Gradient Shaping for Multi-Constraint Safe Reinforcement Learning

Yihang Yao[1] YIHANGYA@ANDREW.CMU.EDU
Zuxin Liu[1] ZUXINL@ANDREW.CMU.EDU
Zhepeng Cen[1] ZCEN@ANDREW.CMU.EDU
Peide Huang[1] PEIDEH@ANDREW.CMU.EDU
Tingnan Zhang[2] TINGNAN@GOOGLE.COM
Wenhao Yu[2] MAGICMELON@GOOGLE.COM
Ding Zhao[1] DINGZHAO@ANDREW.CMU.EDU

[1]***Carnegie Mellon University***, [2]***Google Deepmind***

Editors: A. Abate, K. Margellos, A. Papachristodoulou

Abstract

Online safe reinforcement learning (RL) involves training a policy that maximizes task efficiency while satisfying constraints via interacting with the environments. In this paper, our focus lies in addressing the complex challenges associated with solving multi-constraint (MC) safe RL problems. We approach the safe RL problem from the perspective of Multi-Objective Optimization (MOO) and propose a unified framework designed for MC safe RL algorithms. This framework highlights the manipulation of gradients derived from constraints. Leveraging insights from this framework and recognizing the significance of *redundant* and *conflicting* constraint conditions, we introduce the Gradient Shaping (GradS) method for general Lagrangian-based safe RL algorithms to improve the training efficiency in terms of both reward and constraint satisfaction. Our extensive experimentation demonstrates the effectiveness of our proposed method in encouraging exploration and learning a policy that improves both safety and reward performance across various challenging MC safe RL tasks as well as good scalability to the number of constraints. The full paper with the appendix is available on our website: https://sites.google.com/view/mc-grads/home.

Keywords: Safe Reinforcement Learning, Multi-Objective Optimization, Multi-task Learning

1. Introduction

Despite the great success of deep reinforcement learning (RL) in recent years (Levine et al., 2020; Silver et al., 2017; Brunke et al., 2022; Li, 2023), ensuring safety (i.e., constraint satisfaction) is one key challenge when deploying them to real-world applications (Hu et al., 2023; Liu et al., 2022; Zhao et al., 2021; Xu et al., 2022; Wachi and Sui, 2020; Zhang et al., 2023). Safe RL has been a common approach to address the difficulties faced by agents operating in complex and safety-critical tasks (Gu et al., 2022; Thananjeyan et al., 2021; Zhang et al., 2020; Zhao et al., 2023; Cheng et al., 2023; Wachi et al., 2021), such as autonomous driving (Isele et al., 2018; Hsu et al., 2023a), home service (Ding et al., 2022; Hsu et al., 2023b), legged robots (Kim et al., 2023b), and UAV locomotion (Qin et al., 2021; Zheng et al., 2021). Safe RL aims to maximize the cumulative reward within a constrained policy set (Yang et al., 2022; Thananjeyan et al., 2021; Bharadhwaj et al., 2020; Khattar et al., 2022; Yao et al., 2023; Ma et al., 2022). By explicitly incorporating safety constraints into the policy learning process, agents can adeptly navigate the trade-off between task performance and safety constraints, rendering them well-suited for real-world tasks. (Brunke et al., 2021; Yao et al., 2023).

In real-world applications, agents often face multiple constraints (Kim et al., 2023b; Lin et al., 2024). For example, an autonomous driving vehicle must avoid collisions, prevent over-speeding, stay on the road, and adhere to various traffic rules and social norms simultaneously (Feng et al., 2023). Nevertheless, despite the advancements in safe RL, the development of algorithms for MC safe learning that can effectively handle multiple costs remains a challenging issue (Kim et al., 2023a). Many existing methods only consider a single constraint during training (Achiam et al., 2017). The extension of the Lagrangian method to MC settings is a potential solution. However, such approaches can be sensitive to the initialization of Lagrange multipliers and the learning rate, leading to extensive hyperparameter tuning costs (Xu et al., 2021; Achiam et al., 2017; Chow et al., 2019). Furthermore, these methods may introduce instability issues in scenarios with multiple constraints, thus limiting their scalability. CRPO method (Xu et al., 2021) has been proposed to randomly select one constraint for policy consideration at each step to handle multiple constraints. Unfortunately, considering one constraint at a time becomes inefficient with an increasing number of constraints.

Empirical findings have indicated that MC safe RL poses more challenges compared to single-cost settings (Liu et al., 2023a; Kim et al., 2023a). In this study, we analyze the MC safe RL problem through the lens of constraint types, identifying two challenging MC safe RL settings: *redundant* and *conflicting* constraints. To address these challenges, we propose the constraint gradient shaping (GradS) technique from the standpoint of Multi-Objective Optimization (MOO), ensuring compatibility with general Lagrangian-based safe RL algorithms. The main contributions are summarized as follows:

1. We introduce a unified framework for Lagrangian-based MC safe RL algorithms from the perspective of Multi-Objective Optimization (MOO). Within this framework, the major difference among Lagrangian-based MC safe RL methods is the strategy dealing with gradients induced by constraints.

2. We propose the gradient shaping (GradS) method for MC safe RL algorithms. The proposed method can tackle the challenging *redundant* and *conflicting* MC safe RL settings. Our theoretical analysis further provides insights into the convergence of our approach.

3. We conduct extensive evaluations of our method: The proposed GradS method and baselines are evaluated on the MC safe RL tasks modified from common safe RL benchmarks `Bullte-Safety-Gym` (Gronauer, 2022) and `Safety-Gymnasium` (Ji et al., 2023). The results demonstrate that GradS can significantly improve safety and reward performance in MC tasks.

