Proactive Constrained Policy Optimization with Preemptive Penalty

Ning Yang[1*], **Pengyu Wang**[1, 2], **Guoqing Liu**[3], **Haifeng Zhang**[1], **Pin Lyu**[1], **Jun Wang**[4]
[1]Institute of Automation, Chinese Academy of Sciences
[2]School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), Longgang, Shenzhen, Guangdong, 518172, P.R. China
[3]Microsoft Research
[4]University College London
ning.yang@ia.ac.cn

Abstract

Safe Reinforcement Learning (RL) often faces significant issues such as constraint violations and instability, necessitating the use of constrained policy optimization, which seeks optimal policies while ensuring adherence to specific constraints like safety. Typically, constrained optimization problems are addressed by the Lagrangian method, a post-violation remedial approach that may result in oscillations and overshoots. Motivated by this, we propose a novel method named Proactive Constrained Policy Optimization (PCPO) that incorporates a preemptive penalty mechanism. This mechanism integrates barrier items into the objective function as the policy nears the boundary, imposing a cost. Meanwhile, we introduce a constraint-aware intrinsic reward to guide boundary-aware exploration, which is activated only when the policy approaches the constraint boundary. We establish theoretical upper and lower bounds for the duality gap and the performance of the PCPO update, shedding light on the method's convergence characteristics. Additionally, to enhance the optimization performance, we adopt a policy iteration approach. An interesting finding is that PCPO demonstrates significant stability in experiments. Experimental results indicate that the PCPO framework provides a robust solution for policy optimization under constraints, with important implications for future research and practical applications.

Code — https://github.com/gracefulning/PCPO
Extended Appendix —
https://github.com/gracefulning/PCPO

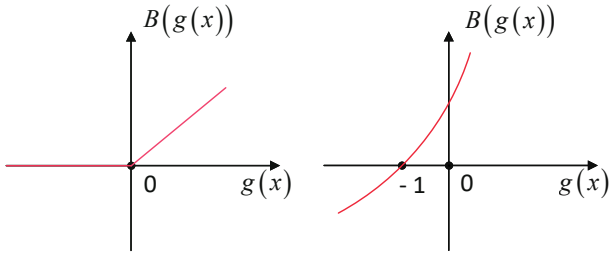
Introduction

Security learning plays a crucial role in the field of computer science, as it is essential for ensuring data privacy, withstanding adversarial attacks, enhancing the security of intelligent systems, and meeting compliance requirements. Recently, this field has seen substantial advancements in tackling complex challenges such as safe exploration (Dalal et al. 2018), the application of Lyapunov methods (Chow et al. 2018b; Hao et al. 2024), and constrained optimization (Tessler, Mankowitz, and Mannor 2018; Gu et al. 2024b; Zhang et al. 2024). Despite these advancements, the dynamic nature of network security threats, coupled with the vast scale and complexity of computer systems, continues to pose significant challenges.

In the realm of safe Reinforcement Learning (RL), also known as constrained RL, the Lagrangian method has emerged as a predominant tool (Ghosh 2025). Its simplicity and effectiveness in achieving optimal constraint satisfaction (Chow et al. 2019; Tessler, Mankowitz, and Mannor 2018). Nonetheless, this approach faces several challenges that hinder its application in constrained optimization problems. One issue is that the control of Lagrange multipliers exhibits hysteresis, implying that when the constraints are violated, the Lagrange multipliers cannot instantaneously transition to an expected value (Stooke, Achiam, and Abbeel 2020; Honari et al. 2024). Another challenge is that the Lagrangian method cannot serve as a barrier near the boundary of the constraints. In other words, when a policy approaches the constraint boundary, there is no penalty or change in gradient. This limitation of the method is demonstrated in Figure 1, which illustrates the relationship between constraint violation value $g(x)$ and Lagrange multiplier or barrier items $B(g(x))$. Figure 1(a) uses the Lagrange multiplier method to demonstrate that the penalty value $B(g(x)) = 0$ when $g(x) \leq 0$ (constraint satisfied); when $g(x) > 0$ (constraint violated), the penalty value $B(g(x)) > 0$. Figure 1(b) describes the extended barrier function method, where for $-1 < g(x) < 0$, the barrier term $B(g(x)) > 0$, demonstrating the *preemptive penalty* mechanism adopted by the barrier function method.

Considering the intricate nature of constraint management in optimization problems and the shortcomings of traditional Lagrangian methods, we introduce a preemptive penalty to mitigate constraint violations and enhance optimization performance. We propose the Proactive Constrained Policy Optimization (PCPO) with a preemptive penalty. The advantages of our method are as follows: (1) It exhibits a strictly positive gradient, which increases as a satisfied constraint approaches violation during optimization. This pushes the constraint back towards the achievable range. (2) Another crucial advantage is that the derivatives of our method provide the implicit dual variables, ensuring duality-gap guarantees. Our contributions in this paper are outlined as fol-

*Correspondence to: Ning Yang



(a) Lagrange multiplier (b) Log-barrier extension

Figure 1: Lagrange and barrier terms vs. constraint violations.

lows:

- *Methodology*: Within the PCPO approach, we address constraint violations by integrating a preemptive penalty, embedding the barrier term directly into the objective function. This approach proactively penalizes potential constraint breaches, circumventing the issue of zero gradients. Additionally, we introduce a constraint-aware intrinsic reward to guide boundary-aware exploration, capturing the potential impact degree of each sample on the strategy relative to the constraint boundary.
- *Theoretical Framework*: Our analysis includes a derivation of the approximate Kullback-Leibler (KL) divergence for the Gaussian model. We establish a theoretical upper bound on the duality gap, alongside a lower bound on the performance improvements following a PCPO update, ensuring a robust and stable theoretical foundation for our methodologies.
- *Experiment Results*: Our results validate that the PCPO method effectively elevates the quality of solutions and markedly diminishes the incidence of constraint violations. Furthermore, the proposed method has been verified in terms of generalization and sensitivity analysis.

