{d}_0^{\pi}(x'|x) \\ &= \sum_{x',u'}\left|\tilde{\pi}(u'|x') - \pi(u'|x')\right|\tilde{d}_0^{\pi}(x'|x) \\ &= 2\mathbb{E}_{x'\sim\tilde{d}_0^{\pi}(\cdot|x)}\left[D_{TV}(\tilde{\pi}\|\pi)[x']\right].\end{aligned}$$

Therefore,

$$\left\|\tilde{d}_0^{\tilde{\pi}}(\cdot|x) - \tilde{d}_0^{\pi}(\cdot|x)\right\|_1 \le \frac{2\gamma}{1-\gamma}\mathbb{E}_{x'\sim\tilde{d}_0^{\pi}(\cdot|x)}\left[D_{TV}(\tilde{\pi}\|\pi)[x']\right].$$

□

Theorem 2. *For any policies $\tilde{\pi}$ and π, and any state $x \in \mathcal{X}$, define*

$$A_F^{\pi}(x,u) = \mathbb{E}_{x'\sim P(\cdot|x,u)}[c(x) + (1-c(x))\gamma F^{\pi}(x') - F^{\pi}(x)],$$

and $L_{\tilde{\pi}}^{\pi}(x) = \mathbb{E}_{x'\sim d_0^{\pi}(\cdot|x),u'\sim\tilde{\pi}(\cdot|x')}\left[A_F^{\pi}(x',u')\right]$, $\epsilon_F^{\tilde{\pi}} = \max_x\left|\mathbb{E}_{u\sim\tilde{\pi}(\cdot|x)}[A_F^{\pi}(x,u)]\right|$. Then,

$$F^{\tilde{\pi}}(x) - F^{\pi}(x) \ge \frac{L_{\tilde{\pi}}^{\pi}(x)}{1-\gamma} - \frac{2\gamma\epsilon_F^{\tilde{\pi}}}{(1-\gamma)^2}\mathbb{E}_{x'\sim d_0^{\pi}(\cdot|x)}[D_{TV}(\tilde{\pi}\|\pi)[x']],$$

$$F^{\tilde{\pi}}(x) - F^{\pi}(x) \le \frac{L_{\tilde{\pi}}^{\pi}(x)}{1-\gamma} + \frac{2\gamma\epsilon_F^{\tilde{\pi}}}{(1-\gamma)^2}\mathbb{E}_{x'\sim d_0^{\pi}(\cdot|x)}[D_{TV}(\tilde{\pi}\|\pi)[x']],$$

where $D_{TV}(\tilde{\pi}\|\pi)[x'] = (1/2)\sum_u |\tilde{\pi}(u|x') - \pi(u|x')|$ is the total variational divergence between action distributions at x'. Furthermore, the bounds are tight (when $\tilde{\pi} = \pi$, the LHS and RHS are identically zero).

Proof. Begin with the bounds from Lemma 7 and bound the divergence by Lemma 8. □

A.3 Proof of performance bounds

Corollary 2. *The optimal solution to Problem (11), denoted π_{k+1}, satisfies the following two properties:*

1. (Feasibility enhancement) Let $\epsilon_F^{\pi_{k+1}} = \max_x |\mathbb{E}_{u\sim\pi_{k+1}(\cdot|x)}[A_F^{\pi}(x,u)]|$, we have

$$\mathbb{E}_{x\sim d_{init}}\left[\mathbb{I}[F^{\pi_k}(x) \le 0]F^{\pi_{k+1}}(x)\right] \le \frac{\sqrt{\delta}\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}, \tag{12a}$$

$$\mathbb{E}_{x\sim d_{init}}\left[\mathbb{I}[F^{\pi_k}(x) > 0]\left(F^{\pi_{k+1}}(x) - F^{\pi_k}(x)\right)\right] \le \frac{\sqrt{\delta}\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}. \tag{12b}$$

2. (Value improvement) Let $\epsilon^{\pi_{k+1}} = \max_x |\mathbb{E}_{u\sim\pi_{k+1}(\cdot|x)}[A^{\pi}(x,u)]|$, we have

$$\mathbb{E}_{x\sim d_{init}}\left[\mathbb{I}[F^{\pi_k}(x) \le 0]\left(V^{\pi_{k+1}}(x) - V^{\pi_k}(x)\right)\right] \ge -\frac{\sqrt{\delta}\gamma\epsilon^{\pi_{k+1}}}{(1-\gamma)^2}. \tag{13}$$

Proof. Consider the following two problems:

$$\begin{aligned}
\max_{\pi}\ & \mathbb{E}_{x\sim d^{\pi_k},u\sim\pi}\left[\mathbb{I}[F^{\pi_k}(x) \le 0]A^{\pi_k}(x,u)\right] \\
\text{s.t. } & \mathbb{E}_{x\sim d^{\pi_k},u\sim\pi}\left[\mathbb{I}[F^{\pi_k}(x) \le 0](F^{\pi_k}(x) + A_F^{\pi_k}(x,u)/(1-\gamma))_+\right] \le 0 \\
& \mathbb{E}_{x\sim d^{\pi_k}}\left[\mathbb{I}[F^{\pi_k}(x) \le 0]D_{KL}(\pi\|\pi_k)[x]\right] \le \delta/2 \\
& \mathbb{E}_{x\sim d^{\pi_k}}\left[\mathbb{I}[F^{\pi_k}(x) > 0]D_{KL}(\pi\|\pi_k)[x]\right] \le \delta/2,
\end{aligned} \tag{19}$$