2. Related Work

Safe RL has been approached through various methods. Researchers have proposed many techniques employing constrained optimization techniques to learn a constraint-satisfaction policy (Garcıa and Fernández, 2015; Gu et al., 2022; Flet-Berliac and Basu, 2022), such as the Lagrangian-based approach (Bhatnagar and Lakshmanan, 2012; Chow et al., 2017; As et al., 2022; Ding and Lavaei, 2023), where the Lagrange multipliers can be optimized along with the policy parameters (Liang et al., 2018; Tessler et al., 2018; Ray et al., 2019). Alternatively, some works approximate the constrained RL problem with Taylor expansions (Achiam et al., 2017) or through variational inference (Liu et al., 2022). They then solve for the dual variable using convex optimization (Yu et al., 2019; Yang et al., 2020; Gu et al., 2021; Kim and Oh, 2022). For MC settings, many works propose to consider all the constraints equally (Fernando et al., 2022; As et al., 2022), some techniques consider the constraints that violate the most, and other methods randomly activate one constraint for policy update. One recent concurrent work (Kim et al., 2023a) proposes the gradient integration method to manage infeasibility issues in MC Safe RL. However, this method is limited to the TRPO-based methods and is hard to generalize to other algorithms. The systematical analysis for MC safe RL is still a largely unexplored area.

Multi-Objective Optimization (MOO) considers how to train a single model that can meet a variety of different requirements (Huang et al., 2022; Yu et al., 2020; Caruana, 1997). The MOO formulation has been extended to many different settings, including supervised learning (Yang and Hospedales, 2016; Zamir et al., 2018), and reinforcement learning (Wilson et al., 2007; Sodhani et al., 2021). For the Multi-Objective RL (MORL), existing works learn a policy that is optimal in the Pareto Frontier with a given trading-off among tasks (Roijers et al., 2013; Zhang and Golovin, 2020). In recent years, researchers also interpreted safe RL from the perspective of MORL. However, they are primarily focusing on multiple task rewards and preference settings (Huang et al., 2022) and single-constraint settings (Liu et al., 2023b), but not particular MC safe RL problems.

3. Unified Framework for MC Safe RL

In this section, we introduce the proposed unified framework for Lagrangian-based MC safe RL.

3.1. Preliminary

Constrained Markov Decision Process (CMDP) $\mathcal{M}$ is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \boldsymbol{c}, \mu_0)$ (Altman, 1998), where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, and $\mu_0 : \mathcal{S} \to [0, 1]$ is the initial state distribution. CMDP augments MDP with an additional element $\mathbf{c} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^N_{\geq 0}$ to characterize the cost of violating the constraint, where N is the cost dimension. An MC safe RL problem is specified by a CMDP and a constraint threshold vector $\boldsymbol{\epsilon} \in \mathbb{R}^N_{\geq 0}$. Let $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ denote the policy and $\tau = \{s_1, a_1, ...\}$ denote the trajectory. The value functions are $V^\pi_r(\mu_0) = \mathbb{E}_{\tau\sim\pi,s_0\sim\mu_0}[\sum_{t=0}^{\infty} \gamma^t r(t)], \boldsymbol{V}^\pi_{c_i}(\mu_0) = \mathbb{E}_{\tau\sim\pi,s_0\sim\mu_0}[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{c}_i(t)], i = 1, 2, ..., N$, which is the expectation of discounted return under the policy π and the initial state distribution μ_0. Denote $\preceq$ as an element-wise partial order, the goal of MC safe RL is to find the policy that maximizes the reward return while constraining the cost return under the pre-defined threshold $\boldsymbol{\epsilon}$:

$$\pi^* = \arg\max_\pi V^\pi_r, \quad s.t. \quad \boldsymbol{V}^\pi_{\boldsymbol{c}} \preceq \boldsymbol{\epsilon}, \quad (\boldsymbol{V}^\pi_{\boldsymbol{c}} \in \mathbb{R}^N_{\geq 0}, \ \boldsymbol{\epsilon} \in \mathbb{R}^N_{\geq 0}). \tag{1}$$

To solve this problem, Lagrangian-based safe RL algorithms can be formulated to find:

$$(\pi^*, \boldsymbol{\lambda}^*) = \arg\max_\lambda \min_\pi \mathcal{J}(\pi, \boldsymbol{\lambda}), \quad \mathcal{J}(\pi, \boldsymbol{\lambda}) = -V^\pi_r + \boldsymbol{\lambda}^T(\boldsymbol{V}^\pi_{\boldsymbol{c}} - \boldsymbol{\epsilon}) \tag{2}$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, ..., \lambda_N]^T$ is the Lagrangian multiplier corresponding to the primary problem (1). In practice, we can update $(\pi, \boldsymbol{\lambda})$ iteratively (Stooke et al., 2020).

3.2. Unified framework: MC Safe RL as MOO

In multi-objective optimization (MOO), we are given $K \geq 2$ different tasks, each associated with a loss function (Fernando et al., 2022). With this, at t-th step, updating π_t via solving (2) is to find:

$$\pi^*_t = \arg\min_{\pi_t} \left[-V^{\pi_t}_r + \boldsymbol{\lambda}^T_{\boldsymbol{t}}(\boldsymbol{V}^{\pi_t}_{\boldsymbol{c}} - \boldsymbol{\epsilon})\right], \tag{3}$$

For simplicity, we will omit the subscript t and superscript π in the following. The gradient ∇J for policy π is:

$$\nabla J = -\nabla V_r + \nabla J_c, \quad \nabla J_c = \boldsymbol{w}^T \boldsymbol{G}, \tag{4}$$

where $\boldsymbol{G} := [g_1, ..., g_N]$ is the constraint gradient vector, $g_i = \boldsymbol{\lambda}_i \nabla \boldsymbol{V}_{\boldsymbol{c}_i}$ is the i-th constraint gradient, and $\boldsymbol{w} \succeq \boldsymbol{0}$ is a non-negative weight vector of the constraint gradients. With this formulation, many commonly used methods for MC Safe RL can be categorized as:

(1) Vanilla Method: For common safe RL algorithms (Fernando et al., 2022; As et al., 2022), they consider all the constraints equally, with a uniform weight:

$$\boldsymbol{w} = \boldsymbol{1} \tag{5}$$

(2) CRPO[1] Method: Methods such as CRPO (Xu et al., 2021) that randomly select constraints for policy update at each time can be formulated as:

$$\|\boldsymbol{w}\|_0 = 1, \quad \boldsymbol{w}_i = 1,\ i \sim \text{uniform}(1, N) \tag{6}$$