Related Work

Safe RL is a crucial branch of RL aimed at addressing constrained optimization or control problems, incorporating a variety of specialized methods to ensure safety during learning (Gu et al. 2024b). Primal methods directly tackled the primal problem by alternating optimizing between the objective function and constraints, which could lead to instability in network training (Wachi, Shen, and Sui 2024). The safeguard/safety layer approach, which included the projection method (Yang et al. 2020), may face challenges in finding suitable projections. Penalty methods attempted to manage constraints through reward shaping or regularization, and choosing the right penalty factor remained difficult (Tessler, Mankowitz, and Mannor 2018). Barrier-based methods enforced safety through barrier regularization, ensuring feasibility within the optimization process (Liu, Ding, and Liu 2020; Wang et al. 2023). Direct policy optimization leveraged a surrogate function, but this approach was susceptible to approximation errors (Chow et al. 2019).

Primal-dual methods, widely utilized in safe RL, are extensively developed to approximate the primal problem by optimizing the dual problem. However, (Chow et al. 2018a) highlighted that updates to dual variables do not ensure immediate constraint satisfaction. Enhancing this approach, (Achiam et al. 2017) resets dual variables in each iteration to ensure consistent adherence to constraints. (Wagener, Boots, and Cheng 2021) adopted Lagrangian dual gradient ascent for updating Lagrange multipliers, thus enhancing model adaptability. (Ding and Lavaei 2023) proposed a periodically restarted optimistic primal-dual Proximal Policy Optimization (PPO) to effectively address time-varying constraints. (Ying et al. 2024) investigated scalable primal-dual methods in safe multi-agent RL, broadening the applicability of these techniques to more complex systems. Expanding the scope, (Chen et al. 2024) proposed an adaptive primal-dual method, attempting to solve the dual problem in safe RL by adjusting two adaptive learning processes to Lagrange multipliers.

Meanwhile, constrained intrinsic rewards in safe RL have also garnered extensive research attention (Gu et al. 2024a). To address the issue of introducing safety constraints, (Kwon et al. 2024) proposed a constrained reward framework to balance the performance objective of the total reward and the safety constraints. Furthermore, (Li, Zhu, and Grossklags 2025) uses the metric distance between the current optimal policy and the theoretically optimal policy as an intrinsic reward, and optimizes the safety strategy through reward shaping. However, despite these advancements, the typical application of penalties post-violation in these methods could still lead to potential network instability (Ibrahim et al. 2024; Su et al. 2025), prompting further exploration into preemptive penalty mechanisms to enhance safety measures.

Constrained Markov Decision Process

A Constrained Markov Decision Process (CMDP) (Altman 2021) is denoted by a tuple $\langle \mathcal{S}, \mathcal{A}, P, \nu, R, C \rangle$, where $\mathcal{S}$ represents the state space and $\mathcal{A}$ denotes the action space, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a probabilistic transition process with components $P_\pi(s'|s) \in \mathbf{R}$, $\nu : \mathcal{S} \to [0, 1]$ is a probability distribution over initial states, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, and set C contains a set of cost functions $C_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}, i = 1, 2, ..., m$. A policy π is a mapping from states to distributions over actions $\mathcal{S} \times \mathcal{A} \to [0, 1]$, and Π is a set of policies. Our goal is to discover a stationary policy that maximizes the expected discounted reward $J(\pi) := \mathbf{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$ under constraints, where the agent's actions dictate the subsequent rewards and states. Considering the initial state s_0, the state-value function is $V^\pi(s) := \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)|s_0 = s\right]$. The action-value function (Silver 2015) can similarly be decomposed $Q^\pi(s, a) := \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)|s_0 = s, a_0 = a\right]$. The advantage function (Mnih et al. 2016) is $A^\pi(s, a) := Q^\pi(s, a) - V^\pi(s)$. The discounted probabilistic transition distribution (Achiam et al. 2017) is denoted by $d_\pi(s) := (1-\gamma) \sum_{t=0}^{\infty} \gamma^t p_\pi^t(s)$ where $p_\pi^t(s) = P(s_t = s|\pi)$ is a vector with $p_\pi^t(s) \in \mathbb{R}$. The C_i-return, defined as the expected cumulative discounted cost for cost function i, is given by $J_{C_i}(\pi) := \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t C_i(s_t, a_t)]$ where $C_i(s_t, a_t)$ repre-

sents the cost incurred at state s_t and action a_t under cost function i. The set of feasible policies is $\Pi_C := \{\pi \in \Pi : J_{C_i}(\pi) \leq d_i, i = 1, 2, ..., m\}$. The optimal policy of the CMDP is given by $\pi^* = \arg\max_{\pi \in \Pi_C} J(\pi)$.

Similar to the value function V^π, action-value function Q^π, and advantage function A^π for reward, we denote the cost value function, cost action-value function, and cost advantage function as $V^\pi_{C_i}$, $Q^\pi_{C_i}$, and $A^\pi_{C_i}$.

Proactive Constrained Policy Optimization

Building on this foundation, the implementation of PCPO with a preemptive penalty can be summarized in two steps:

1. *Design of Constraint-Aware Intrinsic Reward:* A constraint-aware intrinsic reward is introduced to guide boundary-aware exploration, which is activated only when the policy approaches the constraint boundary and encourages behaviors that remain within the feasible region.
2. *Optimization with Preemptive Penalty:* The PCPO finds the constrained updated policy with an extended barrier function method featuring a preemptive penalty mechanism that adopts a conservative policy iteration.

Problem Formulation

Initially, a constrained optimization problem is formulated. Samples are then drawn from the environment to update policies, considering a set of parameterized policy spaces Π_θ with a fixed neural network. At the k-th iteration, the policy π_{θ_k} is updated to $\pi_{\theta_{k+1}}$ under constraints. The constrained objective function is defined as follows:

$$\max_{\pi_\theta \in \Pi_\theta} \quad \mathbb{E}_{s\sim d_{\pi_{\theta_k}}, a\sim\pi_\theta}[A^{\pi_{\theta_k}}(s,a)] \tag{1a}$$

$$\text{s.}t. \quad J_{C_i}(\pi_{\theta_k}) + \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi_{\theta_k}} \\ a\sim\pi_\theta}} \left[A^{\pi_{\theta_k}}_{C_i}(s,a)\right] \leq d_i,$$

$$i = 1, 2, \ldots, m \tag{1b}$$

$$\mathop{\mathbb{E}}_{s\sim d_{\pi_{\theta_k}}} [D_{KL}(\pi_\theta(.|s)||\pi_{\theta_k}(.|s))] \leq \delta, \tag{1c}$$

where δ is a small constant. The objective function, as defined in (1a), incorporates the advantage function in its expectation form, which provides a more accurate estimate of actions than the Q-function. Constraint (1b) ensures compliance with the necessary conditions, while Constraint (1c), a trust region constraint, guarantees monotonic performance improvements by ensuring a bounded policy update and a non-negative expected advantage at each state.