and

$$\begin{aligned}
\min_{\pi}\ & \mathbb{E}_{x\sim d_0^{\pi_k},u\sim\pi}\left[\mathbb{I}[F^{\pi_k}(x) > 0]A_F^{\pi_k}(x,u)\right] \\
\text{s.t. } & \mathbb{E}_{x\sim d^{\pi_k},u\sim\pi}\left[\mathbb{I}[F^{\pi_k}(x) \le 0](F^{\pi_k}(x) + A_F^{\pi_k}(x,u)/(1-\gamma))_+\right] \le 0 \\
& \mathbb{E}_{x\sim d^{\pi_k}}\left[\mathbb{I}[F^{\pi_k}(x) \le 0]D_{KL}(\pi\|\pi_k)[x]\right] \le \delta/2 \\
& \mathbb{E}_{x\sim d^{\pi_k}}\left[\mathbb{I}[F^{\pi_k}(x) > 0]D_{KL}(\pi\|\pi_k)[x]\right] \le \delta/2,
\end{aligned} \tag{20}$$

We prove that they have the same optimal solution. Let π_{in} and π_{out} denote the optimal solutions to Problem (19) and (20), respectively. Construct the following policy:

$$\pi_{k+1}(\cdot|x) = \begin{cases} \pi_{\text{in}}(\cdot|x), & x \in \mathrm{X}^{\pi_k}, \\ \pi_{\text{out}}(\cdot|x), & \text{otherwise.} \end{cases}$$

We first prove that π_{k+1} satisfies the constraints of Problem (19) and (20). For the first constraint, we have

$$\begin{aligned}
&\mathbb{E}_{x\sim d^{\pi_k},u\sim\pi_{k+1}}\left[\mathbb{I}[F^{\pi_k}(x) \le 0](F^{\pi_k}(x) + A_F^{\pi_k}(x,u)/(1-\gamma))_+\right] \\
=\ &\mathbb{E}_{x\sim d^{\pi_k},u\sim\pi_{\text{in}}}\left[\mathbb{I}[F^{\pi_k}(x) \le 0](F^{\pi_k}(x) + A_F^{\pi_k}(x,u)/(1-\gamma))_+\right] \le 0.
\end{aligned}$$

For the second and third constraints, we have

$$\begin{aligned}
&\mathbb{E}_{x\sim d^{\pi_k}}\left[\mathbb{I}[F^{\pi_k}(x) \le 0]D_{KL}(\pi_{k+1}\|\pi_k)[x]\right] = \mathbb{E}_{x\sim d^{\pi_k}}\left[\mathbb{I}[F^{\pi_k}(x) \le 0]D_{KL}(\pi_{\text{in}}\|\pi_k)[x]\right] \le \delta/2, \\
&\mathbb{E}_{x\sim d^{\pi_k}}\left[\mathbb{I}[F^{\pi_k}(x) > 0]D_{KL}(\pi_{k+1}\|\pi_k)[x]\right] = \mathbb{E}_{x\sim d^{\pi_k}}\left[\mathbb{I}[F^{\pi_k}(x) > 0]D_{KL}(\pi_{\text{out}}\|\pi_k)[x]\right] \le \delta/2.
\end{aligned}$$

Thus, π_{k+1} satisfies the constraints of both problems. For the objective function of Problem (19), we have

$$\mathbb{E}_{x\sim d^{\pi_k},u\sim\pi_{k+1}}\left[\mathbb{I}[F^{\pi_k}(x) \le 0]A^{\pi_k}(x,u)\right] = \mathbb{E}_{x\sim d^{\pi_k},u\sim\pi_{\text{in}}}\left[\mathbb{I}[F^{\pi_k}(x) \le 0]A^{\pi_k}(x,u)\right],$$

which proves that π_{k+1} is the optimal solution to Problem (19). For the objective function of Problem (20), we have

$$\mathbb{E}_{x\sim d_0^{\pi_k},u\sim\pi_{k+1}}\left[\mathbb{I}[F^{\pi_k}(x) > 0]A_F^{\pi_k}(x,u)\right] = \mathbb{E}_{x\sim d_0^{\pi_k},u\sim\pi_{\text{out}}}\left[\mathbb{I}[F^{\pi_k}(x) > 0]A_F^{\pi_k}(x,u)\right],$$

which proves that π_{k+1} is the optimal solution to Problem (20). Thus, π_{k+1} is the optimal solution to both Problem (19) and (20). Since the original problem (11) is the summation of Problem (19) and (20), π_{k+1} is also the optimal solution to Problem (11).

As the optimal solution to Problem (19) and (20), π_{k+1} must be better than any other feasible solution to these two problems. Specifically, it must be better π_k. Since

$$\mathbb{E}_{u\sim\pi_k}\left[A^{\pi_k}(x,u)\right] = \mathbb{E}_{u\sim\pi_k}\left[A_F^{\pi_k}(x,u)\right] = 0, \forall x \in \mathcal{X},$$

we have

$$\begin{aligned}
&\mathbb{E}_{x\sim d^{\pi_k},u\sim\pi_{k+1}}\left[\mathbb{I}[F^{\pi_k}(x)>0]A^{\pi_k}(x,u)\right] \geq \mathbb{E}_{x\sim d^{\pi_k},u\sim\pi_k}\left[\mathbb{I}[F^{\pi_k}(x)>0]A^{\pi_k}(x,u)\right] = 0,\\
&\mathbb{E}_{x\sim d_0^{\pi_k},u\sim\pi_{k+1}}\left[\mathbb{I}[F^{\pi_k}(x)>0]A_F^{\pi_k}(x,u)\right] \leq \mathbb{E}_{x\sim d_0^{\pi_k},u\sim\pi_k}\left[\mathbb{I}[F^{\pi_k}(x)>0]A_F^{\pi_k}(x,u)\right] = 0.
\end{aligned}$$