(3) Min-Max method: Safe RL methods that penalize the cost that violates the constraint the most for policy updates at each time can be formulated as:

$$\|\boldsymbol{w}\|_0 = 1, \quad \boldsymbol{w}_{i^*} = 1,\ i^* = \arg\max(\boldsymbol{V}_{c_i} - \boldsymbol{\epsilon}_i) \tag{7}$$

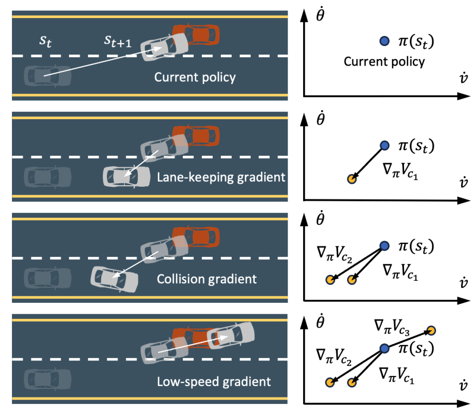


Figure 1: Illustration of constraint types. c_1 is the lane-keeping cost, c_2 is the collision avoidance cost, and c_3 is the low-speed cost.

4. Gradient Shaping for MC Safe RL

Based on empirical findings in both previous works (Liu et al., 2023a; Kim et al., 2023a) and this work, MC safe RL presents greater difficulty compared to single-constraint ones. Thus, before delving into the proposed method, we outline the critical conditions essential for understanding MC safe RL, particularly focusing on various constraint types.

4.1. Constraint Types in MC Safe RL

Based on the constraint gradient similarity, we define the relationship between two distinct constraints $\boldsymbol{V}^\pi_{\boldsymbol{c}_i} \le \boldsymbol{\epsilon}_i$ and $\boldsymbol{V}^\pi_{\boldsymbol{c}_j} \le \boldsymbol{\epsilon}_j$ for $i \neq j$ given a policy π. Note that the gradients are closely related to the current policy π. We utilize the cosine similarity, which has been used in many previous works (Du et al., 2018), as the similarity function $sim(\cdot, \cdot)$. Denote θ as the parameter for the policy π.

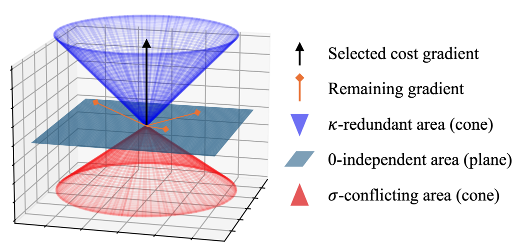


Figure 2: Illustration of elimination area.

Definition 1 (σ-conflicting constraints) *The constraints $\boldsymbol{V}^\pi_{\boldsymbol{c}_i} \le \boldsymbol{\epsilon}_i$ and $\boldsymbol{V}^\pi_{\boldsymbol{c}_j} \le \boldsymbol{\epsilon}_j$ are σ-conflicting constraints if and only if:*

$$sim(\nabla_\theta \boldsymbol{V}^\pi_{\boldsymbol{c}_i}, \nabla_\theta \boldsymbol{V}^\pi_{\boldsymbol{c}_j}) \le -\sigma, \tag{8}$$

Conflicting constraints drive the policy in conflicting directions if both are activated.

1. We modify the original CRPO to a Lagrangian version. Please refer to the experiment and appendix for more details.

Definition 2 (**κ-redundant constraints**) *The constraints $\boldsymbol{V}_{\boldsymbol{c}_i}^{\pi} \leq \boldsymbol{\epsilon}_i$ and $\boldsymbol{V}_{\boldsymbol{c}_j}^{\pi} \leq \boldsymbol{\epsilon}_j$ are κ-redundant constraints if and only if:*

$$sim(\nabla_\theta \boldsymbol{V}_{\boldsymbol{c}_i}^{\pi}, \nabla_\theta \boldsymbol{V}_{\boldsymbol{c}_j}^{\pi}) \geq \kappa, \tag{9}$$

Redundant constraints drive the policy in almost the same direction if both are activated.

Definition 3 (**η-independent constraints**) *The constraints $\boldsymbol{V}_{\boldsymbol{c}_i}^{\pi} \leq \boldsymbol{\epsilon}_i$ and $\boldsymbol{V}_{\boldsymbol{c}_j}^{\pi} \leq \boldsymbol{\epsilon}_j$ are η-independent constraint if and only if:*

$$-\eta \leq sim(\nabla_\theta \boldsymbol{V}_{\boldsymbol{c}_i}^{\pi}, \nabla_\theta \boldsymbol{V}_{\boldsymbol{c}_j}^{\pi}) \leq \eta, \tag{10}$$

Independent constraints drive the policy in "independent" directions if both are activated.

For simplicity, we will omit the subscript θ in the following context. With the toy example in Figure. 1, we illustrate the aforementioned *redundant* and *conflicting* constraints, which are two primary optimization issues in MC safe RL. In this common autonomous driving scenario, we consider three constraints: the lane-keeping constraint to keep the car on the lane, the collision constraint to prevent accidents with other vehicles, and the minimum speed limit constraint to prevent congestion. In the case shown in Figure. 1, for current policy, the lane-keeping constraint c_1 and the collision constraint c_2 are redundant, while c_1 and the low-speed constraint c_3 are conflicting. Notably, *redundant* and *conflicting* constraints are not inherently problematic. In fact, simply averaging constraint gradients should lead to the optimal policy for MC safe RL problems. However, for online safe RL algorithms, *redundant* constraints lead to over-conservativeness by over-estimating the effect of constraints, while *conflicting* constraints result in exploration instability as getting stuck in local optimum, both of which are detrimental to online safe RL agent learning.