Optimization with Preemptive Penalty Method

A policy is employed in the previous subsection to enhance the quality of optimization. Building on this approach, this subsection focuses on reducing constraint violations through the use of constrained policy optimization, a type of policy search algorithm designed for CMDP. This algorithm includes updates that approximately solve constrained optimization problems, effectively minimizing potential penalties. Further drawing from the concept of barrier functions, the preemptive penalty method is employed to ensure that solutions remain within the feasible region. We propose the PCPO with a preemptive penalty by embedding barrier terms into the objective function (Kervadec et al. 2022). As the policy nears the boundary of the constraints, the preemptive penalty mechanism automatically activates, effectively reducing violations by proactively preventing them before they occur. The mathematical formulations for the simplified objective function (1a) and Constraints (1b) are as follows:

$$f(\pi_\theta) = \mathbb{E}_{s\sim d_{\pi_{\theta_k}}, a\sim\pi_\theta}\left[A^{\pi_{\theta_k}}(s,a)\right] \tag{2}$$

$$g_{C_i}(\pi_\theta) = J_{C_i}(\pi_{\theta_k}) + \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi_{\theta_k}} \\ a\sim\pi_\theta}} \left[A^{\pi_{\theta_k}}_{C_i}(s,a)\right] - d_i$$

$$\leq 0. \tag{3}$$

To solve this constrained problem, we use a barrier function approach. Though this is not the standard Lagrangian dual formulation, it can be interpreted as an approximation that implicitly defines dual variables. We formulate the barrier problem as follows:

$$\max_{\pi_\theta \in \Pi_\theta} \quad \bar{G}(\pi_\theta) \tag{4a}$$

$$\text{s.}t. \quad \mathop{\mathbb{E}}_{s\sim d_{\pi_{\theta_k}}} [D_{KL}(\pi_\theta||\pi_{\theta_k})] - \delta \leq 0, \tag{4b}$$

where $\bar{G}(\pi_\theta) = f(\pi_\theta) - \sum_{i=1}^m \varphi_\tau(g_{C_i}(\pi_\theta))$. Here, $\varphi_\tau(.)$ is an extended log-barrier function. Because the standard log barrier $-\log(-g_{C_i}(\pi_\theta))$ has an unbounded gradient near the boundary, we use an affine extension that is matched at $g_{C_i}(\pi_\theta) = -\frac{1}{\tau^2}$ to obtain a bounded and stable barrier. Formally, the extended log-barrier is defined as:

$$\varphi_\tau(g_{C_i}(\pi_\theta)) = \begin{cases} -\frac{1}{\tau}\log(-g_{C_i}(\pi_\theta)) & g_{C_i}(\pi_\theta) \leq -\frac{1}{\tau^2}, \\ \tau g_{C_i}(\pi_\theta) - \frac{1}{\tau}\log(\frac{1}{\tau^2}) + \frac{1}{\tau} & \text{otherwise.} \end{cases} \tag{5}$$

where $\tau > 0$ controls the curvature and smoothness of the barrier. This construction preserves the penalization effect of the standard log barrier in the interior region while avoiding the gradient singularity near active constraints. And the gradient of $\varphi_\tau(g_{C_i}(\pi_\theta))$ is:

$$\nabla\varphi_\tau(g_{C_i}(\pi_\theta)) = \frac{1}{\tau(\gamma-1)\cdot g_{C_i}} \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi_{\theta_k}} \\ a\sim\pi_\theta}} \left[\nabla_\theta \log\pi(a \mid s) \cdot \frac{A^{\pi_{\theta_k}}_{C_i}(s,a)}{(\gamma-1)\cdot\varphi_r(g_{C_i})}\right] \tag{6}$$

Note that problem (4) can be viewed as an approximation to the dual problem of the primal problem (1). The implicit dual variables λ_i^* in this approximation are related to the barrier function derivatives, as will be shown in the duality gap theorem.

While the initial formulation of $\bar{G}(\pi_\theta)$ in problem (4) integrates the objective function and barrier terms to address constraint violations proactively, it lacks an explicit mechanism to guide exploration behaviors, particularly when the policy is near constraint boundaries.

To mitigate this, we introduce the constraint-aware intrinsic reward I^{π_θ} in problem (4), which can be constructed as a cost-sensitive approximation of the per-sample gradient contribution to the logbarrier term. From an information-theoretic perspective, this term captures how much each sample potentially shifts the policy with respect to the constraint boundary. The constraint-aware intrinsic reward $I^{\pi_\theta}_{C_i}$ is formulated proportionally to the magnitude of the cost advantage function normalized by the barrier term, reflecting the alignment between policy gradients and constraint violation risks. Specifically, the reward I^{π_θ} takes the following form:

$$I^{\pi_\theta}_{C_i} \propto \left| \frac{A^{\pi_\theta}_{C_i}(s,a)}{(1-\gamma)\cdot g_{C_i}(\pi_\theta)} \right| \tag{7}$$

Eq.(7) captures constraint sensitivity per unit slack, guiding exploration toward regions where actions strongly affect constraint satisfaction. To modulate the total exploration intensity during training, the intrinsic reward structure employs a gating mechanism based on proximity to constraint values, i.e., we multiply the normalized intrinsic reward by a gating function $\sigma(\alpha_1 d_i - \alpha_2|g_{C_i}(\pi_\theta)|))$, which activates only when the policy approaches the constraint boundary. Meanwhile, a Softmax function is used to map the scale to $(0, 1]$ for numerical stability:

$$\begin{aligned} I^{\pi_\theta}_{C_i} &= \sigma(\alpha_1 d_i - \alpha_2|g_{C_i}(\pi_\theta)|)) \\ &\cdot \text{Softmax}\left(\beta \cdot \left| \frac{A^{\pi_\theta}_{C_i}(s,a)}{(1-\gamma)\cdot \max(|g_{C_i}(\pi_\theta)|,\epsilon)} \right|\right) \end{aligned} \tag{8}$$