By Theorem 2, we have

$$\begin{aligned}
F^{\pi_{k+1}}(x) \leq F^{\pi_k}(x) &+ \frac{1}{1-\gamma}\mathbb{E}_{x'\sim d_0^{\pi_k}(\cdot|x),u'\sim\pi_{k+1}}[A_F^{\pi_k}(x',u')]\\
&+ \frac{2\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}\mathbb{E}_{x'\sim d^{\pi_k}(\cdot|x)}[D_{TV}(\pi_{k+1}\|\pi_k)[x']].
\end{aligned} \tag{21}$$

For all $x \in \mathrm{X}^{\pi_k}$ and all $u \in \mathcal{U}$ such that $\pi_{k+1}(u|x) > 0$, we have

$$F^{\pi_k}(x) + A_F^{\pi_k}(x,u)/(1-\gamma) \leq 0 \Rightarrow A_F^{\pi_k}(x,u) \leq 0.$$

Take expectations inside the feasible region on both sides of (21),

$$\begin{aligned}
\mathbb{E}_{x\sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x)\leq 0]F^{\pi_{k+1}}(x)] &\leq \mathbb{E}_{x\sim d_{\text{init}}}\left[\mathbb{I}[F^{\pi_k}(x)\leq 0]F^{\pi_k}(x)\right]\\
&\quad + \mathbb{E}_{x\sim d_0^{\pi_k},u\sim\pi_{k+1}}[\mathbb{I}[F^{\pi_k}(x)\leq 0]A_F^{\pi_k}(x,u)]\\
&\quad + \frac{2\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}\mathbb{E}_{x\sim d^{\pi_k}}[\mathbb{I}[F^{\pi_k}(x)\leq 0]D_{TV}(\pi_{k+1}\|\pi_k)[x]]\\
&\leq \frac{2\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}\mathbb{E}_{x\sim d^{\pi_k}}[\mathbb{I}[F^{\pi_k}(x)\leq 0]D_{TV}(\pi_{k+1}\|\pi_k)[x]].
\end{aligned}$$

Using the relationship $D_{TV}(p\|q) \leq \sqrt{D_{KL}(p\|q)/2}$ and Jensen's inequality, we have

$$\begin{aligned}
\mathbb{E}_{x\sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x)\leq 0]F^{\pi_{k+1}}(x)] &\leq \frac{2\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}\mathbb{E}_{x\sim d^{\pi_k}}\left[\mathbb{I}[F^{\pi_k}(x)\leq 0]\sqrt{D_{KL}(\pi_{k+1}\|\pi_k)[x]/2}\right]\\
&\leq \frac{2\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}\sqrt{\mathbb{E}_{x\sim d^{\pi_k}}[\mathbb{I}[F^{\pi_k}(x)\leq 0]D_{KL}(\pi_{k+1}\|\pi_k)[x]]/2}\\
&\leq \frac{2\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}\cdot\sqrt{\delta/4}\\
&= \frac{\sqrt{\delta}\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2},
\end{aligned}$$

which proves the first inequality of the feasibility enhancement property.

Rearrange (21) and take expectations outside the feasible region, we have

$$\begin{aligned}
&\mathbb{E}_{x\sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x)>0](F^{\pi_{k+1}}(x) - F^{\pi_k}(x)]\\
\leq\ &\mathbb{E}_{x\sim d_0^{\pi_k},u\sim\pi_{k+1}}[\mathbb{I}[F^{\pi_k}(x)>0]A_F^{\pi_k}(x,u)]\\
&+ \frac{2\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}\mathbb{E}_{x\sim d^{\pi_k}}[\mathbb{I}[F^{\pi_k}(x)>0]D_{TV}(\pi_{k+1}\|\pi_k)[x]]\\
\leq\ &\frac{2\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}\mathbb{E}_{x\sim d^{\pi_k}}[\mathbb{I}[F^{\pi_k}(x)>0]D_{TV}(\pi_{k+1}\|\pi_k)[x]]\\
\leq\ &\frac{2\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}\sqrt{\mathbb{E}_{x\sim d^{\pi_k}}[\mathbb{I}[F^{\pi_k}(x)>0]D_{KL}(\pi_{k+1}\|\pi_k)[x]]/2}\\
\leq\ &\frac{\sqrt{\delta}\gamma\epsilon_F^{\pi_{k+1}}}{(1-\gamma)^2}.
\end{aligned}$$

This proves the second inequality of the feasibility enhancement property.
By Corollary 1 in the CPO paper [12], we have

$$V^{\pi_{k+1}}(x) - V^{\pi_k}(x) \geq \frac{1}{1-\gamma}\mathbb{E}_{x'\sim d^{\pi_k}(\cdot|x),u'\sim\pi_{k+1}}[A^{\pi_k}(x',u')] - \frac{2\gamma\epsilon^{\pi_{k+1}}}{(1-\gamma)^2}\mathbb{E}_{x'\sim d^{\pi_k}(\cdot|x)}[D_{TV}(\pi_{k+1}\|\pi_k)[x']]. \tag{22}$$