Algorithm 1 Gradient Shaping (GradS)

Input: policy π

Output: shaped constraint gradient ∇J_c

Shuffling the constraint indices
▷ *Initialize the candidate gradient set*
$\mathcal{G} \leftarrow \{g_1 := \lambda_1 \nabla V_{c_1}\}$
▷ *Get the candidate gradient set $\mathcal{G}$*
for $i = 2, ..., n$ **do**
    if $-\sigma < sim(i, j) < \kappa,\ \forall j \in \{\mathcal{G}\}$ **then**
        ▷ *Add this constraint into the set*
        $\mathcal{G} \leftarrow \mathcal{G} \cup \{g_i := \lambda_i \nabla V_{c_i}\}$
    end if
end for
▷ *Select constraint gradient*
$g_c \sim \text{uniform}(\mathcal{G})$
Return: $\nabla J_c = \nabla V_c^G = g_c\, |\mathcal{G}|/N$

4.2. Gradient Shaping

The objective of our approach is to address the challenges posed by *redundant* and *conflicting* constraints, aiming to eliminate over-conservativeness resulting from *redundant* constraints and escape local optima to resolve *conflicting* constraints. In this section, we outline our strategy for shaping the constraint gradients. We also provide a theoretical analysis demonstrating that GradS still guarantees convergence in the next section. The core idea for GradS is to first get a candidate constraint gradient set $\mathcal{G}$ via eliminating the *redundant* and *conflicting* constraints, then randomly select one constraint gradient in set $\mathcal{G}$ for policy update. The proposed algorithm operates as follows:

(1) Initially, it shuffles the constraint gradients and computes the cosine similarity between each pair of constraint gradients ∇V_{c_i} and ∇V_{c_j}. (2) Next, it initializes the candidate gradient set with the first gradient $\mathcal{G} \leftarrow \{g_1 := \lambda_1 \nabla V_{c_1}\}$. (3) It then selects gradients sequentially: if a newly chosen gradient g_i is neither κ-redundant nor γ-conflicting with any other gradient in the set $\mathcal{G}$, it is added

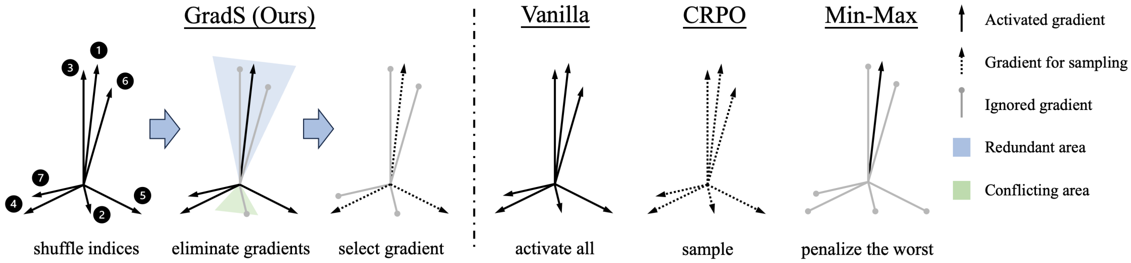


Figure 3: Illustration for constraint gradients shaping.

to the set $\mathcal{G} \leftarrow \mathcal{G} \cup \{g_i := \lambda_i \nabla V_{c_i}\}$. Otherwise, it skips this constraint. (4) After the selection process, the constraint candidate set $\mathcal{G}$ is obtained. Then it randomly samples a gradient from $\mathcal{G}$, and multiplied by a scaling factor as the constraint gradient $V_c^G = g_{\tilde{i}}\, |\mathcal{G}|/N$, where $\tilde{i}$ denotes the index for the selected constraint $\tilde{i} \sim \text{uniform}(\mathcal{G}), g_{\tilde{i}} = \mathcal{G}[\tilde{i}]$. The scaling term $|\mathcal{G}|/N$ is used to ensure stability. The process is described in Algorithm 1. The illustration of the proposed GradS method and the comparison with baseline methods are shown in Figure 3.

The GradS method, although straightforward, mitigates constraints by excluding *redundant* and *conflicting* gradients, which induce over-conservativeness and exploration issues for online safe RL, and selects cost gradients that are *independent*, which makes the policy update more efficient as well as considering most cost information as shown in Figure. 2. Moreover, it encourages exploration by sampling from the gradients after the elimination process instead of aggregating them. In practice, the GradS method can be applied to general Lagrangian-based safe RL algorithms (discussed in this paper) and has the potential for extension to general safe RL algorithms. The proposed GradS method also falls into the framework (4) as to find the weight $\boldsymbol{w}$:

$$\|\boldsymbol{w}\|_0 = 1, \quad \boldsymbol{w}_i = |\mathcal{G}|/N, \quad i \sim \text{uniform}(\text{index}(\mathcal{G})), \tag{11}$$

where $\mathcal{G}$ is the set for candidate cost gradients as mentioned above and shown in Alg. 1, and the sampling "$\sim$" means to sample from the corresponding indices of gradients in the candidate set.

4.3. Theoritical analysis

In this section, we theoretically analyze the performance of GradS with the convergence guarantee. We first have these two common assumptions in safe RL:

Assumption 1 (Slater's condition) *The feasible policy exists, i.e., $\exists \pi$, such that $\boldsymbol{V}_{\boldsymbol{c}}^{\pi} \preceq \boldsymbol{\epsilon}$.*

The feasibility assumption ensures that the Lagrangian $\boldsymbol{\lambda}$ corresponding to the optimization problem (2) is bounded.

Assumption 2 (Bounded and smooth gradients) *Assuming the constraint gradient components are bounded and smooth, i.e., for some constants $G, L > 0$,*

$$\|\nabla V_{c_i}\| \leq G, \quad \left|\boldsymbol{u}^T \nabla^2 V_{c_i} \boldsymbol{u}\right| \leq L\|\boldsymbol{u}\|^2, \ \forall \boldsymbol{u} \in \mathbb{R}^d \tag{12}$$

where $\mathbb{R}^d$ characterizes the policy gradient space. With these mild assumptions, we can ensure that the constraint gradient after GradS is still bounded as shown in Theorem 4.

Theorem 4 (Convergence analysis) *Denote the number of removed κ-redundant and σ-conflicting constraints at iteration time step t as $N_R(\kappa, t), N_C(\sigma, t)$, the total optimized time step as T, the learning rate for every optimization step is α, then for the safety performance, i.e., if we only consider constraint gradient $\nabla V_c^G(\theta_t)$, the policy gradient can be bounded as*

$$\mathbb{E}_t\left[\left\|\nabla V_c^G(\theta_t)\right\|^2\right] \leq \frac{V_c(\theta_0) - V_c^*}{T\alpha} + G^2\left(\mathbb{E}_t\left[N_R(\kappa,t)\right] + \mathbb{E}_t\left[N_C(\sigma,t)\right]\right) + \frac{\alpha G^2 L}{2} \tag{13}$$

The proof is available in the appendix. This bound consists of three terms. The first term relates to the initialization parameters, the second term arises from the elimination of *redundant* and *conflicting* constraints, and the third term is due to the sampling of gradients from the candidate set. The last two terms result from the proposed GradS, which are our "noise ball" terms: the terms that are in some sense "causing" GradS to converge not to a point with zero gradients but rather to some reason nearby, thus we can improve the learning efficiency by avoiding getting stuck in local optimum.

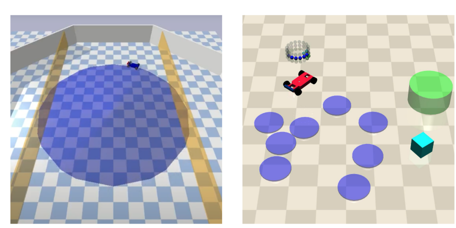
5. Experiments

We aim to address three primary questions in the experiment section: (a) Can baseline methods effectively learn policies that are both safe and rewarding in the challenging MC tasks? (b) How does the proposed GradS method perform in the MC environments? (c) What is the scalability of the proposed GradS method concerning the number of constraints in safe RL tasks? To answer these questions, we employ the following experiment setup to assess GradS and the baseline approaches.

5.1. Experiment setup

Tasks. We utilize several continuous control tasks for robot locomotion commonly employed in previous studies (Achiam et al., 2017; Chow et al., 2019; Zhang et al., 2020). The simulation environments are sourced from public benchmark `Bullet-Safety-Gym` (Gronauer, 2022) and `Safety-Gymnasium` (Ji et al., 2023). We consider two tasks (`Circle` and `Goal`) as shown in Figure 4 and train with various robots (`Point, Ball, Car`, and `Drone`). In the `Circle` environment, agents are rewarded for following a circular path. In the `Goal` task, agents are rewarded for reaching the goal cube. The details of the tasks can be found in the appendix. We name the task as "Robot"-"task", for example, `BC` means "Ball-Circle", and `CG` means "Car-Goal".

Constraints. In the aforementioned tasks, the original environment only provides single-dimensional cost information. To better simulate real-world scenarios, we introduce three representative costs: **Boundary/collision cost:** agents incur a cost if they cross the boundary or collide with the obstacles. **High-velocity cost:** agents receive a cost if they exceed the upper-velocity limit. **Low-velocity cost:** agents receive a cost if their speed falls below the lower-velocity limit. All costs are binary. A detailed explanation of the costs is provided in the appendix. Intuitively, boundary/collision cost and high-velocity cost are likely *redundant* constraints since high speed might also result in crossing the boundary or collision. High-velocity cost and low-velocity cost are likely *conflicting* constraints as they potentially tend to pull the policy in conflicting optimization directions if both are activated. We create tasks considering the first two types of constraints with the suffix "`-v2`", and tasks with all three types of constraints with the suffix "`-v3`" (more challenging).



(a) Circle task (b) Goal task

Figure 4: Task visualization.

Metrics. We compare the methods in terms of episodic reward (the higher, the better) and episodic constraint cost violations (the lower, the better), which have been used in many related works (Liu et al., 2023b; Li et al., 2023). We normalized the cost, and reported the most-violated cost among all the constraints (then the cost threshold becomes 1):

$$\text{cost-N} = \max_i \{c_i/\epsilon_i\} \tag{14}$$

Algorithms and baselines. For the safe RL algorithms, we select commonly used model-free off-policy algorithms, `SAC-Lag` and `DDPG-Lag`, and model-free on-policy methods, `PPO-Lag` and `TRPO-Lag`. As introduced in Section 3.2, the baseline methods are `Vanilla`, `CRPO`, and `Min-Max`. For the CRPO method, we modify it to a Lagrangian version for a fair comparison. More details and results including practical implementation, and training curves are provided in the appendix.

Env	Method	GradS (ours)		Vanilla		CRPO		Min-Max	
		Reward ↑	Cost-N ↓	Reward ↑	Cost-N ↓	Reward ↑	Cost-N ↓	Reward ↑	Cost-N ↓
BC-v2	PPO-L	271.63 ± 24.08	1.03 ± 0.12	260.2 ± 19.44	0.88 ± 0.17	288.04 ± 15.79	1.85 ± 0.11	245.41 ± 21.18	1.16 ± 0.12
	TRPO-L	329.85 ± 10.39	1.20 ± 0.06	283.88 ± 35.16	1.01 ± 0.04	362.35 ± 5.87	4.51 ± 0.11	331.41 ± 10.69	1.24 ± 0.11
	SAC-L	229.00 ± 33.64	1.09 ± 0.24	228.83 ± 16.61	1.18 ± 0.29	271.92 ± 28.16	1.49 ± 0.26	223.45 ± 7.20	1.13 ± 0.24
	DDPG-L	170.10 ± 50.92	1.06 ± 0.18	177.46 ± 48.51	1.03 ± 0.15	195.66 ± 32.35	0.97 ± 0.33	202.16 ± 29.17	1.11 ± 0.19
	Average	**250.15**	**1.10**	**237.59**	**1.03**	279.49	2.38	200.61	1.16
CC-v2	PPO-L	220.71 ± 9.63	1.04 ± 0.15	137.02 ± 15.09	0.61 ± 0.26	233.20 ± 10.11	1.85 ± 0.16	165.67 ± 29.97	0.81 ± 0.24
	TRPO-L	242.69 ± 8.69	1.01 ± 0.06	218.47 ± 13.85	1.03 ± 0.11	266.01 ± 7.91	2.14 ± 0.16	239.51 ± 9.72	1.01 ± 0.05
	SAC-L	175.69 ± 104.28	1.40 ± 0.86	34.91 ± 62.89	5.57 ± 12.24	146.39 ± 57.96	1.12 ± 0.67	46.39 ± 57.96	1.08 ± 1.35
	DDPG-L	227.89 ± 1.32	1.00 ± 0.32	176.62 ± 17.85	1.00 ± 0.14	237.92 ± 8.57	1.93 ± 0.09	175.75 ± 18.65	1.13 ± 0.69
	Average	**216.75**	**1.11**	147.76	2.05	218.38	1.76	**156.83**	**1.01**
DC-v2	PPO-L	253.21 ± 65.49	0.88 ± 0.15	137.87 ± 36.57	0.80 ± 0.16	186.30 ± 59.54	0.98 ± 0.26	164.19 ± 44.54	0.98 ± 0.13
	TRPO-L	404.16 ± 41.15	0.93 ± 0.14	306.14 ± 67.78	0.89 ± 0.09	414.74 ± 69.34	1.72 ± 0.83	359.51 ± 58.89	0.84 ± 0.09
	SAC-L	413.30 ± 76.61	0.96 ± 0.12	281.47 ± 76.09	0.71 ± 0.45	544.76 ± 68.24	3.01 ± 0.28	211.14 ± 58.74	0.70 ± 0.22
	DDPG-L	399.05 ± 44.12	0.92 ± 0.12	195.77 ± 44.53	0.94 ± 0.14	555.65 ± 52.31	3.12 ± 0.23	234.45 ± 16.91	0.84 ± 0.14
	Average	**367.43**	**0.92**	**230.3**	**0.84**	425.36	2.21	**242.32**	**0.84**
BC-v3	PPO-L	214.00 ± 57.16	0.98 ± 0.12	40.52 ± 25.33	1.02 ± 0.54	339.23 ± 72.88	1.85 ± 0.49	28.89 ± 33.33	1.84 ± 1.28
	TRPO-L	309.96 ± 25.77	0.93 ± 0.62	262.01 ± 14.24	1.09 ± 0.13	653.86 ± 58.67	3.66 ± 0.17	14.36 ± 10.01	1.32 ± 1.73
	SAC-L	253.25 ± 1.76	0.14 ± 0.12	0.06 ± 2.88	3.57 ± 0.06	855.29 ± 0.85	3.12 ± 0.04	-10.89 ± 44.52	3.14 ± 1.96
	DDPG-L	395.23 ± 71.12	1.04 ± 0.60	354.78 ± 10.97	1.04 ± 0.15	936.82 ± 83.08	3.06 ± 0.04	363.11 ± 14.93	0.93 ± 0.13
	Average	**293.11**	**0.77**	164.35	1.68	503.8	2.92	98.87	1.81
CC-v3	PPO-L	199.42 ± 28.33	0.62 ± 0.49	17.85 ± 46.46	3.13 ± 3.33	211.08 ± 16.37	1.85 ± 0.49	-6.86 ± 14.94	5.99 ± 1.45
	TRPO-L	175.53 ± 36.64	0.65 ± 0.89	33.83 ± 85.95	1.18 ± 0.66	220.11 ± 50.05	1.99 ± 1.93	36.77 ± 98.88	1.01 ± 0.53
	SAC-L	199.66 ± 56.06	1.18 ± 1.22	-66.29 ± 36.21	5.23 ± 0.98	207.12 ± 19.21	1.12 ± 0.73	-12.37 ± 21.65	2.69 ± 1.09
	DDPG-L	214.97 ± 4.05	0.44 ± 0.06	103.78 ± 60.52	2.20 ± 1.65	213.63 ± 8.37	0.88 ± 0.44	1.07 ± 36.21	1.24 ± 0.49
	Average	**197.38**	**0.72**	22.29	2.94	212.99	1.44	4.65	2.73
DC-v3	PPO-L	416.34 ± 59.33	0.97 ± 0.11	257.29 ± 21.35	1.54 ± 0.14	426.94 ± 61.96	1.92± 0.27	215.96 ± 125.53	1.52 ± 0.57
	TRPO-L	554.44 ± 55.20	1.05 ± 0.12	535.91 ± 57.58	2.01 ± 0.10	539.57 ± 21.83	2.30 ± 0.82	525.27 ± 56.78	2.11 ± 0.14
	SAC-L	590.60 ± 224.05	1.00 ± 0.12	114.94 ± 80.11	1.48 ± 0.79	728.02 ± 217.83	3.57 ± 1.42	194.09 ± 117.81	1.49 ± 1.16
	DDPG-L	643.25 ± 77.19	1.01 ± 0.04	338.91 ± 30.98	1.98 ± 0.48	839.36 ± 52.66	4.43 ± 1.42	322.05 ± 86.86	1.68 ± 0.46
	Average	**551.15**	**1.01**	311.76	1.75	633.47	3.06	**314.34**	**1.70**

Table 1: Evaluation results of the Bullet-safety-gym tasks. The cost threshold is 1. ↑ / ↓: the higher/lower, the better. Each value is averaged over 20 episodes and 5 seeds. Shade : the two most rewarding agents, **bold**: all the safe agents (cost-N $\leq$ 1) or two safest agents if none is absolutely constraint-satisfactory.

Env	Method	GradS (ours)		Vanilla		CRPO		Min-Max	
		Reward ↑	Cost-N ↓	Reward ↑	Cost-N ↓	Reward ↑	Cost-N ↓	Reward ↑	Cost-N ↓
PG-v2	PPO-L	16.74 ± 2.05	0.84 ± 0.46	1.54 ± 1.16	1.03 ± 0.29	18.27 ± 6.13	1.75 ± 0.70	6.76 ± 6.39	1.39 ± 0.59
CG-v2	PPO-L	30.57 ± 1.77	1.11 ± 0.34	0.18 ± 0.41	1.12 ± 0.64	31.35 ± 1.32	1.03 ± 0.12	2.65 ± 4.67	1.20 ± 0.76
Average		**23.66**	**0.98**	1.72	1.08	24.81	1.39	4.71	1.30
PG-v3	PPO-L	18.09 ± 2.74	1.04 ± 0.81	7.26 ± 7.87	0.85 ± 0.32	19.11 ± 2.46	1.68 ± 0.42	1.35 ± 3.98	1.19 ± 1.54
CG-v3	PPO-L	2.22 ± 5.20	0.98 ± 0.16	-2.22 ± 5.20	1.28 ± 0.45	9.75 ± 5.39	1.93 ± 0.33	-1.15 ± 1.89	1.63 ± 1.01
Average		**10.16**	**1.01**	**2.52**	**1.07**	14.43	1.81	0.10	1.16

Table 2: Evaluation results of the Safety-gymnasium tasks. The cost threshold is 1. ↑ / ↓: the higher/lower, the better. Each value is averaged over 20 episodes and 5 seeds. Shade : the two most rewarding agents, **bold**: all the safe agents (cost-N ≤ 1) or two safest agents if none is absolutely constraint-satisfactory. We selected PPO-Lag for the base safe RL algorithm since the original single-cost envs are already challenging for others such as SAC-Lag as reported by Liu et al. (2023a).

5.2. Challenges for MC Safe RL

The performance of the baseline method `Vanilla` is summarized in Table. 1, 2 and Figure. 5. It is evident that in "`-v2`" settings, `Vanilla` struggles to learn a rewarding policy due to the over-conservativeness caused by *redundant* constraints. In "`-v3`" settings, `Vanilla` encounters difficulties in exploration induced by the *conflicting* constraints, ultimately leading to the failure to learn a reasonable policy. This observation highlights the challenges posed by MC settings for safe RL, as (1) *redundant* constraints contribute to over-conservativeness in the policy update, since the agent would overestimate the effort to ensure safety, and (2) *conflicting* constraints restrict the exploration capabilities of online safe RL algorithms, causing the policy to converge to a local optimum around the initial points, resulting in the agent getting stuck due to the dominating gradients of conflicting constraints if it deviates from this point. The unsatisfactory reward and safety performance of `Vanilla` methods in MC safe RL settings underscore the importance of exploring efficient MC safe RL algorithms.

5.3. GradS performance comparison in MC Safe RL

The performance of GradS and other baseline methods `CRPO` and `Min-Max` is also summarized in Table. 1, 2, and Figure. 5. In "`-v2`" settings with *redundant* constraints, `Min-Max` exhibits strong performance since it consistently penalizes the most violated constraint, thus eliminating the negative impact of *redundant* constraints, and avoids over-conservativeness in the policy update. However, in "`-v3`" settings with *conflicting* constraints, it struggles to achieve a rewarding policy as it becomes trapped in local optima due to the lack of random exploration. Conversely, the `CRPO` method explores well with a high reward in conflicting settings, benefiting from its stochastic constraint gradient selection and resulted superior exploration capabilities, thereby avoiding entrapment in local optima. Nevertheless, it fails to ensure constraint satisfaction in the presence of *redundant* constraints due to potential imbalances between different types of cost. Specifically, if one type of *redundant* constraints significantly outweighs others in terms of quantity, the `CRPO` method would disproportionately activate constraints from this type, potentially overlooking other constraints.

The proposed GradS method demonstrates strong performance, exhibiting high rewards and small cost violations across both "`-v2`" and "`-v3`" MC-safe RL tasks from both benchmarks. In "`-v2`" tasks involving *redundant* constraints, GradS overcomes issues of over-conservativeness akin to those observed in the `Vanilla` method by eliminating *redundant* gradients. Furthermore, the

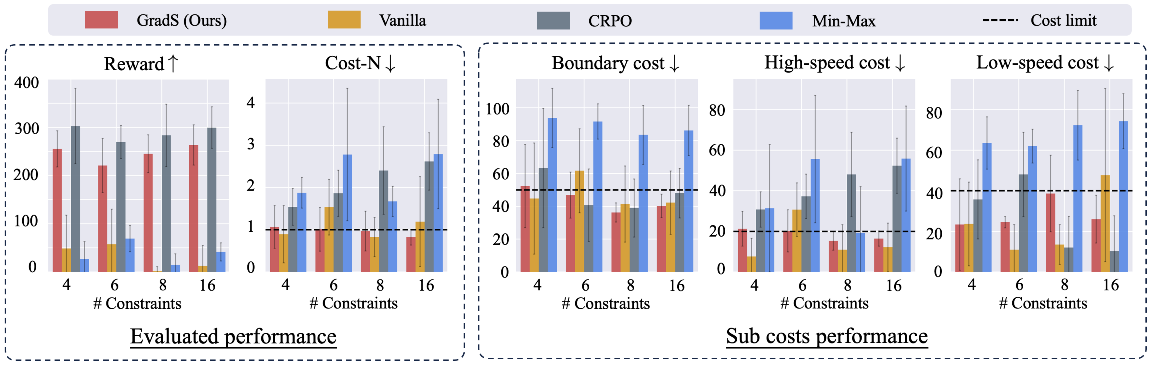


Figure 5: Scalability analysis: The x-axis in each figure means the constraint number in the tasks. The first two figures show the reward and normalized costs, while the remaining three show the representative cost returns. The bar charts represent the mean value and the error bars represent the standard deviation. All plots are averaged among 5 random seeds and 10 trajectories for each seed. ↑ / ↓: the higher/lower, the better.

elimination of redundant constraints reduces the risk of neglecting minor constraints, a drawback of the `CRPO`. In "`-v3`" scenarios with *conflicting* constraints, GradS excels in performance with high reward and low cost violation compared to the baseline algorithm as it eliminates the conflicting constraints and enables stochastic constraint gradients to encourage exploration.

5.4. Scalability analysis

The results of the cost dimension scalability experiment are shown in Figure. 5. We utilize the Ball-Circle (BC-v3) task to evaluate the algorithms across various constraint quantities. Here we increase the number of costs by creating new constraints with similar velocity thresholds and boundary positions (see appendix for details). It is evident that the baseline methods `vanilla` and `Min-max` struggle to learn safe policies, as they encounter difficulties in effectively exploring the action and observation space in the MC tasks. The baseline method `CRPO` succeeds in learning a rewarding yet unsafe policy, attributed to its lack of ability to manage imbalanced constraints. In contrast, the proposed GradS method demonstrates consistent performance as the number of constraints varies, highlighting the scalability of our approach.

6. Conclusion

In this paper, we proposed a unified framework for Lagrangian-based MC Safe RL algorithms from the standpoint of Multi-Objective Optimization (MOO), and analyze the MC safe RL problem through the lens of constraint types, identifying two challenging MC safe RL settings: *redundant* and *conflicting* constraints. To address these challenges, we propose the constraint gradient shaping (GradS) technique, ensuring compatibility with general Lagrangian-based safe RL algorithms. Our analysis highlights the necessity of developing efficient and effective algorithms for handling multiple costs, shedding light on the critical importance of addressing multi-cost constraints in safe RL settings. The extensive experimental results reconfirm that GradS effectively solves the MC safe RL problems in both *redundant* and *conflicting* constraint settings, and is safer, and more rewarding than baseline methods. By proposing the GradS technique and providing a comprehensive analysis, we hope to contribute to the advancement of safe RL algorithms and their successful implementation in real-world complex and safety-critical environments. The limitation of this work is the additional computational burden when calculating the gradient similarity. The future work contains the extension to offline safe RL settings.

Acknowledgment

The work is partially supported by Google Deepmind with an unrestricted grant. The authors also want to acknowledge the support from the National Science Foundation under grants CNS-2047454.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, pages 22–31. PMLR, 2017.

Eitan Altman. Constrained markov decision processes with total cost criteria: Lagrangian approach and dual linear program. *Mathematical methods of operations research*, 48(3):387–417, 1998.

Yarden As, Ilnura Usmanova, Sebastian Curi, and Andreas Krause. Constrained policy optimization via bayesian world models. *arXiv preprint arXiv:2201.09802*, 2022.

Homanga Bharadhwaj, Aviral Kumar, Nicholas Rhinehart, Sergey Levine, Florian Shkurti, and Animesh Garg. Conservative safety critics for exploration. *arXiv preprint arXiv:2010.14497*, 2020.

Shalabh Bhatnagar and K Lakshmanan. An online actor–critic algorithm with function approximation for constrained markov decision processes. *Journal of Optimization Theory and Applications*, 153 (3):688–708, 2012.

Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5, 2021.

Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022.

Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

Yikun Cheng, Pan Zhao, and Naira Hovakimyan. Safe and efficient reinforcement learning using disturbance-observer-based control barrier functions. In *Learning for Dynamics and Control Conference*, pages 104–115. PMLR, 2023.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1):6070–6120, 2017.

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

Hongyuan Ding, Yan Xu, Benjamin Chew Si Hao, Qiaoqiao Li, and Antonis Lentzakis. A safe reinforcement learning approach for multi-energy management of smart home. *Electric Power Systems Research*, 210:108120, 2022.

Yuhao Ding and Javad Lavaei. Provably efficient primal-dual reinforcement learning for cmdps with non-stationary objectives and constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7396–7404, 2023.

Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*, 2018.

Shuo Feng, Haowei Sun, Xintao Yan, Haojie Zhu, Zhengxia Zou, Shengyin Shen, and Henry X Liu. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615(7953): 620–627, 2023.

Heshan Devaka Fernando, Han Shen, Miao Liu, Subhajit Chaudhury, Keerthiram Murugesan, and Tianyi Chen. Mitigating gradient bias in multi-objective learning: A provably convergent approach. In *The Eleventh International Conference on Learning Representations*, 2022.

Yannis Flet-Berliac and Debabrota Basu. Saac: Safe reinforcement learning as an adversarial game of actor-critics. *arXiv preprint arXiv:2204.09424*, 2022.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Sven Gronauer. Bullet-safety-gym: Aframework for constrained reinforcement learning. 2022.

Shangding Gu, Jakub Grudzien Kuba, Munning Wen, Ruiqing Chen, Ziyan Wang, Zheng Tian, Jun Wang, Alois Knoll, and Yaodong Yang. Multi-agent constrained policy optimisation. *arXiv preprint arXiv:2110.02793*, 2021.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, Yaodong Yang, and Alois Knoll. A review of safe reinforcement learning: Methods, theory and applications. *arXiv preprint arXiv:2205.10330*, 2022.

Kai-Chieh Hsu, Duy Phuong Nguyen, and Jaime Fernàndez Fisac. Isaacs: Iterative soft adversarial actor-critic for safety. In *Learning for Dynamics and Control Conference*, pages 90–103. PMLR, 2023a.

Kai-Chieh Hsu, Allen Z Ren, Duy P Nguyen, Anirudha Majumdar, and Jaime F Fisac. Sim-to-lab-to-real: Safe reinforcement learning with shielding and generalization guarantees. *Artificial Intelligence*, 314:103811, 2023b.

Haimin Hu, Zixu Zhang, Kensuke Nakamura, Andrea Bajcsy, and Jaime F Fisac. Learning-aware safety for interactive autonomy. *arXiv preprint arXiv:2309.01267*, 2023.

Sandy Huang, Abbas Abdolmaleki, Giulia Vezzani, Philemon Brakel, