Balance Reward and Safety Optimization for Safe Reinforcement Learning: A Perspective of Gradient Manipulation

Shangding Gu[1], Bilgehan Sel[2], Yuhao Ding[3], Lu Wang[4], Qingwei Lin[4], Ming Jin[2*], Alois Knoll[1*]

[1]Technical University of Munich
[2] Virginia Tech
[3]University of California, Berkeley
[4]Microsoft Research

Abstract

Ensuring the safety of Reinforcement Learning (RL) is crucial for its deployment in real-world applications. Nevertheless, managing the trade-off between reward and safety during exploration presents a significant challenge. Improving reward performance through policy adjustments may adversely affect safety performance. In this study, we aim to address this conflicting relation by leveraging the theory of gradient manipulation. Initially, we analyze the conflict between reward and safety gradients. Subsequently, we tackle the balance between reward and safety optimization by proposing a soft switching policy optimization method, for which we provide convergence analysis. Based on our theoretical examination, we provide a safe RL framework to overcome the aforementioned challenge, and we develop a Safety-MuJoCo Benchmark to assess the performance of safe RL algorithms. Finally, we evaluate the effectiveness of our method on the Safety-MuJoCo Benchmark and a popular safe benchmark, Omnisafe. Experimental results demonstrate that our algorithms outperform several state-of-the-art baselines in terms of balancing reward and safety optimization.

Introduction

Reinforcement Learning (RL) has demonstrated remarkable performance in various scenarios (Gu et al. 2022b), such as the game of Go (Silver et al. 2016), autonomous driving (Kiran et al. 2021; Gu et al. 2022a), and robotics (Kober, Bagnell, and Peters 2013; Gu et al. 2023b). However, the majority of RL methods are restricted to simulation environments due to safety concerns associated with deploying RL in real-world settings. To address this issue, numerous safe RL methods have been proposed to tackle the safety challenge.

For instance, Constrained Policy Optimization (CPO) (Achiam et al. 2017) is developed to ensure reward monotonic improvement while maintaining safety. PPO Lagrangian and TRPO Lagrangian methods (Ray, Achiam, and Amodei 2019) are introduced to address the balance between reward and safety performance by employing Lagrangian optimization. Additionally, safe exploration methods based on a Gaussian Process (GP) (Sui et al. 2015) are developed to guarantee exploration safety by utilizing a GP to model the exploration safety of the state space. However, these methods may not effectively resolve the conflict reward and cost gradients, and balance reward and safety optimization. A key question that is raised in this domain is: How can we handle the balance between reward and safety optimization?

In this research, we aim to address the key question by leveraging the theory of gradient manipulation, wherein we conduct a detailed examination of the changes in gradients associated with reward and safety. Based on our theoretical analysis, we propose the projection constraint-rectified policy optimization (PCRPO) method, designed to alleviate the conflict between reward and safety optimization while maintaining a balance between their optimization levels, in which soft switching policy optimization through gradient manipulation is proposed and a slack technique is introduced for adjusting the emphasis on safety optimization. Particularly, we evaluate the effectiveness of our method on a multitude of challenging tasks, and conduct ablation experiments to thoroughly examine the performance of our method. The empirical findings suggest that our approach outperforms strong baselines concerning the balance between reward maximization and safety preservation.

The present study offers several significant contributions to the field, which are enumerated as follows: (1) The introduction of a novel problem concerning safe RL involving conflicts between reward and cost gradients. (2) The development of a safe RL framework employing soft switching via gradient manipulation. (3) The establishment of a new benchmark for Safe RL evaluation, designed to assess the performance of safe RL algorithms; (4) The demonstration that the practical algorithms proposed in this study surpass existing state-of-the-art baselines with respect to both reward and safety performance.

Related Work

In recent years, numerous safe RL methods have been proposed to ensure RL safety (Gu et al. 2022b, 2023a). These safe RL methods can be briefly categorized into three main groups: (1) Control theory-based safe RL: These methods leverage principles from control theory, such as model predictive control and Lyapunov functions, to ensure that the

*Equally advise. Corresponding authors: Ming Jin (*jinming@vt.edu*) and Shangding Gu (*shangding.gu@tum.de*).

agent operates within safety constraints while learning optimal behavior (Koller et al. 2018). (2) Formal methods-based safe RL: These approaches employ formal verification and synthesis techniques, such as temporal logic, to guarantee that the learned policies satisfy safety specifications (Fulton and Platzer 2018). (3) Constrained optimization-based safe RL: These methods focus on optimizing the agent's behavior while adhering to safety constraints. Techniques like constrained policy optimization and Lagrangian relaxation are used to ensure that the RL algorithm respects the safety limits during learning (Brunke et al. 2022).

Specifically, from the control theory perspective, Lyapunov functions are employed to ensure learning safety by constraining the action space of exploration (Chow et al. 2018, 2019). Although Lyapunov function-based methods can demonstrate good performance in ensuring learning safety, defining specific Lyapunov functions requires a system model, and it is usually challenging to find a function that can handle general safe RL problems. From the formal methods perspective, some methods based on formal techniques are proposed to guarantee RL safety. For example, temporal logic verification is used to verify safe actions during exploration (Li and Belta 2019). Such methods can rigorously ensure learning safety. However, external knowledge is needed to define the safe state and action space, which may be difficult to deploy in real-world RL applications.

Compared to the aforementioned methods, constrained optimization-based safe RL methods have gained considerable attention due to their relative maturity and broad applicability. One branch of constrained optimization-based safe RL methods encompasses primal-dual methods (Boyd and Vandenberghe 2004). A notable method within this branch is CPO, which employs TRPO (Schulman et al. 2015) in constrained optimization and can nearly guarantee hard constraints via a line search method (Nocedal and Yuan 1998). PPO-Lagrangian, another representative primal-dual optimization method (Zhou and et al. 2023; Calian and et al. 2020), is developed based on Lagrangian optimization and dynamically adjusts the Lagrangian multiplier in response to safety violations. Following CPO and PPO-Lagrangian, recent state-of-the-art baselines, such as PCPO (Yang et al. 2020) and CUP (Yang et al. 2022), are proposed to ensure learning safety. Another branch of constrained optimization-based safe RL methods consists of primal methods (Boyd and Vandenberghe 2004). CRPO (Xu, Liang, and Lan 2021), a representative method for primal optimization, directly enhances reward performance while ensuring learning safety within the primal problem. In contrast to primal-dual-based methods, primal-based methods offer ease of implementation and are not burdened by hyperparameter tuning issues related to dual variables. Moreover, primal-based safe RL methods do not necessitate feasible initialization. However, poor initialization can adversely affect the performance of primal-dual optimization-based methods (Xu, Liang, and Lan 2021).

The methods mentioned above do not explicitly analyze and address the gradient conflicts between reward and cost optimization. This oversight can lead to significantly negative effects on safe RL performance. In contrast to previous works, our proposed method, which is based on primal optimization, necessitates only gradients from the objective and the costs to ensure safe exploration. This is a key difference from other methods like CRPO, where gradient conflicts may lead to unsafe exploration and wasted samples during training. By focusing on resolving these gradient conflicts, our approach aims to provide a more effective solution for safe RL applications.

Problem Formulation

Markov Decision Processes An infinite-horizon Markov Decision Process MDP$(\mathcal{S}, \mathcal{A}, P, r, \gamma)$ is specified by: a state space $\mathcal{S}$; an action space $\mathcal{A}$; a transition dynamics $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, where $P(s'|s, a)$ is the probability of transition from state s to state s' when action a is taken; a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, where $r(s, a)$ is the instantaneous reward when taking action a in state s; a discount factor $\gamma \in [0, 1)$. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ represents that the decision rule the agent uses, i.e. the agent takes action a with probability $\pi(a|s)$ in state s. Given a policy π, the value function $V^\pi : \mathcal{S} \to \mathbb{R}$ is defined to characterize the discounted sum of the rewards earned under π, i.e.

$$V_r^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, \pi, s_0 = s\right], \ \forall s \in \mathcal{S} \quad (1)$$

where the expectation is taken over all possible trajectories, in which $a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$. When the initial state is sampled from some distribution ρ, we slightly abuse the notation and define the value function as

$$V_r^\pi(\rho) := \mathbb{E}_{s\sim\rho}\left[V^\pi(s)\right]. \quad (2)$$

The action-value function (or Q-function) $Q_r^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ under policy π is defined as

$$Q_r^\pi(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, \pi, s_0 = s, a_0 = a\right], \quad (3)$$

which can be interpreted as the expected total reward with an initial state $s_0 = s$ and an initial action $a_0 = a$.

Constrained MDP In a Constrained Markov Decision Process CMDP$(\mathcal{S}, \mathcal{A}, P, r, \mathbf{c}, \mathbf{b}, \gamma)$, besides the reward function r, we have a cost function $\mathbf{c} = (c_1, \ldots, c_n) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^n$ and a threshold $\mathbf{b} = (b_1, \ldots, b_n) \in \mathbb{R}^n$. In the safety-critical environments, the agent aims at maximizing the expected (discounted) cumulative reward for a given initial distribution ρ while satisfying constraints on the expected (discounted) cumulative cost, i.e.,

$$\max_{\pi\in\Pi} V_r^\pi(\rho), \ \text{ s.t. } V_{c_i}^\pi(\rho) \le b_i, \ \forall i = 1, \ldots, n. \quad (4)$$

where the expectation is taken over all possible trajectories, and $V_r^\pi(\rho)$ and $V_{c_i}^\pi(\rho)$ denote the value function corresponding to the reward and cost functions, respectively.

Primal vs Primal-dual Approaches The current safe RL methods can be generally categorized into the **primal** and **primal-dual** approaches. In **primal-dual** optimization, the primal-dual approaches convert the constrained problem (4) into an unconstrained one by augmenting the objective with

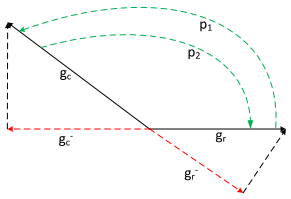

Figure 1: Conflicts between reward and cost optimization.

a sum of constraints weighted by their corresponding dual variables $\boldsymbol{\lambda}$. The associated Lagrangian function $L(\pi, \boldsymbol{\lambda})$ is defined as:

$$L(\pi, \boldsymbol{\lambda}) \coloneqq V_r^\pi(\rho) - \boldsymbol{\lambda}^T \left(V_{\mathbf{c}}^\pi(\rho) - \mathbf{b}\right), \tag{5}$$

where $\boldsymbol{\lambda} \in \mathbb{R}_+^n$. The safe policy is learned from applying a certain policy optimization update such as (natural) policy gradient alternatively with a gradient descent type update for the dual variables: $\pi_{t+1} = \pi_t + \eta_1 \widetilde{\nabla_\pi L}(\pi_t, \boldsymbol{\lambda}_t), \boldsymbol{\lambda}_{t+1} = \mathcal{P}_U\left(\boldsymbol{\lambda}_t - \eta_2 \left(V_{\mathbf{c}}^\pi(\rho) - \mathbf{b}\right)\right), \quad \text{for } t = 0, 1, 2, \ldots,$ where $\eta_1 > 0$, $\eta_2 > 0$ are step-sizes, $\widetilde{\nabla_\pi L}$ can be the policy gradient or its variants, and the dual feasible region $U \coloneqq [0, C_0]$ is an interval that contains $\boldsymbol{\lambda}^\star$.

In **primal** optimization, the necessity for dual variables is eliminated, enabling the immediate optimization of rewards and costs. Such approaches, exemplified by CRPO (Xu, Liang, and Lan 2021), have demonstrated superior outcomes compared to conventional primal-dual techniques with guaranteed convergence. Nevertheless, when transitioning between reward and cost optimization, conflicting relationships may arise between reward gradients $\boldsymbol{g_r}$ and cost gradients $\boldsymbol{g_{c_i}}$. This conflict f_{rc_i} has the potential to negatively impact the efficacy of primal methods in terms of both reward and safety performance. As depicted in Figure 1, the reward gradient is represented by $\mathbf{g}_r$, while the cost gradient is denoted by $\mathbf{g}_c$. Additionally, $\mathbf{g}_c^-$ signifies the projection of the cost gradient $\mathbf{g}_c$ onto the plane of the reward gradient $\mathbf{g}_r$, and $\mathbf{g}_r^-$ refers to the projection of the reward gradient $\mathbf{g}_r$ onto the plane of the cost gradient $\mathbf{g}_c$. During the primal optimization, a transition from cost optimization to reward optimization occurs. In this scenario, the cost gradient optimization process adversely impacts the reward optimization process (p_2). Consequently, the current gradient is expressed as $\mathbf{g} = \mathbf{g}_r - \mathbf{g}_c^-$. Conversely, when switching from reward optimization to cost optimization (p_1), the current gradient is given by $\mathbf{g} = \mathbf{g}_c - \mathbf{g}_r^-$.

Consequently, it is crucial to ascertain a method for delicately balancing the relationship between rewards and costs while simultaneously mitigating the adverse effects of conflicting gradients and minimizing the optimization oscillation on the overall performance of the approach. To satisfy the above requirements, a safe RL problem is rewritten as Equation (6), where f_{rc_i} denotes the deviation between gradients, which could result in conflicting gradients and oscillations in the optimization process.

$$\begin{aligned} &\max_{\pi \in \Pi} V_r^\pi(\rho), \\ &\text{s.t. } V_{c_i}^\pi(\rho) \le b_i, \ \forall i = 1, \ldots, n. \\ &\quad \min_{\pi \in \Pi} f_{rc_i} = f(\mathbf{g_r}, \mathbf{g_{c_i}}). \end{aligned} \tag{6}$$

Method

In order to tackle the safe RL problem, as illustrated in Equation (6), it is imperative to first address the conflict between reward and cost gradients. To this end, we introduce a novel soft switching optimization solution that employs gradient manipulation to achieve a balanced relationship between these gradients. This approach incorporates a slack mechanism designed to gently optimize both reward and cost. Subsequently, we analyse the gradient change by soft switching. Then, the convergence analysis is provided. Lastly, we present a safe RL framework though gradient manipulation based on primal optimization, and a practical algorithm that effectively facilitates the implementation of our proposed method in real-world scenarios.

Soft Switching through Gradient Manipulation

By leveraging the gradient manipulation, we effectively regulate the switching transitions and minimize the oscillations between reward and cost optimization. The objective of soft switching in this context is to enhance the overall efficiency, performance, and reliability of the algorithms, while simultaneously reducing the deviation of gradients between reward and cost components. As illustrated in Figure 2, the cost gradient of constraint i is denoted by $\mathbf{g}_{c_i}$. For the sake of simplicity, we represent it as $\mathbf{g}_c$. The projection gradient of $\mathbf{g}_c$ on the normal plane of gradient $\mathbf{g}_r$ is given by $\mathbf{g}_{c_i}^+$, while the projection gradient of $\mathbf{g}_r$ on the normal plane of gradient $\mathbf{g}_{c_i}$ is represented by $\mathbf{g}_r^+$. The angle between gradients g_r and g_c is denoted by θ.

During the gradient manipulation process, the gradient projection is employed if the angle θ exceeds 90°, as demonstrated in Equation (8) (A simplified illustration of this scenario can be observed in Figure 2 (a)), where β_r^+ and β_c^+ denote the weights of gradient $\mathbf{g}_r^+$ and gradient $\mathbf{g}_c^+$, respectively. Conversely, when the angle θ is less than or equal to 90°, Equation (9) is leveraged to handle gradient manipulation (A simplified illustration of this scenario can be observed in Figure 2 (b)), where β_r and β_c denote the weights of gradient $\mathbf{g}_r$ and gradient $\mathbf{g}_c$, respectively. Our approach aims to minimize optimization oscillations by reducing the deviation between reward and cost gradients, $f_{rc} = f(\mathbf{g_r}, \mathbf{g_c}) = \theta$, particularly for conflicting gradients between reward and cost optimization. This finally allows us to identify a gradient that can effectively satisfy safety constraints while simultaneously enhancing reward performance. In next section, we will analyze how gradient change with soft switching.

$$\mathbf{g}_r^+ = \mathbf{g}_r - \frac{\mathbf{g}_r \cdot \mathbf{g}_c}{|\mathbf{g}_c|^2}\mathbf{g}_c, \quad \mathbf{g}_c^+ = \mathbf{g}_c - \frac{\mathbf{g}_c \cdot \mathbf{g}_r}{|\mathbf{g}_r|^2}\mathbf{g}_r, \tag{7}$$

$$\mathbf{g} = \beta_r^+ \mathbf{g}_r^+ + \beta_c^+ \mathbf{g}_c^+, \tag{8}$$

$$\mathbf{g} = \beta_r \mathbf{g}_r + \beta_c \mathbf{g}_c. \tag{9}$$

Gradient Analysis with Soft Switching

In this analysis, for the purpose of simplification, we consider β_r^+, β_c^+, β_r and β_c to be equal to 0.5. Other cases follow a

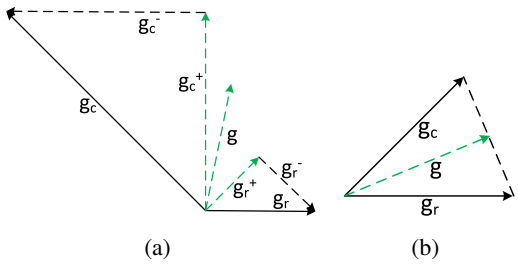

Figure 2: Soft switching through gradient manipulation.

similar pattern and can be easily proven based on our analytical framework. In instances where $\theta \geq 90°$, we have two strategies to handle the conflict gradients between reward and safety optimization. The first strategy is to leverage Equation (9) to address the policy gradient, as demonstrated in Figure 3 (a), results in the gradient being represented by $\mathbf{g}^-$, which is indicated by the red arrow. The second strategy is to employ Equation (8) that allows the gradient to be depicted as $\mathbf{g}$, denoted by the green dashed line in Figure 3 (a). To assess which gradient manipulation is better, we provide the following analysis for this instance. Specifically, by capitalizing on geometric properties, it can be observed that, when $\theta \geq 90°$, the projection of gradient $\mathbf{g}_r$ on the normal plane of gradient $\mathbf{g}_c$ is $\mathbf{g}_r^+$, and the projection of gradient $\mathbf{g}_c$ on the normal plane of gradient $\mathbf{g}_r$ is $\mathbf{g}_c^+$. Consequently, under such conditions, the following property is maintained:

$$\mathbf{g}^- = \frac{\mathbf{g}_r + \mathbf{g}_c}{2} = \frac{\frac{\mathbf{g}_r}{\|\mathbf{g}_r\|}\left(\|\mathbf{g}_r\|\right) + \frac{\mathbf{g}_c}{\|\mathbf{g}_c\|}\left(\|\mathbf{g}_c\|\right)}{2}, \tag{10}$$

$$\cos\left(\theta\right) = \frac{\mathbf{g_r} \cdot \mathbf{g_c}}{\|\mathbf{g_r}\|\,\|\mathbf{g_c}\|}, \tag{11}$$

with Equation (7) and Equation (11), we can observe,

$$\mathbf{g} = \frac{\mathbf{g}_r^+ + \mathbf{g}_c^+}{2} = \frac{\left(\mathbf{g}_r - \frac{\mathbf{g}_r \cdot \mathbf{g}_c}{\|\mathbf{g}_c\|^2}\mathbf{g}_c\right) + \left(\mathbf{g}_c - \frac{\mathbf{g}_c \cdot \mathbf{g}_r}{\|\mathbf{g}_r\|^2}\mathbf{g}_r\right)}{2}$$
$$= \frac{\left(\frac{\mathbf{g}_r}{\|\mathbf{g}_r\|}\|\mathbf{g}_r\| - \frac{cos(\theta)\|\mathbf{g}_r\|\|\mathbf{g}_c\|}{\|\mathbf{g}_c\|^2}\mathbf{g}_c\right) + \left(\frac{\mathbf{g}_c}{\|\mathbf{g}_c\|}\|\mathbf{g}_c\| - \frac{cos(\theta)\|\mathbf{g}_c\|\|\mathbf{g}_r\|}{\|\mathbf{g}_r\|^2}\mathbf{g}_r\right)}{2}. \tag{12}$$

Under the condition of $\theta \geq 90°, cos(\theta) \leq 0$, we can observe,

$$\left(-\frac{cos(\theta)\|\mathbf{g}_r\|\|\mathbf{g}_c\|}{\|\mathbf{g}_c\|^2}\right) + \left(-\frac{cos(\theta)\|\mathbf{g}_c\|\|\mathbf{g}_r\|}{\|\mathbf{g}_r\|^2}\right) \geq 0$$
$$\Longrightarrow \|\mathbf{g}\| \geq \|\mathbf{g}^-\|. \tag{13}$$

Hence, when $\theta \geq 90°$, the strategy of Equation (8) proves to be more effective than the strategy of Equation (9) in handling gradient deviations. This indicates that the second strategy can successfully mitigate gradient degradation while addressing conflicting gradients. Specifically, the gradient $\mathbf{g}$ can be considered an equilibrium gradient that strikes a suitable balance between reward optimization and safety constraints. This implies that an increase in the reward or cost expected gradient cannot be achieved by altering its gradient manipulation, given that other gradients remain unmodified.

Under the circumstance where $\theta < 90°$ and $cos(\theta) > 0$, as illustrated in Figure 3 (b), the projection gradients of $\mathbf{g}_r$ on the normal plane of $\mathbf{g}_c$ and $\mathbf{g}_c$ on the normal plane of $\mathbf{g}_r$ yield gradients $\mathbf{g}_r^-$ and $\mathbf{g}_c^-$. Upon observation, it can be noted that,

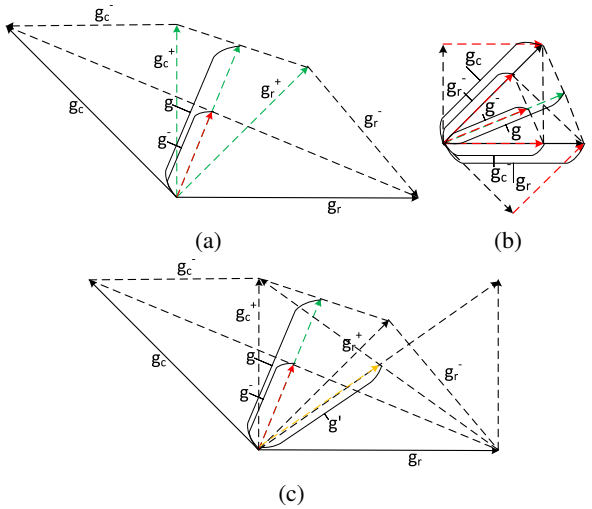

Figure 3: Analysis of Soft switching through gradient manipulation.

$$\mathbf{g}^- = \frac{\mathbf{g}_r^+ + \mathbf{g}_c^+}{2}$$
$$= \frac{\left(\frac{\mathbf{g}_r}{\|\mathbf{g}_r\|}\|\mathbf{g}_r\| - \frac{cos(\theta)\|\mathbf{g}_r\|\|\mathbf{g}_c\|}{\|\mathbf{g}_c\|^2}\mathbf{g}_c\right) + \left(\frac{\mathbf{g}_c}{\|\mathbf{g}_c\|}\|\mathbf{g}_c\| - \frac{cos(\theta)\|\mathbf{g}_c\|\|\mathbf{g}_r\|}{\|\mathbf{g}_r\|^2}\mathbf{g}_r\right)}{2}, \tag{14}$$

with $\theta < 90°, cos(\theta) > 0$, the following property holds,

$$-\left(\frac{cos(\theta)\|\mathbf{g}_r\|\|\mathbf{g}_c\|}{\|\mathbf{g}_c\|^2}\right) - \left(\frac{cos(\theta)\|\mathbf{g}_c\|\|\mathbf{g}_r\|}{\|\mathbf{g}_r\|^2}\right) < 0, \tag{15}$$

$$\mathbf{g} = \frac{\mathbf{g}_r + \mathbf{g}_c}{2} = \frac{\frac{\mathbf{g}_r}{\|\mathbf{g}_r\|}\|\mathbf{g}_r\| + \frac{\mathbf{g}_c}{\|\mathbf{g}_c\|}\|\mathbf{g}_c\|}{2}. \tag{16}$$

Thus, we can observe $\|\mathbf{g}\| > \|\mathbf{g}^-\|$. In this example, $\theta < 90°$, the gradient managed by the second strategy, as illustrated in Equation (8), can be observed in Equation(14). Concurrently, the gradient addressed using the first strategy, as depicted in Equation(9), is presented in Equation (16). Upon examination, it becomes evident that under these conditions, the first strategy surpasses the second strategy in effectively handling deviations in reward and cost gradients. Furthermore, the first strategy is capable of mitigating gradient degradation while simultaneously reducing gradient deviation.

Furthermore, as illustrated in Figure 3 (c), the yellow dashed line signifies the updated gradient, $\mathbf{g}'$, generated by Algorithm 1 of gradient surgery (Yu et al. 2020), the angle, θ^s, between $\mathbf{g}'$ and the original cost gradient, $\mathbf{g}_c$, remains greater than $90°$. This observation implies that the relation between $\mathbf{g}'$ and $\mathbf{g}_c$ continues to exhibit a conflicting nature, which may potentially result in inadequate handling of the original cost gradient. Based on the subsequent analysis as shown in Equation (17), our gradient manipulation approach demonstrates improved performance. It is important to note that Algorithm 1 of gradient surgery (Yu et al. 2020) does not

consider cases where $\theta < 90°$, which might be insufficient for addressing optimization oscillations effectively. In the upcoming experiment section, we also provide ablation experiments to investigate the effectiveness of different gradient manipulation methods.

$$\begin{aligned}\mathbf{g}' &= \frac{\mathbf{g}_r + \mathbf{g}_c^+}{2} = \frac{\mathbf{g}_r + \left(\mathbf{g}_c - \frac{\mathbf{g}_c \cdot \mathbf{g}_r}{\|\mathbf{g}_r\|^2}\mathbf{g}_r\right)}{2}, \\ &= \frac{\frac{\mathbf{g}_r}{\|\mathbf{g}_r\|}\|\mathbf{g}_r\| + \left(\frac{\mathbf{g}_c}{\|\mathbf{g}_c\|}\|\mathbf{g}_c\| - \frac{cos(\theta)\|\mathbf{g}_c\|\|\mathbf{g}_r\|}{\|\mathbf{g}_r\|^2}\mathbf{g}_r\right)}{2}, \\ &\Longrightarrow \|\mathbf{g}\| > \|\mathbf{g}'\|.\end{aligned} \tag{17}$$

A Framework for Safe Reinforcement Learning with Soft Switching

In this section, we present a comprehensive framework referred to as PCRPO, which iteratively optimizes performance until convergence is achieved. As demonstrated in Algorithm 1 in Appendix A, we propose a novel approach with slack techniques to address the deviation of reward and cost gradients, particularly for conflicting gradients.

Case one: In the event that the slack value tends toward infinity, i.e., $h^+ \to +\infty$ and $h^- = 0$, the optimization process is adapted based on the satisfaction of safety constraints. When a safety violation occurs, the optimization exclusively focuses on safety by employing Equation (19), where w is the parameters represented by neural networks, η is the step size of gradient update. Conversely, if safety constraints are satisfied, the optimization process incorporates the projection gradient, as delineated in Equation (21). **Case two:** In the event that the slack value is denoted by $h^+ = 0$ and $h^- \to -\infty$, a safety violation necessitates the enhancement of the reward and concurrent reduction of cost by employing Equation (21). Conversely, when safety requirements are fulfilled, the focus shifts solely to the optimization of reward performance, as demonstrated by Equation (18).

Case three: In situations where the slack value is confined to the range of $+\infty > h^+ > 0$ and $0 > h^- > -\infty$, several circumstances can be observed. If upper slack, lower slack, and safety violations occur simultaneously, the optimization process is devoted solely to addressing safety concerns, as indicated by Equation (19). In the absence of upper slack violations, while lower slack and safety violations transpire concurrently, the strategy involves enhancing the reward and concurrently reducing the cost by employing Equation (21). Conversely, when upper slack and safety violations are not present, but a lower slack violation persists, the same approach of augmenting the reward and minimizing the cost is implemented using Equation (21). Finally, in the absence of violations related to upper slack, lower slack, and safety, the primary focus is directed towards optimizing reward performance, as demonstrated by Equation (18).

$$w_{t+1} = w_t + \eta\bar{\Delta}_t^r, \mathbf{g}^r = \bar{\Delta}_t^r, \tag{18}$$

$$w_{t+1} = w_t - \eta\bar{\Delta}_t^{c_i}, \mathbf{g}^c = -\bar{\Delta}_t^{c_i}, \tag{19}$$

where from Lemma 5.1 of (Agarwal et al. 2021), we have

$$\bar{\Delta}_t^r = (1-\gamma)^{-1}\bar{Q}_t^{r_t}(s,a), \bar{\Delta}_t^{c_i} = (1-\gamma)^{-1}\bar{Q}_t^{c_i,t}(s,a). \tag{20}$$

$$w_{t+1} = w_t + \eta\cdot\left(\frac{\mathbf{g}_r^+ + \mathbf{g}_c^+}{2}\right), w_{t+1} = w_t + \eta\cdot\left(\frac{\mathbf{g}_r + \mathbf{g}_c}{2}\right). \tag{21}$$

Inspired by CRPO (Xu, Liang, and Lan 2021), we implement our algorithm within the context of the primal optimization setting. Similarly, we initially evaluate the policy and subsequently improve it while addressing safety constraints.

Policy Evaluation During the policy evaluation step, the objective is to learn Q-functions that accurately evaluate the previous policy π_t. To accomplish this, we train distinct Q-functions for both reward and constraints.

$$Q_{i,k+1}^{\pi_w}(s,a) = Q_{i,k}^{\pi_w} + \ell_k\left[r_i(s,a) + \gamma Q_{i,k}^{\pi_w}(s',a') - Q_{i,k}^{\pi_w}(s,a)\right], \tag{22}$$

where $s \sim \mu_{\pi_w}, a \sim \pi_w(s), s' \sim \mathrm{P}(\cdot \mid s,a), a' \sim \pi_w(s')$, ℓ_k is the learning rate and i denotes the reward or any of the constraints. $Q_i(s,a)$ can be estimated via $Q_{i,K_{\mathrm{TD}}}^{\pi_w}(s,a)$, where K_{TD} is the iteration number of using TD learning methods.

Policy Improvement for Reward and Safety The policy gradient (Sutton et al. 1999) of the reward value function $f_r(\pi_w)$ has been derived as $\nabla f_r(\pi_w) = \mathbb{E}\left[Q_r^{\pi_w}(s,a)\phi_w(s,a)\right]$, where $\phi_w(s,a) := \nabla_w \log \pi_w(a \mid s)$ is the score function. Similarly, for the value function of cost i, we have $\nabla f_{c_i}(\pi_w) = \mathbb{E}\left[Q_{c_i}^{\pi_w}(s,a)\phi_w(s,a)\right]$.

In scenarios where the optimization of both reward and safety i is desired, it is necessary to select a non-conflicting gradient descent $\boldsymbol{d}$ on the natural gradients of reward and cost gradients. This selection aims to optimize reward and safety individually, subject to the constraint that the KL divergence between the updated and previous policy remains below a specified threshold.

$$\boldsymbol{d} = \frac{\mathbf{g}_r^+ + \mathbf{g}_{c_i}^+}{2} \text{ or } \frac{\mathbf{g}_r + \mathbf{g}_{c_i}}{2}. \tag{23}$$

Correlation-Reduction for Stochastic Gradient Manipulation In practical applications, the challenge of acquiring imprecise policy gradient feedback is frequently encountered. This imprecision stems from the restricted number of sampled trajectories employed to estimate $Q_r^{\pi_{w_t}}$ or $Q_{c_i}^{\pi w_t}$, subsequently introducing stochastic noise into the system. The study conducted in (Zhou et al. 2022) has revealed that, within a stochastic setting, conventional gradient manipulation techniques may fail to converge to a optimal solution.

Let the weight factors $\boldsymbol{\lambda}t = (\lambda_t^r, \lambda_t^{c_1}, \ldots, \lambda_t^{c_n})$ be represented as $\boldsymbol{d} = \lambda_t^r\mathbf{g}_r + \lambda_t^{c_i}\mathbf{g}_{c_i}$ for the time step t. The primary cause of this convergence failure lies in the substantial correlation between the weight factors $\boldsymbol{\lambda}t$ and the stochastic gradients, resulting in a biased composite gradient. To address this issue within the context of the PCRPO framework, we concentrate on the specific conditions that ensure the variance of the natural policy gradient estimator progressively approaches zero. One possible approach to achieve this involves utilizing TD learning in (22) to estimate $Q^{\pi w_t}i$, assuming K_{TD} is adequately large.

Comparison to CRPO

Compared to CRPO (Xu, Liang, and Lan 2021), a primal safe RL algorithm, our algorithm exhibits two distinct differences:

the addition of upper and lower slack values to the constraint thresholds, and the unique approach we take to optimize the policy concerning both reward and safety. CRPO focuses on optimizing for reward, only shifting to safety optimization if a safety constraint is hard violated. This can lead to a back-and-forth between reward and cost optimizations, particularly when constraints are near their threshold boundaries. To circumvent such oscillations and to prevent any performance degradation in other objectives, we employ a projected gradient descent approach. This ensures a balanced and efficient way of handling both reward and safety concerns.

Convergence Analysis

For the iterates π_{w_t} manipulated using our proposed method, **we can guarantee performance monotonic improvement and convergence to the optimal performance in cases where** $180° > \theta \geq 0°$. Our theorem enables the derivation of other settings in a straightforward manner. Please refer to Appendix B for detailed theorems and their corresponding proofs.

Experiments

In the experimental section, we investigate the constraint satisfaction of policies trained using our proposed method and compare its performance with state-of-the-art (SOTA) safe RL algorithms. Employing the developed benchmark, *Safety-MuJoCo Benchmark*, we first compare our approach with a representative primal optimization-based safe RL method, CRPO (Xu, Liang, and Lan 2021), a strong baseline established in 2021. CRPO demonstrates superior performance in comparison to several SOTA baselines such as PDO (Ray, Achiam, and Amodei 2019).

To further emphasize the effectiveness of our method, we implement our algorithm on a popular benchmark, *Omnisafe* (Ji et al. 2023), and compare it with several representative primal-dual optimization-based safe RL methods. These methods include SOTA baselines, such as PCPO (Yang et al. 2020), CUP (Yang et al. 2022), and PPO Lagrangian (PPO-Lag) (Ji et al. 2023). By contrasting our approach with these established methods, we aim to demonstrate the potential advantages and improvements that our proposed method offers in terms of safety and efficiency for RL applications with safety constraints.

In particular, our proposed method belongs to the category of primal optimization-based safe RL approaches, thereby avoiding the challenges associated with hyperparameter tuning related to dual variables. Additionally, our method does not necessitate feasible initialization, in contrast to some primal-dual optimization-based methods where poor initialization can adversely impact performance (Xu, Liang, and Lan 2021). By circumventing these challenges, our proposed method aims to provide a more reliable and efficient solution for ensuring safety in RL applications. Additionally, we conduct ablation studies to investigate the impact of diverse cost limits and sensitivity to slack bounds.

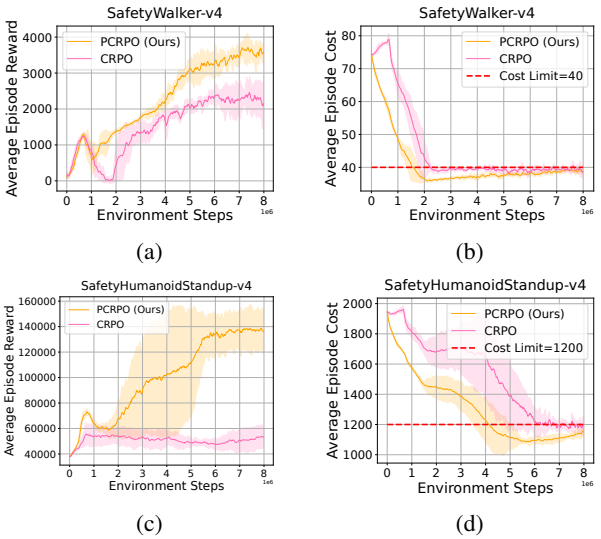

Figure 4: Compared with CRPO on the SafetyWalker and SafetyHumanoidStandup Tasks. To encourage more learning exploration, we initiate the optimization of safety after 640000 steps.

Experiments on *Safety-MuJoCo Benchmark*

We have developed a *Safety-MuJoCo Benchmark*[1] based on MuJoCo (Todorov, Erez, and Tassa 2012) to evaluate the performance of safe RL algorithms. This benchmark differs from traditional safe RL benchmarks, such as *Omnisafe*[2] (Ji et al. 2023) that is developed based on Safety Gym[3] (Ray, Achiam, and Amodei 2019). In *Omnisafe*, the cost constraint is set as the velocity limit, and the reward is determined by the speed at which the robot runs. Conversely, our benchmark considers not only velocity constraints but also the health of the robot. For example, whether the robot falls or whether its joints exceed the limit values of motion control are taken into account. For a comprehensive description of these considerations, please refer to Appendix C. As depicted in Figure 4 (a) and (b), our method demonstrates remarkably superior performance in comparison to CRPO with respect to both reward maximization and safety preservation. Similarly, Figure 4 (c) and (d) illustrate our method's significant improvement over CRPO in terms of reward and cost performance.

Experiments on *Omnisafe* Benchmark

We implement our algorithm on *Omnisafe* and compare its performance with several SOTA baselines within the *Omnisafe* framework. The safety constraint is set as a constant threshold, where if the agent moves at a higher velocity than this threshold, it incurs a cost of 1 per time step. We test our method, PCRPO, alongside PCPO, CUP, and PPOLag methods on various environments, including Hopper and Ant.

As illustrated in Figure 5 (a) and (b), on the SafetyHopperVelocity-v1 task, our algorithm exhibits supe-

[1]https://github.com/SafeRL-Lab/Safety-MuJoCo.git
[2]https://github.com/PKU-Alignment/omnisafe.git
[3]https://github.com/openai/safety-gym.git

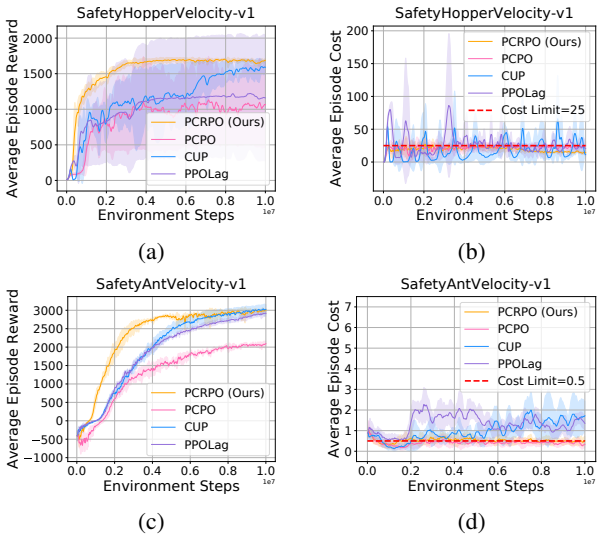

Figure 5: Compared with PCPO, CUP, PPOLag baselines on SafetyHopperVelocity-v1 and SafetyAntVelocity-v1 tasks.

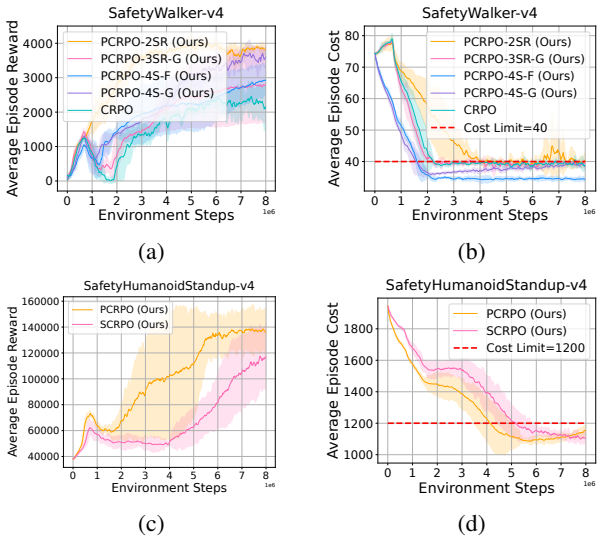

Figure 6: (a & b) Ablation experiments of different slack settings on the SafetyWalker task. (c & d) Ablation experiments of different gradient manipulation methods on the SafetyHumanoidStandup task. To encourage more learning exploration, we initiate the safety optimization after 640000 steps on the SafetyWalker task.

rior reward performance compared to SOTA baselines and maintains reliable safety. In contrast, SOTA baselines such as CUP and PPOLag struggle to ensure safety, and their reward performance is worse than our algorithm. Notably, our approach outperforms PCPO in both reward and safety performance. In Figure 5 (c) and (d), our algorithm effectively ensures complete safety on the SafetyAntVelocity-v1 task while achieving comparable reward performance. Specifically, our algorithm demonstrates greater safety than CUP and PPOLag, which can not ensure safety on the task. While PCPO can also ensure safety, its reward performance is inferior to our algorithm. Furthermore, our algorithm demonstrates faster convergence than the baselines.

Ablation Experiments

Ablation Experiments of Slack Settings As illustrated in Figures 6 (a) and (b), we perform an ablation study on various slack settings. PCRPO-2SR represents $h_i^+ \to +\infty, h_i^- = 0$, where we primarily optimize reward while slightly ensuring safety. PCRPO-3SR-G denotes $h_i^+ = 20, h_i^- = 0$, with h_i^+ gradually decreasing to zero as the number of iteration steps increases. In this setting, we aim to optimize reward and safety simultaneously when $(b_i + h_i^+) > C_i > b_i$. PCRPO-4S-F corresponds to $h_i^+ = 20, h_i^- = -20$, where we optimize safety and reward at the static slack boundary. As the experiment results show, our algorithm's cost value converges to the boundary. PCRPO-4S-G represents $h_i^+ = 20, h_i^- = -20$, with h_i^+ and h_i^- gradually decreasing to zero as the number of iteration steps increases. In this setting, we optimize safety and reward at the dynamic slack boundary, and as demonstrated by the experimental results, our algorithm's cost value converges to the cost limit while maintaining good reward performance. Notably, all our algorithms demonstrate superior results compared to CRPO in terms of balancing reward and safety optimization. This experimental setup allows us to analyze the impact of various slack configurations on the performance of our method, providing insights into the balance between reward and safety optimization.

Ablation Experiments of Gradient Manipulation Methods As depicted in Figures 6 (c) and (d), we employ the gradient manipulation technique as outlined in Algorithm 1 of gradient surgery (Yu et al. 2020) for learning safety, named as SCRPO. The experimental results demonstrate that our proposed approach outperforms SCRPO in terms of safety and reward performance. These findings further corroborate the consistency of our theoretical analysis presented in the gradient analysis section.

Conclusion

In this study, we address the issue of gradient conflicts between reward and cost by employing gradient manipulation. Specifically, we first propose a novel solution called PCRPO, which incorporates soft switching to balance reward and safety optimization in safe RL. Moreover, a slack technique is developed to help alleviate the conflict between reward and safety optimization. Our theoretical analysis demonstrates that our method can guarantee performance monotonic improvement while also analyzing the upper and lower bounds of the performance update. Then, we evaluate the effectiveness of our method using the *Safety-MuJoCo Benchmark* that we developed, as well as a popular safe RL benchmark, *Omnisafe*. Finally, the experimental results show that our method outperforms the strong baselines, indicating its superior performance in addressing the challenges associated with safe RL.

References

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

Agarwal, A.; Kakade, S. M.; Lee, J. D.; and Mahajan, G. 2021. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *The Journal of Machine Learning Research*, 22(1): 4431–4506.

Boyd, S. P.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Brunke, L.; Greeff, M.; Hall, A. W.; Yuan, Z.; Zhou, S.; Panerati, J.; and Schoellig, A. P. 2022. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5: 411–444.

Calian, D. A.; and et al. 2020. Balancing Constraints and Rewards with Meta-Gradient D4PG. In *ICLR*.

Chow, Y.; Nachum, O.; Duenez-Guzman, E.; and Ghavamzadeh, M. 2018. 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 preprint arXiv:1901.10031*.

Fulton, N.; and Platzer, A. 2018. Safe reinforcement learning via formal methods: Toward safe control through proof and learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Gu, S.; Chen, G.; Zhang, L.; Hou, J.; Hu, Y.; and Knoll, A. 2022a. Constrained reinforcement learning for vehicle motion planning with topological reachability analysis. *Robotics*, 11(4): 81.

Gu, S.; Kshirsagar, A.; Du, Y.; Chen, G.; Peters, J.; and Knoll, A. 2023a. A human-centered safe robot reinforcement learning framework with interactive behaviors. *Frontiers in Neurorobotics*, 17.

Gu, S.; Kuba, J. G.; Chen, Y.; Du, Y.; Yang, L.; Knoll, A.; and Yang, Y. 2023b. Safe multi-agent reinforcement learning for multi-robot control. *Artificial Intelligence*, 319: 103905.

Gu, S.; Yang, L.; Du, Y.; Chen, G.; Walter, F.; Wang, J.; Yang, Y.; and Knoll, A. 2022b. A review of safe reinforcement learning: Methods, theory and applications. *arXiv preprint arXiv:2205.10330*.

Ji, J.; Zhou, J.; Zhang, B.; Dai, J.; Pan, X.; Sun, R.; Huang, W.; Geng, Y.; Liu, M.; and Yang, Y. 2023. OmniSafe: An Infrastructure for Accelerating Safe Reinforcement Learning Research. *arXiv preprint arXiv:2305.09304*.

Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A. A.; Yogamani, S.; and Pérez, P. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6): 4909–4926.

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

Koller, T.; Berkenkamp, F.; Turchetta, M.; and Krause, A. 2018. Learning-based model predictive control for safe exploration. In *2018 IEEE conference on decision and control (CDC)*, 6059–6066. IEEE.

Li, X.; and Belta, C. 2019. Temporal logic guided safe reinforcement learning using control barrier functions. *arXiv preprint arXiv:1903.09885*.

Nocedal, J.; and Yuan, Y.-x. 1998. Combining trust region and line search techniques. In *Advances in Nonlinear Programming: Proceedings of the 96 International Conference on Nonlinear Programming*, 153–175. Springer.

Ray, A.; Achiam, J.; and Amodei, D. 2019. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7(1): 2.

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.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Sui, Y.; Gotovos, A.; Burdick, J.; and Krause, A. 2015. Safe exploration for optimization with Gaussian processes. In *International conference on machine learning*, 997–1005. PMLR.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 5026–5033. IEEE.

Xu, T.; Liang, Y.; and Lan, G. 2021. Crpo: A new approach for safe reinforcement learning with convergence guarantee. In *International Conference on Machine Learning*, 11480–11491. PMLR.

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. In *International Conference on Learning Representations*.

Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33: 5824–5836.

Zhou, S.; Zhang, W.; Jiang, J.; Zhong, W.; Gu, J.; and Zhu, W. 2022. On the Convergence of Stochastic Multi-Objective Gradient Manipulation and Beyond. *Advances in Neural Information Processing Systems*, 35: 38103–38115.

Zhou, Z.; and et al. 2023. Gradient-adaptive pareto optimization for constrained reinforcement learning. In *AAAI*, volume 37, 11443–11451.