Take expectations inside the feasible region,

$$\begin{aligned}
&\mathbb{E}_{x\sim d_{\text{init}}}[\mathbb{I}[F^{\pi_k}(x)\leq 0](V^{\pi_{k+1}}(x) - V^{\pi_k}(x))] \\
\geq\ & \frac{1}{1-\gamma}\mathbb{E}_{x\sim d^{\pi_k}}[\mathbb{I}[F^{\pi_k}(x)\leq 0]A^{\pi_k}(x,u)] \\
&- \frac{2\gamma\epsilon^{\pi_{k+1}}}{(1-\gamma)^2}\mathbb{E}_{x\sim d^{\pi_k}}[\mathbb{I}[F^{\pi_k}(x)\leq 0]D_{TV}(\pi_{k+1}\|\pi_k)[x]] \\
\geq\ & -\frac{2\gamma\epsilon^{\pi_{k+1}}}{(1-\gamma)^2}\mathbb{E}_{x\sim d^{\pi_k}}[\mathbb{I}[F^{\pi_k}(x)\leq 0]D_{TV}(\pi_{k+1}\|\pi_k)[x]] \\
\geq\ & -\frac{\sqrt{\delta}\gamma\epsilon^{\pi_{k+1}}}{(1-\gamma)^2}.
\end{aligned}$$

This proves the value improvement property and thus finishes the proof. □

B Practical implementation

B.1 Derivation of GAE of CDF

For a given trajectory $x_1, x_2, x_3, \ldots$, define

$$\delta_{F,t} = c_t + (1-c_t)\gamma F(x_{t+1}) - F(x_t).$$

Consider the multi-step TD errors of the CDF up to k steps:

$$\begin{aligned}
A_F^{(1)} &= \delta_{F,t} \\
&= -F(x_t) + c_t + (1-c_t)\gamma F(x_{t+1}), \\
A_F^{(2)} &= \delta_{F,t} + (1-c_t)\gamma\delta_{F,t+1} \\
&= c_t + (1-c_t)\gamma(c_{t+1} + (1-c_{t+1})\gamma F(x_{t+2})) - F(x_t) \\
&= -F(x_t) + c_t + (1-c_t)c_{t+1}\gamma + (1-c_t)(1-c_{t+1})\gamma^2 F(x_{t+2}), \\
&\vdots \\
A_F^{(k)} &= \sum_{l=0}^{k-1}\gamma^l\prod_{s=0}^{l-1}(1-c_{t+s})\,\delta_{F,t+l} \\
&= -F(x_t) + c_t + (1-c_t)c_{t+1}\gamma + (1-c_t)(1-c_{t+1})c_{t+2}\gamma^2 + \ldots \\
&\quad + \prod_{s=0}^{k-2}(1-c_{t+s})\,c_{t+k-1}\gamma^{k-1} + \prod_{s=0}^{k-1}(1-c_{t+s})\,\gamma^k F(x_{t+k})
\end{aligned}$$

The GAE of the CDF is the exponentially-weighted average of these k-step TD errors:

$$
\begin{aligned}
A_F &= (1-\lambda)\left(A_F^{(1)} + \lambda A_F^{(2)} + \lambda^2 A_F^{(3)} + \dots\right) \\
&= (1-\lambda)\Bigg(\delta_{F,t} + \lambda(\delta_{F,t} + (1-c_t)\gamma\delta_{F,t+1}) + \lambda^2(\delta_{F,t} + (1-c_t)\gamma\delta_{F,t+1} \\
&\quad + (1-c_t)(1-c_{t+1})\gamma^2\delta_{F,t+2} + \dots)\Bigg) \\
&= (1-\lambda)\Bigg(\left(1+\lambda+\lambda^2+\dots\right)\delta_{F,t} + \lambda\gamma(1-c_t)\left(1+\lambda+\lambda^2+\dots\right)\delta_{F,t+1} \\
&\quad + (\lambda\gamma)^2(1-c_t)(1-c_{t+1})\left(1+\lambda+\lambda^2+\dots\right)\delta_{F,t+2} + \dots\Bigg) \\
&= (1-\lambda)\left(\frac{1}{1-\lambda}\delta_{F,t} + (1-c_t)\frac{\lambda\gamma}{1-\lambda}\delta_{F,t+1} + (1-c_t)(1-c_{t+1})\frac{(\lambda\gamma)^2}{1-\lambda}\delta_{F,t+2} + \dots\right) \\
&= \sum_{l=0}^{\infty}(\lambda\gamma)^l \prod_{s=0}^{l-1}(1-c_{t+s})\delta_{F,t+l}.
\end{aligned}
$$

B.2 Pseudocode

Algorithm 1: Feasible policy optimization (FPO)

Initialize : Network parameters ϕ, ω, θ.

for *each epoch* **do**
  `// Sample data`
  for *each sample step* **do**
    Sample action $u \sim \pi_\theta(\cdot|x)$;
    Get next state x', reward r, and indicator for constraint violation c from environment;
  end
  Compute GAEs of return and cost along sampled trajectories;
  `// Update networks`
  for *each update step* **do**
    Update feasibility network $\phi \leftarrow \phi - \eta\nabla_\phi L_F(\phi)$; `// Equation (15)`
    Update value network $\omega \leftarrow \omega - \eta\nabla_\omega L_V(\omega)$; `// Equation (16)`
    Update policy network $\theta \leftarrow \theta - \eta\nabla_\theta L_\pi(\theta)$; `// Equation (17)`
  end
end

C Experiments

The Safety-Gymnasium benchmark [31] and the Omnisafe toolbox [32] are both released under the Apache License 2.0.

All experiments are conducted on a workstation equipped with Intel(R) Xeon(R) Gold 6246R CPUs (32 cores, 64 threads), an NVIDIA GeForce RTX 3090 GPU, and 256GB of RAM. A single experimental trial—comprising one environment, one algorithm, and one random seed—takes about 2 hours to execute. Executing all experiments with a properly configured concurrent running scheme requires approximately 400 hours.

C.1 Hyperparameters

Table 1: Hyperparameters

Category	Hyperparameter	Value
Shared	Number of vector environments	20
	Steps per epoch	20000
	Batch size	20000 for navigation tasks 4000 for velocity tasks
	Reward discount factor	0.99
	Cost discount factor	0.95
	Cost limit	0
	GAE λ	0.95
	Actor learning rate	3e-5 for PointCircle 3e-4 for CarCircle, Ant, HalfCheetah, Hopper, and Walker2d 1e-4 for others
	Actor learning rate schedule	linear decay to 0
	Actor network hidden sizes	(64, 64)
	Actor activation function	Tanh
	Critic learning rate	3e-4
	Critic network hidden sizes	(64, 64)
	Critic activation function	Tanh
	Network weight initialization method	Kaiming uniform
	Optimizer	Adam
	Entropy coefficient	0.01 for Hopper and Walker2d 0 for others
	Critic norm coefficient	0.001
	Target KL divergence	0.02
	Maximum gradient norm	40
PPO	IS ratio clip	0.2
Lagrangian	Initial multiplier	0.001
	Multiplier learning rate	0.035
FPO	Feasibility threshold ϵ	0.1
	Maximum weight $w_{\max}$	10
	Feasibility ratio margin α	1.2

C.2 Training curves

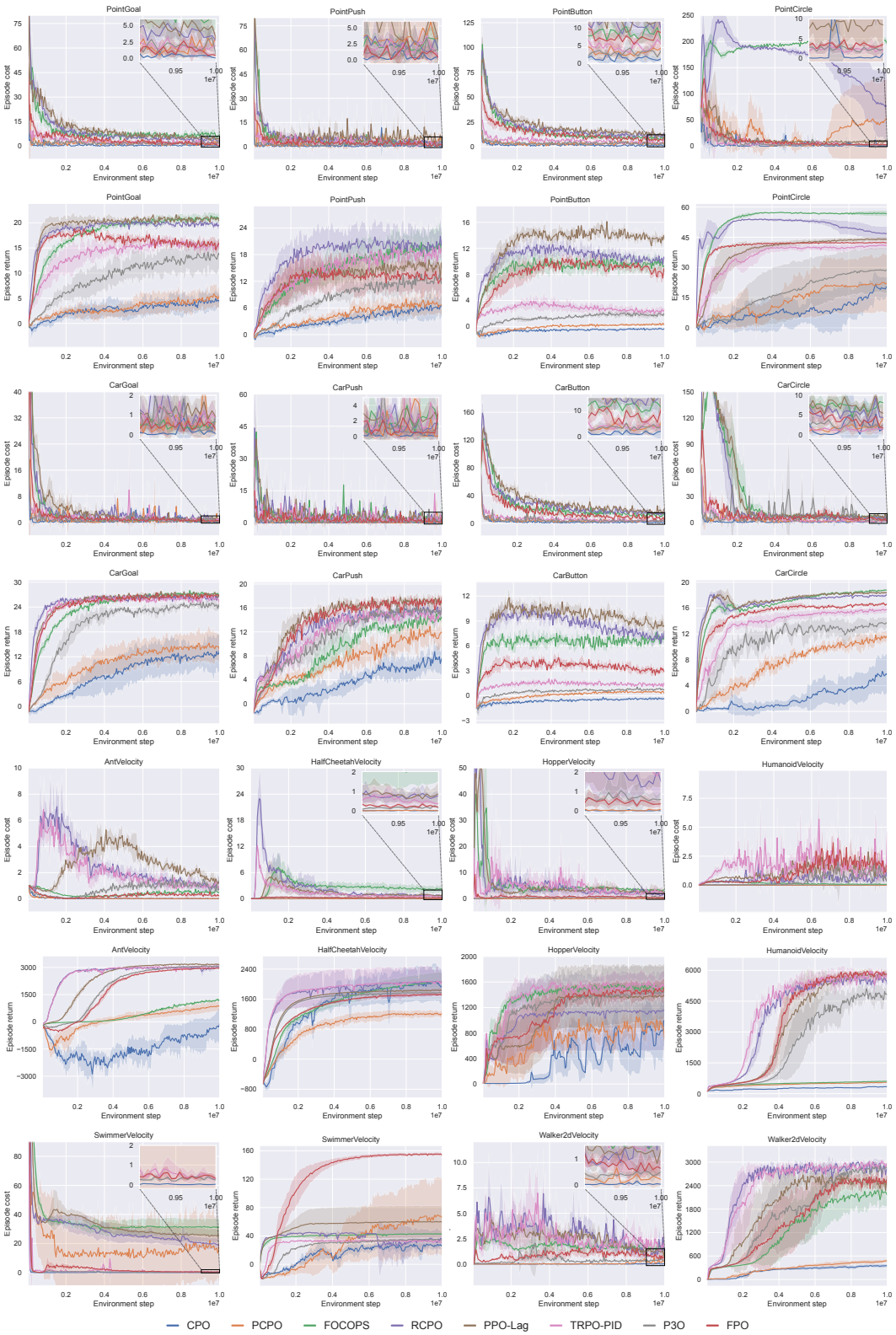


Figure 6: Training curves on all 14 environments. Rows form cost-return pairs for groups of four environments; the last pair places SwimmerVelocity and Walker2dVelocity side by side. Curves are five-seed means; shading is mean $\pm\ 1.96s/\sqrt{5}$ for across-seed sample standard deviation s.

C.3 Final scores

Table 2: Average cost and return in the last 10% iterations

	AntVelocity		CarButton		CarCircle	
Algorithm	Cost	Return	Cost	Return	Cost	Return
CPO	0.01 ± 0.01	-460.19 ± 740.62	1.22 ± 0.38	-0.38 ± 0.18	1.80 ± 2.51	5.30 ± 2.65
PCPO	0.01 ± 0.00	823.17 ± 292.59	3.52 ± 0.42	0.42 ± 0.09	1.78 ± 0.88	11.14 ± 1.01
FOCOPS	0.53 ± 0.07	1147.42 ± 69.22	11.83 ± 0.67	6.78 ± 0.22	6.98 ± 1.12	18.68 ± 0.24
RCPO	1.00 ± 0.46	2942.00 ± 145.65	15.00 ± 1.72	7.13 ± 0.41	5.48 ± 0.80	17.89 ± 0.20
PPO-Lag	1.46 ± 0.32	3163.00 ± 45.47	16.49 ± 1.99	8.56 ± 0.58	6.86 ± 1.14	18.34 ± 0.25
TRPO-PID	0.97 ± 0.20	3050.21 ± 37.43	3.96 ± 0.45	1.35 ± 0.09	1.26 ± 0.43	15.66 ± 0.60
P3O	0.98 ± 0.17	3065.16 ± 18.30	3.48 ± 1.14	0.76 ± 0.09	2.55 ± 0.38	13.46 ± 1.05
FPO	0.29 ± 0.05	2950.89 ± 40.78	7.54 ± 0.69	3.06 ± 0.19	3.73 ± 0.99	16.55 ± 0.24
	CarGoal		CarPush		HalfCheetahVelocity	
Algorithm	Cost	Return	Cost	Return	Cost	Return
CPO	0.15 ± 0.13	12.52 ± 4.26	0.31 ± 0.36	7.31 ± 1.96	0.01 ± 0.01	2008.63 ± 478.16
PCPO	0.55 ± 0.29	14.43 ± 3.71	1.10 ± 0.79	11.56 ± 2.02	0.01 ± 0.01	1201.48 ± 97.06
FOCOPS	0.47 ± 0.04	27.18 ± 0.16	1.74 ± 0.88	14.13 ± 0.41	2.18 ± 0.88	2047.42 ± 257.71
RCPO	1.10 ± 0.19	26.24 ± 0.33	1.98 ± 0.75	15.52 ± 1.35	0.75 ± 0.31	2059.44 ± 394.40
PPO-Lag	1.03 ± 0.39	27.02 ± 0.12	2.30 ± 0.25	17.08 ± 0.64	0.83 ± 0.43	1831.80 ± 63.95
TRPO-PID	0.81 ± 0.22	26.07 ± 0.26	2.27 ± 1.48	14.78 ± 1.01	0.56 ± 0.21	2057.14 ± 390.16
P3O	0.74 ± 0.41	24.41 ± 0.82	1.01 ± 0.91	15.50 ± 0.71	0.16 ± 0.03	1770.94 ± 25.48
FPO	0.48 ± 0.09	26.86 ± 0.14	0.89 ± 0.16	17.02 ± 0.66	0.25 ± 0.02	1723.63 ± 51.23
	HopperVelocity		HumanoidVelocity		PointButton	
Algorithm	Cost	Return	Cost	Return	Cost	Return
CPO	0.01 ± 0.01	808.94 ± 258.62	0.01 ± 0.00	334.46 ± 39.26	1.29 ± 0.25	-0.37 ± 0.05
PCPO	0.03 ± 0.05	923.05 ± 77.47	0.00 ± 0.00	539.65 ± 25.04	3.22 ± 0.74	0.32 ± 0.20
FOCOPS	3.31 ± 0.62	1502.24 ± 63.07	0.08 ± 0.01	594.57 ± 35.36	8.69 ± 1.31	9.63 ± 1.18
RCPO	1.97 ± 1.07	1139.05 ± 618.61	0.63 ± 0.19	5555.85 ± 218.92	11.52 ± 0.78	10.24 ± 0.34
PPO-Lag	2.89 ± 1.70	1376.34 ± 501.10	1.43 ± 0.29	5742.21 ± 250.88	12.38 ± 0.61	13.43 ± 0.37
TRPO-PID	2.57 ± 1.61	1531.96 ± 183.29	1.72 ± 0.86	5706.46 ± 197.10	4.76 ± 0.58	2.34 ± 0.62
P3O	0.66 ± 0.54	1429.71 ± 430.99	1.05 ± 0.75	4792.15 ± 400.03	3.02 ± 0.51	1.79 ± 0.42
FPO	0.44 ± 0.21	1462.41 ± 150.27	1.83 ± 0.47	5842.85 ± 75.03	7.36 ± 1.12	8.48 ± 0.74
	PointCircle		PointGoal		PointPush	
Algorithm	Cost	Return	Cost	Return	Cost	Return
CPO	1.26 ± 2.18	19.09 ± 6.21	0.39 ± 0.23	4.41 ± 1.96	0.48 ± 0.29	5.97 ± 2.77
PCPO	46.57 ± 78.77	21.76 ± 14.25	1.78 ± 0.72	5.07 ± 2.09	3.08 ± 3.15	6.61 ± 2.72
FOCOPS	200.57 ± 7.47	56.83 ± 1.50	6.99 ± 1.99	20.87 ± 1.06	2.22 ± 0.57	19.42 ± 4.26
RCPO	85.71 ± 60.89	47.23 ± 3.35	4.12 ± 0.62	19.77 ± 0.31	2.78 ± 1.41	19.84 ± 4.02
PPO-Lag	8.52 ± 3.72	43.95 ± 1.39	5.03 ± 0.99	20.74 ± 0.20	5.29 ± 0.39	15.19 ± 1.32
TRPO-PID	2.43 ± 0.96	40.84 ± 2.10	1.82 ± 0.39	15.24 ± 1.08	1.88 ± 0.47	18.01 ± 0.98
P3O	2.42 ± 0.80	28.54 ± 11.22	1.84 ± 0.88	13.29 ± 3.32	1.31 ± 0.84	12.28 ± 3.40
FPO	3.32 ± 0.09	42.30 ± 0.33	1.26 ± 0.07	15.53 ± 0.96	1.10 ± 0.40	12.86 ± 3.78
	SwimmerVelocity		Walker2dVelocity			
Algorithm	Cost	Return	Cost	Return		
CPO	0.02 ± 0.01	26.99 ± 6.05	0.02 ± 0.02	348.08 ± 48.28		
PCPO	17.71 ± 24.39	66.14 ± 53.32	0.22 ± 0.21	454.43 ± 54.78		
FOCOPS	31.33 ± 6.06	42.82 ± 1.45	1.56 ± 0.48	2200.94 ± 324.12		
RCPO	19.86 ± 4.13	34.31 ± 14.90	1.26 ± 0.25	2849.69 ± 109.22		
PPO-Lag	25.53 ± 4.87	59.85 ± 22.59	1.28 ± 0.53	2489.10 ± 196.43		
TRPO-PID	0.54 ± 0.23	32.46 ± 2.75	1.84 ± 1.11	2927.93 ± 55.02		
P3O	0.29 ± 0.01	34.92 ± 1.56	0.39 ± 0.06	2823.77 ± 145.66		
FPO	0.43 ± 0.14	155.00 ± 2.44	0.80 ± 0.28	2508.45 ± 216.26		

C.4 Sensitivity analysis of the feasibility threshold

We evaluate the sensitivity of the feasibility threshold ϵ across four settings ($\epsilon = 0.01, 0.05, 0.1, 0.2$), as shown in Table 3. The results indicate that for $\epsilon = 0.05$ and $\epsilon = 0.1$, both the average cost and return fluctuate within a narrow and acceptable range. This suggests that the algorithm is robust to the specific choice of ϵ within this interval. In these cases, FPO effectively balances the trade-off between exploring for higher rewards and adhering to safety constraints. When $\epsilon = 0.01$, the smaller estimated feasible region reduces cost in several tasks but can make learning overly conservative, as in AntVelocity and HopperVelocity, or destabilize the cost-return trade-off, as in SwimmerVelocity. However, when ϵ is increased to 0.2, we observe degradation in safety performance in several

environments. This is particularly evident in the PointCircle task, where the average cost spikes significantly. Such an excessively large threshold incorrectly labels critical states as feasible, leading to insufficient safety updates and a breakdown of safety guarantees. In conclusion, FPO is robust within a moderate range (e.g., 0.05 to 0.1), whereas either an overly small or overly large threshold can degrade performance.

Table 3: Sensitivity analysis of the feasibility threshold ϵ across different environments.

	$\epsilon = 0.01$		$\epsilon = 0.05$		$\epsilon = 0.1$		$\epsilon = 0.2$	
Environment	Cost	Return	Cost	Return	Cost	Return	Cost	Return
PointCircle	0.31	40.86	3.28	42.33	3.32	42.30	102.01	46.24
PointButton	3.49	1.64	5.88	6.15	7.36	8.48	8.91	10.20
CarGoal	0.76	26.08	0.40	26.80	0.48	26.86	0.82	26.85
CarPush	0.42	14.51	0.79	17.27	0.89	17.02	1.75	17.33
AntVelocity	0.03	-18.42	0.27	2834.59	0.29	2950.89	0.82	2915.30
HalfCheetahVelocity	0.08	1113.59	0.46	1757.84	0.25	1723.63	0.86	1757.84
HopperVelocity	0.01	897.62	0.43	1526.56	0.44	1462.41	1.51	1663.16
SwimmerVelocity	28.33	35.11	0.58	155.34	0.43	155.00	0.69	157.94

C.5 Additional ablations

Unless otherwise stated, the entries below are point estimates of cost and return averaged over the final 10% of training epochs.

State-dependent weight We compare the proposed smooth barrier-like weight with a linear alternative $w(x) = Cr(x)$. The linear alternative weakens constraint satisfaction in most environments and substantially reduces the return in HopperVelocity, supporting a smooth penalty that grows more sharply near the estimated feasible-region boundary.

Table 4: Ablation of the state-dependent weight. Each entry is cost/return.

Environment	Proposed $w(x)$	Linear $w(x)$
PointCircle	3.32/42.30	28.82/32.28
PointButton	7.36/8.48	17.36/16.49
CarGoal	0.48/26.86	0.58/26.53
CarPush	0.89/17.02	3.45/16.92
AntVelocity	0.29/2950.89	2.30/3131.60
HalfCheetahVelocity	0.25/1723.63	1.71/1815.63
HopperVelocity	0.44/1462.41	3.86/329.68
SwimmerVelocity	0.43/155.00	1.51/156.77

CDF versus CVF We replace the first-violation CDF with a standard cumulative cost value function (CVF). The CVF variant generally incurs higher cost and can substantially reduce return in HopperVelocity and SwimmerVelocity. CarGoal is the only task in which it improves both metrics, possibly because rare violations make positive first-violation targets sparse while cumulative costs provide a denser signal. More generally, CDF matches problems in which entering an infeasible region is the primary concern, whereas CVF may be preferable when cumulative violation is the relevant constraint semantics.

Cost limits for a Lagrangian baseline To test whether a zero cost limit unfairly disadvantages Lagrangian methods, we evaluate TRPO-PID with limits 0, 5, and 10 on two representative environments. Increasing the limit moves TRPO-PID along the cost-return frontier: return increases by permitting more violations, but cost and return do not improve simultaneously. The final multipliers are approximately 30, 15, and 10 for limits 0, 5, and 10, respectively, and increase gradually rather than diverging.

Table 5: CDF-CVF ablation. Each entry is cost/return.

Environment	FPO with CDF	FPO with CVF
PointCircle	3.32/42.30	22.93/47.02
PointButton	7.36/8.48	14.35/15.16
CarGoal	0.48/26.86	0.34/26.91
CarPush	0.89/17.02	2.17/17.39
AntVelocity	0.29/2950.89	1.47/3052.22
HalfCheetahVelocity	0.25/1723.63	0.43/1867.51
HopperVelocity	0.44/1462.41	3.34/1389.96
SwimmerVelocity	0.43/155.00	0.24/35.34

Table 6: TRPO-PID cost-limit ablation. Each entry is cost/return.

Environment	TRPO-PID (0)	TRPO-PID (5)	TRPO-PID (10)	FPO
PointButton	4.76/2.34	5.33/4.07	9.49/8.67	7.36/8.48
CarButton	3.96/1.35	6.07/2.52	12.07/4.21	7.54/3.06

C.6 Evaluation with visual observations

We additionally evaluate whether FPO can approximate a feasibility boundary from high-dimensional observations in SafetyCarRace1Vision-v0, a visual safe-control task based on CarRace. The agent receives an $84 \times 84 \times 3$ first-person RGB image and a 63-dimensional proprioceptive and navigation vector, without pretrained visual representations. The visual encoder uses three convolutional layers with channels $(32, 64, 64)$, 3×3 kernels, stride 2, padding 1, and ReLU activations, followed by a 512-dimensional projection. The vector branch is a $(64, 64)$ Tanh MLP; separate encoders are used for the actor, reward critic, and feasibility/cost critic. Figure 7 shows the training curves. Over the final 10% of training epochs, FPO obtains return 14.97 and cost 1.99, while PPO-Lag obtains return 15.06 and cost 3.17 using the same architecture. In terms of return, FPO learns more slowly early in training, but reaches performance comparable to PPO-Lag later on. In terms of cost, PPO-Lag incurs a large cost peak early in training, while FPO suppresses violations much more quickly and keeps the cost low afterwards. These results provide preliminary evidence, rather than a general guarantee, that FPO can learn useful feasibility estimates from multimodal visual observations.

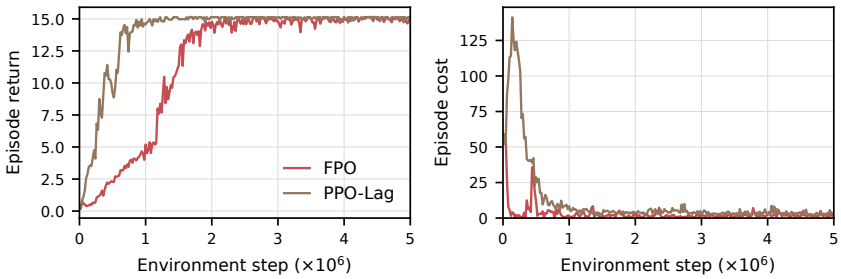


Figure 7: Training curves of FPO and PPO-Lag in SafetyCarRace1Vision-v0. Curves show a single seed without smoothing.

D Broader impacts

While our proposed method is not specifically designed for any downstream application, progress in reinforcement learning (RL) can have broad societal implications. On one hand, improved RL algorithms could enhance safety and efficiency in decision-making across fields like healthcare, robotics, education, and climate management, benefiting society as a whole. On the other hand, these techniques could also be exploited in harmful ways, such as invasive surveillance, deceptive recommendation systems, or exploitative high-frequency trading—particularly if deployed without proper safeguards or ethical oversight. Although this study does not directly tackle these applications, we

recognize the dual-use nature of RL advancements and stress the need for responsible implementation, along with further research into transparency, safety, and fairness in RL systems.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Claims are made in the abstract and the last paragraph of the introduction. They accurately reflect the paper's scope and contributions detailed in Sections 4 and 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Technical limitations concerning sample efficiency, the handcrafted weight, and initial safe exploration are discussed in the Limitations paragraph following the conclusion.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The proofs of all theorems and corollaries are provided either in the main paper or the appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We describe our proposed algorithm in Section 4.4 and provide the pseudocode, hyperparameters, and additional ablations in Appendices B.2, C.1, and C.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

Answer: [Yes]

Justification: We provide our code in the supplemental material, with instructions for reproducing the experimental results.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Implementation details are described in Section 4.4; hyperparameters and additional settings are provided in Appendices C.1, C.5, and C.6.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: The training curves in Section 5 show normal-approximation intervals over five independent seeds, with the calculation stated in the figure captions. The normalized aggregate plot separately reports its across-environment dispersion measure.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Information on computer resources, including CPU, GPU, memory, and computing time are provided in Appendix C.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have carefully reviewed the NeurIPS Code of Ethics and confirm that our research conforms with it in every respect.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss both potential positive and negative societal impacts in Appendix D.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite the original paper of the code used in the paper, and mention the license in Appendix C.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We provide instructions for running our code in the supplemental material.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

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