Here, σ is the sigmoid function. α_1 and α_2 shape the gate: they determine how wide and how sharply the intrinsic reward activates near the constraint boundary. This formulation resembles the "score $\times$ influence" structure of Fisher information, providing a theoretically grounded measure for boundary-level exploration value. The adjusted magnitude controls the relative incentive among eligible actions. This structure quantifies the sensitivity of model outputs to parameter changes. It promotes behaviors that are more likely to remain within the feasible region and improve constraint satisfaction over time, without rewarding unsafe deviations.

With this intrinsic reward, we can now reformulate the barrier problem to better balance reward maximization, constraint satisfaction, and safe exploration. The barrier problem is as follows:

$$\max_{\pi_\theta \in \Pi_\theta} \quad G(\pi_\theta) \tag{9a}$$

$$s.t. \quad \mathop{\mathbb{E}}_{s\sim d_{\pi_{\theta_k}}} [D_{KL}(\pi_\theta||\pi_{\theta_k})] - \delta \le 0, \tag{9b}$$

where $G(\pi_\theta) = f(\pi_\theta) - \sum_{i=1}^m \varphi_\tau(g_{C_i}(\pi_\theta)) + \eta \sum_{i=1}^m I^{\pi_\theta}_{C_i}$. Here, $\eta = \frac{\omega \cdot G^{\max}}{I^{\max}_{C_i}+\epsilon} > 0$, and ϵ is an extremely small constant to avoid the denominator being zero. To further validate the effectiveness of this intrinsic reward in enhancing optimization performance, we establish the following proposition. Proposition 1 shows that intrinsic reward can provide an additional positive boost for policy optimization. The proof is provided in the Appendix.

Proposition 1 (Enhancement with Intrinsic Reward). *Let π_{k+1} and $\bar{\pi}_{k+1}$ be the policies updated from the same π_k under the PCPO objective with and without the constraint intrinsic reward $I^{\pi_\theta}_{C_i}$, respectively. Define the objective function without the intrinsic reward as $\bar{G}(\pi_k)$. Then, we have:*

$$\begin{aligned} G(\pi_{k+1}) - G(\pi_k) \ge & [\bar{G}(\pi_{k+1}) - \bar{G}(\pi_k)] \\ & + \eta \sum_{i=1}^m \left(I^{\pi_{k+1}}_{C_i} - I^{\pi_k}_{C_i} \right). \end{aligned} \tag{10}$$

Theoretical Analysis of Preemptive Penalty

Next, our method is specifically designed to address CMDP problems using a conservative policy iteration scheme. Throughout the iteration optimization process, our proposed method adheres to the primal constraints. To enhance the optimization of dual problems (9), an upper bound of the duality gap and dual variables are derived as follows:

Theorem 1 (Duality Gap). *An upper bound of the duality gap between the primal problem* (1) *and the dual problem* (9) *is given by*

$$G(\boldsymbol{\lambda}^*) - J(\boldsymbol{\pi}^*) \le \frac{m}{\tau} + \eta \sum_{i=0}^m I^{\max}_{C_i}, \tag{11}$$

where $\boldsymbol{\pi}^*$ is the optimal policy, m is the number of constraints and I^{max} is the upper bound of the expected intrinsic reward. The optimal implicit Lagrangian dual variables satisfy

$$\lambda^*_i = \begin{cases} -\frac{1}{\tau g_{C_i}(\boldsymbol{\pi}^*)} & g_{C_i}(\boldsymbol{\pi}^*) \le -\frac{1}{\tau^2}, \\ \tau & \text{otherwise.} \end{cases} \tag{12}$$

For convenience, to analyze the optimal implicit Lagrangian dual variables, we only rewrite $G(\boldsymbol{\pi}^*)$ as $G(\boldsymbol{\pi}^*, \boldsymbol{\lambda}^*)$ in Theorem 1. Theorem 1 describes the relationship between the duality gap and the parameter τ. As τ increases that leads to the gap diminishes. This upper bound shows how dual variables are adjusted in response to violations. The proof process is elaborated in Appendix C.

In the PCPO algorithm, we denote the policy at the k-th iteration as π_{θ_k}, and the updated policy as $\pi_{\theta_{k+1}}$. The update rule of the PCPO algorithm refers to the mathematical mechanism for transitioning from π_{θ_k} to $\pi_{\theta_{k+1}}$, which is achieved by solving Eq.(9), maximizing $G(\pi_\theta)$ while satisfying the KL-divergence constraint to obtain $\pi_{\theta_{k+1}}$. This update rule ensures that the transition from π_{θ_k} to $\pi_{\theta_{k+1}}$ both improves performance and maintains constraint satisfaction. An explicit lower bound is provided for performance improvements between two adjacent iterations in PCPO, which provides a benchmark to evaluate the effectiveness of the PCPO method. The theoretical result is represented as follows:

Theorem 2 (PCPO Update Performance). *A lower bound for consecutive policies π_{k+1} and π_k improvements is*

$$\begin{aligned} & G(\pi_{k+1}) - G(\pi_k) \ge \\ & \begin{cases} \eta_{k+1} - \frac{1}{\tau}\Psi_{k+1} - \eta S_I, & g_{C_i}(\pi_k) \le -\frac{1}{\tau^2}, \\ -md_i + \eta_{k+1} + \sum_i (\eta^k_{C_i} + \eta^{k+1}_{C_i}) - \eta S_I, & \textit{otherwise.} \end{cases} \end{aligned} \tag{13}$$

The quantities in Theorem 2 are defined as follows: $S_I := \sum_{i=1}^{m} I_{C_i}^{\max}$, $\Psi_{k+1} := \sum_{i=1}^{m} \log(2 - \frac{d_i}{2\eta_{C_i}(\pi_{k+1})})$, $\eta_{k+1} := \eta(\pi_{k+1})$, $\eta_{C_i}^k := \eta_{C_i}(\pi_k)$, and $\eta_{C_i}^{k+1} := \eta_{C_i}(\pi_{k+1})$. Moreover, $\epsilon^{\pi_{k+1}} := \max_s \left|\mathbb{E}_{a\sim\pi_{k+1}}\left[A^{\pi_k}\right]\right|$, $\epsilon_{C_i}^{\pi_k} := \max_s \left|\mathbb{E}_{a\sim\pi_k}\left[A_{C_i}^{\pi_k}\right]\right|$, $\eta(\pi_k) := -\frac{\sqrt{2\delta}\gamma}{(1-\gamma)^2}\epsilon^{\pi_k}$, and $\eta_{C_i}(\pi_k) := -\frac{\sqrt{2\delta}\gamma}{(1-\gamma)^2}\epsilon_{C_i}^{\pi_k}$. Theorem 2 reflects the minimal improvement between two iterations, ensuring reward maximization and constraint satisfaction. This lower bound works as a crucial check that guides parameter tuning and enforces consistent policy evolution. The proof is provided in Appendix D.

Proposition 2 (Advantage in Cumulative Constraint Violations). *In a CMDP with the constraint set $\{J_C(\pi) \leq d\}$, for any initial policy π_0 and number of iterations T, the cumulative constraint violation of the PCPO method (denoted $V_P(T)$) and that of Lagrangian-based safe RL methods (denoted $V_L(T)$) satisfy:*

$$V_P(T) \leq V_L(T) - \Delta(T), \tag{14}$$

where $\Delta(T) > 0$ is a gap term positively correlated with T, and $\Delta(T) \to \infty$ as $T \to \infty$.

Based on Theorem 2, we further compare the improvement between adjacent iterations during the training processes of PCPO and Lagrangian-based safe RL methods in Proposition 4. Lagrangian-based safe RL methods effectively manage constraints using Lagrange multipliers to ensure safety while maximizing rewards, as demonstrated in (Chow et al. 2019; Tessler, Mankowitz, and Mannor 2018; Stooke, Achiam, and Abbeel 2020). The proof of Proposition 4 is provided in the Appendix.

Practical Implementation

Parameterized Objectives and Constraints

This section delves into the practical implementation where the dual problem, as outlined in Eq.(9), is resolved directly despite its computational expense and inefficiency. The focus shifts to exploring parameterized policies to develop a practical algorithm capable of managing finite sample counts and arbitrary initial parameters. The notation from the previous sections has been simplified using the parameter vector θ. Therefore, the parameterized objective function is subject to constraints and its gradients are

$$\max_\theta G(\theta) = \max_\theta f(\theta) - \sum_{i=1}^{m} \varphi_\tau(g_{C_i}(\theta)) + \eta \sum_{i=1}^{m} I_{C_i}^{\pi_\theta}(\theta), \tag{15}$$

$$\begin{aligned}&\nabla G(\theta)\\ =&\nabla f(\theta) - \sum_{i=1}^{m} \nabla\varphi_\tau(g_{C_i}(\theta))\nabla g_{C_i}(\theta) + \eta\nabla \sum_{i=1}^{m} I_{C_i}^{\pi_\theta}(\theta).\end{aligned} \tag{16}$$

In addition, the Fisher Information Matrix (FIM) is employed, utilizing analytical computations of the Hessian matrix derived from the KL divergence. As parameterized policies are considered, the previous notation will be extended to incorporate functions of θ instead of π.

The parameter update for Eq.(16) is approximated employing a linear approximation of the objective function and a quadratic approximation for the KL divergence constraint.

$$\theta_{k+1} = \arg\max_\theta [\nabla_\theta G^T(\theta)\,|_{\theta=\theta_k}(\theta - \theta_k)], \tag{17a}$$

$$\text{s.t. } \frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k) \leq \delta, \tag{17b}$$

where H is the Hessian matrix of the KL-divergence that is given by

$$H(\theta_k)_{ij} = \frac{\partial}{\partial\theta_i}\frac{\partial}{\partial\theta_j} \mathop{\mathbb{E}}_{s\sim d_{\pi_{\theta_k}}} [D_{KL}(\theta\|\theta_k)]\Big|_{\theta=\theta_k}. \tag{18}$$

It is important to note that in practice, the Fisher information matrix H may not always be invertible. To address this issue, we consider a regularized version of the Fisher information matrix $\hat{H} = H + \lambda I$, where $\lambda > 0$ is a small regularization parameter and I is the identity matrix. This ensures $\hat{H}$ is positive definite and invertible.

We consider a transformation $p = \hat{H}^{1/2}(\theta - \theta_k)$ in Eq.(17), then Eq.(17) is equivalent to optimizing a linear function on a norm ball:

$$\max_p \quad [\nabla_\theta G^T(\theta)|_{\theta=\theta_k}\hat{H}^{-1/2}p] \tag{19a}$$

$$\text{s.t.} \quad \frac{1}{2}p^T p \leq \delta \tag{19b}$$

With this reformulation, the desired result follows in a relatively straightforward manner. The solution is simply:

$$p^* = \sqrt{2\delta}\cdot\frac{\hat{H}^{-1/2}\nabla G}{\|\hat{H}^{-1/2}\nabla G\|_2} \tag{20}$$

Transforming back to the original variable in Eq.(17) is:

$$\theta^* = \theta_k + \sqrt{2\delta}\cdot\frac{\hat{H}^{-1}\nabla G}{\sqrt{\nabla G^T\hat{H}^{-1}\nabla G}} \tag{21}$$

This approach does not require the strict feasible point assumption in Eq.(17) and effectively handles cases where the Fisher information matrix may be non-invertible.

Estimation of Objectives and Constraints

Generalized Advantage Estimation (GAE) is leveraged to accurately estimate the policy gradient, providing a robust foundation for subsequent policy updates. In this process, a sequence of states is gathered through sampling and policy simulation over a specified number of timesteps to produce a trajectory. At each state-action pair, the Q-value is calculated by summing the future rewards discounted over the trajectory's length.

After estimating the policy gradient, we implement these calculations within a structured policy iteration framework. Algorithm 1 shows a policy iteration method based on the performance improvement bound in (13), employing the minimization-maximization algorithm (Hunter and Lange 2004). A detailed pseudocode of the supplementary materials is available in the Appendix.

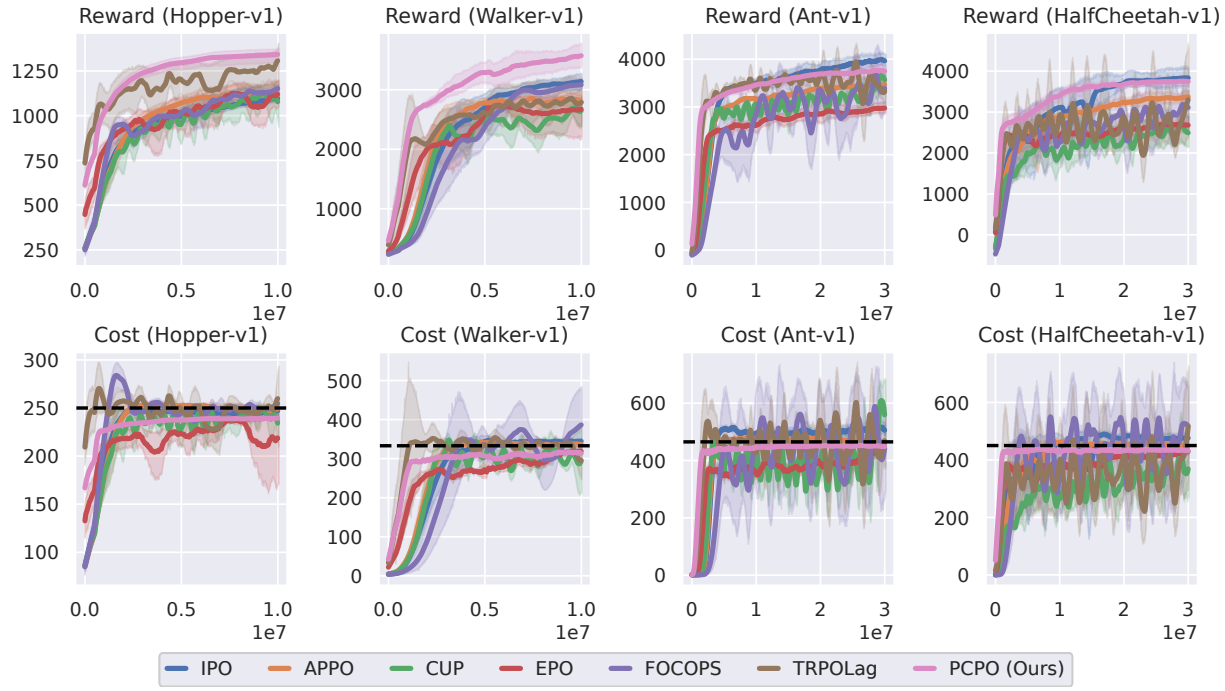


Figure 2: The average performance for IPO, APPO, CUP, EPO, FOCOPS, TRPOLag and PCPO over 6 seeds (about 1000 bootstrap samples). The x-axis indicates the total number of samples, while the y-axis shows the mean total reward/cost return from the most recent 100 episodes. The shaded regions indicate the bootstrap normal 95% confidence interval.

Algorithm 1: PCPO

1: Initialize policy network $\pi_0 = \pi_{\theta_0}$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Run $\pi_k = \pi_{\theta_k}$ and store trajectories in $\mathcal{D}$
4: Estimate $\nabla\hat{G}(\theta), \nabla\hat{\varphi}_\tau(g_{C_i}(\theta))\nabla g_{C_i}(\theta), \nabla I^{\pi_\theta}_{C_i}(\theta)$, with $\mathcal{D}$
5: Update θ_{k+1} using Eq.(17)
6: Empty $\mathcal{D}$
7: **end for**

Experiments

The efficacy of the PCPO algorithm is demonstrated with various robotic agents on both Safe Velocity and Safe Navigation tasks, which are integrated into the Safety-gymnasium (Ji et al. 2023) and executed using the MuJoCo physical simulator. Specifically, in the Safe Velocity tasks, four robotic agents are trained to move under strict speed limits. In Safe Navigation tasks, two agents perform circular-motion tasks within constrained environments.

Seven distinct RL algorithms are employed for comparative analysis within these environments. IPO (Liu, Ding, and Liu 2020) applies a log-barrier to enforce constraints. CUP (Yang et al. 2022) and FOCOPS (Zhang, Vuong, and Ross 2020) represent constrained policy update methods that explore the policy parameter space to satisfy constraints. TRPOLag (Schulman et al. 2015) combines TRPO's trust region with the Lagrangian optimization. APPO (Dai et al. 2023) reduces oscillation in Lagrangian-based learning. EPO (Gao et al. 2024), enhances safe exploration via adaptive penalties. Further details are provided in the Appendix.

Evaluation on Safe Velocity Task

Four MuJoCo environments are selected for training a robotic agent to walk, subject to a speed limit. The cost thresholds are set at 50 percent of the cost required by an unconstrained TRPO agent, which is trained with 10 million samples. According to Figure 2, PCPO consistently outperforms most baseline algorithms across the Safe Velocity tasks while strictly adhering to the speed limits. In comparison with APPO, CUP, EPO, and FOCOPS, PCPO achieves superior overall performance and significantly lower constraint violations. IPO shows competitive returns in certain environments, but this advantage is accompanied by substantial constraint violations, indicating that its improvement largely comes from exploring unsafe regions. In contrast, PCPO maintains constraint satisfaction. While TRPOLag performs comparably well in Ant-v1, PCPO achieves markedly lower violations and reduced oscillation, resulting in a more stable and safety-compliant learning process. In Hopper-v1, TRPOLag appears to continue improving with more iterations; however, this trend may be influenced by its broader exploration of unsafe regions before convergence. Overall, our analysis confirms that PCPO effectively executes all tasks. In Table 1, we summarize the performance of all seven algorithms.

Environment		IPO	APPO	CUP	EPO	FOCOPS	TRPOLag	PCPO
Hopper	Return	1096±27	1150± 58	1077 ± 175	1119 ± 106	1152 ± 28	1307 ± 164	$\mathbf{1342 \pm 59}$
(250.)	Cost	247±10	249±4	234 ± 32	218 ± 62	254 ± 8	259 ± 40	239 ± 3
Walker	Return	3138±192	2798±155	2649 ± 352	2672 ± 789	3084 ± 126	2785 ± 543	$\mathbf{3572 \pm 271}$
(333.)	Cost	344±8	336±8	293 ± 122	318 ± 12	386 ± 117	295 ± 61	313 ± 18
Ant	Return	3966±193	3469±47	3569 ± 218	2975 ± 170	3388 ± 611	3444 ± 374	$\mathbf{3752 \pm 228}$
(465.)	Cost	505±25	460±4	559 ± 180	442 ± 20	443 ± 209	467 ± 110	449 ± 8
HalfCheetah	Return	3835±427	3373±257	2494 ± 524	2680 ± 318	3111 ± 669	3303 ± 1634	$\mathbf{3747 \pm 167}$
(450.)	Cost	476±29	449±4	369 ± 128	429 ± 18	518 ± 188	515 ± 294	432 ± 16

Table 1: Average episode return and cost for IPO, APPO, CUP, EPO, FOCOPS, TRPOLag and PCPO over 6 seeds. The results include the bootstrap mean and normal 95% confidence interval, derived from 1000 bootstrap samples. Cost thresholds are indicated in brackets beneath the environment names.

Evaluation on Safe Navigation Task

For these tasks, we examine the performance of the point and car agents within the Safe Navigation tasks, particularly focusing on the circular motion mode. The objective is to train a robotic agent that can navigate within a circular perimeter while avoiding the boundaries of the environment. Consistent with prior tasks, we display the learning curves and provide numerical summaries. Notably, our findings reveal that PCPO effectively enforces the constraints in both experiments while achieving a commendable reward.

As depicted in Figure A1 and Table A3 in the Appendix, similar to the results of the experiment on Safe Velocity tasks, IPO, CUP, EPO, FOCOPS, and TRPOLag exhibit considerable fluctuations in meeting the target constraints, whereas our method remains highly stable and consistently falls below the cost limitation thresholds throughout the learning process. APPO also achieves relatively stable constraint satisfaction, but its final performance is noticeably lower than that of PCPO. The result indicates that PCPO not only minimizes constraint violations but also converges to a superior solution.

Generalization Analysis

RL algorithm relies heavily on interaction with the environment, adjusting decision-making processes based on rewards and penalties. Traditional methods partition datasets into training, validation, and test sets to evaluate algorithm performance through simulated assessments, yet fail to adequately maintain consistency with RL training dynamics. Thus, such fixed dataset allocations are insufficient for extrapolating algorithm performance levels in actual tasks. In contrast, adaptation to complex tasks and dynamic changes can be more effectively achieved by interacting directly with the environment and learning through trial and error. This approach leads to more accurate solutions.

To better assess the generalization of the PCPO algorithm, an approach utilizing fixed and unseen random seeds is implemented. This method evaluates performance under varied conditions, reflecting real-world scenarios more accurately. Experiments demonstrate that the PCPO algorithm outperforms others by effectively adapting to different test environments and consistently enhancing performance while maintaining lower costs. For specific experimental details and data, please refer to the Appendix.

Sensitivity Analysis

In this analysis, it is investigated the dependency of the PCPO algorithm's performance on the hyperparameters τ. The experiment investigations demonstrate that the algorithm exhibits robustness to variations in this parameter. Experiments are conducted on robots operating under speed constraints to validate this observation.

We conducted experiments with PCPO across 4 distinct τ values, simultaneously maintaining all other parameters fixed. Finally, the experiment observes that environments presenting greater difficulty, exhibit heightened sensitivity to parameter selections. Nevertheless, PCPO demonstrates substantial insensitivity to hyperparameter choices overall. Additionally, we note that our methods gain a more significant return and cost with a greater selection of τ. We also investigate the impact of the weighting parameter ω, which controls the relative contribution of the constraint-aware intrinsic reward with respect to the environment reward. See Appendix for details and data of specific results

Ablation Study

The ablation studies highlight the effectiveness of preemptive penalties in the PCPO method, unlike the TRPOLag method, which does not utilize these penalties, as demonstrated in Figures 2 and A1 in the Appendix, and Tables 1 and A3 in the Appendix. The importance of intrinsic reward for dynamically shaping exploration and improving the effectiveness of safe policy learning is emphasized. We also evaluate the effectiveness of our designed barrier function. The results of Safe Velocity tasks indicate significant performance improvements with the PCPO method. For further details, see Appendix.

Conclusion

In this article, we present PCPO, which integrates preemptive penalties and a constraint-aware intrinsic reward to mitigate oscillation, overshooting, and vanishing gradients. Experiments show that PCPO improves safety during learning and outperforms existing algorithms across standard benchmarks. This method effectively reduces constraint violations, adapts to varying cost limits, and demonstrates strong generalization capabilities. The implementation of PCPO is straightforward, making it accessible to researchers in other fields, particularly in safety-critical applications.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62301559.

References

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained policy optimization. In *International conference on machine learning*, 22–31. PMLR.

Altman, E. 2021. *Constrained Markov decision processes*. Routledge.

Chen, W.; Onyejizu, J.; Vu, L.; Hoang, L.; Subramanian, D.; Kar, K.; Mishra, S.; and Paternain, S. 2024. Adaptive Primal-Dual Method for Safe Reinforcement Learning. arXiv:2402.00355.

Chow, Y.; Ghavamzadeh, M.; Janson, L.; and Pavone, M. 2018a. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(167): 1–51.

Chow, Y.; Nachum, O.; Duenez-Guzman, E.; and Ghavamzadeh, M. 2018b. A lyapunov-based approach to safe reinforcement learning. *Advances in neural information processing systems*, 31.

Chow, Y.; Nachum, O.; Faust, A.; Duenez-Guzman, E.; and Ghavamzadeh, M. 2019. Lyapunov-based Safe Policy Optimization for Continuous Control. arXiv:1901.10031.

Dai, J.; Ji, J.; Yang, L.; Zheng, Q.; and Pan, G. 2023. Augmented Proximal Policy Optimization for Safe Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6): 7288–7295.

Dalal, G.; Dvijotham, K.; Vecerik, M.; Hester, T.; Paduraru, C.; and Tassa, Y. 2018. Safe Exploration in Continuous Action Spaces. arXiv:1801.08757.

Ding, Y.; and Lavaei, J. 2023. Provably efficient primal-dual reinforcement learning for cmdps with non-stationary objectives and constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7396–7404.

Gao, S.; Ding, J.; Fu, L.; Wang, X.; and Zhou, C. 2024. Exterior Penalty Policy Optimization with Penalty Metric Network under Constraints. In *International Joint Conference on Artificial Intelligence*.

Ghosh, R. 2025. Safe Reinforcement Learning for Multi-Agent Systems with Risk Constraints.

Gu, S.; Sel, B.; Ding, Y.; Wang, L.; Lin, Q.; Jin, M.; and Knoll, A. 2024a. Balance reward and safety optimization for safe reinforcement learning: A perspective of gradient manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 21099–21106.

Gu, S.; Yang, L.; Du, Y.; Chen, G.; Walter, F.; Wang, J.; and Knoll, A. 2024b. A review of safe reinforcement learning: Methods, theories and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Hao, G.; Li, Y.; Li, Y.; Jiang, L.; and Zeng, Z. 2024. Lyapunov-based safe reinforcement learning for microgrid energy management. *IEEE transactions on neural networks and learning systems*.

Honari, H.; Soufi Enayati, A. M.; Ghafarian Tamizi, M.; and Najjaran, H. 2024. Meta SAC-Lag: Towards Deployable Safe Reinforcement Learning via MetaGradient-based Hyperparameter Tuning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 619–626.

Hunter, D. R.; and Lange, K. 2004. A tutorial on MM algorithms. *The American Statistician*, 58(1): 30–37.

Ibrahim, S.; Mostafa, M.; Jnadi, A.; Salloum, H.; and Osinenko, P. 2024. Comprehensive Overview of Reward Engineering and Shaping in Advancing Reinforcement Learning Applications. *IEEE Access*, 12: 175473–175500.

Ji, J.; Zhang, B.; Zhou, J.; Pan, X.; Huang, W.; Sun, R.; Geng, Y.; Zhong, Y.; Dai, J.; and Yang, Y. 2023. Safety gymnasium: A unified safe reinforcement learning benchmark. *Advances in Neural Information Processing Systems*, 36.

Kervadec, H.; Dolz, J.; Yuan, J.; Desrosiers, C.; Granger, E.; and Ayed, I. B. 2022. Constrained deep networks: Lagrangian optimization via log-barrier extensions. In *2022 30th European Signal Processing Conference (EUSIPCO)*, 962–966. IEEE.

Kwon, H.; Lee, G.; Lee, J.; and Oh, S. 2024. Safe CoR: A Dual-Expert Approach to Integrating Imitation Learning and Safe Reinforcement Learning Using Constraint Rewards. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2893–2898. IEEE.

Li, Z.; Zhu, D.; and Grossklags, J. 2025. Safe Reinforcement Learning via Episodic Control. *IEEE Access*.

Liu, Y.; Ding, J.; and Liu, X. 2020. IPO: Interior-point policy optimization under constraints. In *Proceedings of the AAAI conference on artificial intelligence*, 4940–4947.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937. PMLR.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International conference on machine learning*, 1889–1897. PMLR.

Silver, D. 2015. Reinforcement learning.

Stooke, A.; Achiam, J.; and Abbeel, P. 2020. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, 9133–9143. PMLR.

Su, T.; Wu, T.; Zhao, J.; Scaglione, A.; and Xie, L. 2025. A review of safe reinforcement learning methods for modern power systems. *Proceedings of the IEEE*.

Tessler, C.; Mankowitz, D. J.; and Mannor, S. 2018. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*.

Wachi, A.; Shen, X.; and Sui, Y. 2024. A Survey of Constraint Formulations in Safe Reinforcement Learning. arXiv:2402.02025.

Wagener, N. C.; Boots, B.; and Cheng, C.-A. 2021. Safe reinforcement learning using advantage-based intervention.

In *International Conference on Machine Learning*, 10630–10640. PMLR.

Wang, Y.; Zhan, S. S.; Jiao, R.; Wang, Z.; Jin, W.; Yang, Z.; Wang, Z.; Huang, C.; and Zhu, Q. 2023. Enforcing Hard Constraints with Soft Barriers: Safe Reinforcement Learning in Unknown Stochastic Environments. In *International Conference on Machine Learning*, 36593–36604.

Yang, L.; Ji, J.; Dai, J.; Zhang, L.; Zhou, B.; Li, P.; Yang, Y.; and Pan, G. 2022. Constrained update projection approach to safe policy optimization. *Advances in Neural Information Processing Systems*, 35: 9111–9124.

Yang, T.-Y.; Rosca, J.; Narasimhan, K.; and Ramadge, P. J. 2020. Projection-based constrained policy optimization. *arXiv preprint arXiv:2010.03152*.

Ying, D.; Zhang, Y.; Ding, Y.; Koppel, A.; and Lavaei, J. 2024. Scalable primal-dual actor-critic method for safe multi-agent rl with general utilities. *Advances in Neural Information Processing Systems*, 36.

Zhang, Q.; Leng, S.; Ma, X.; Liu, Q.; Wang, X.; Liang, B.; Liu, Y.; and Yang, J. 2024. CVaR-constrained policy optimization for safe reinforcement learning. *IEEE transactions on neural networks and learning systems*, 36(1): 830–841.

Zhang, Y.; Vuong, Q.; and Ross, K. 2020. First order constrained optimization in policy space. *Advances in Neural Information Processing Systems*, 33: 15338–15349.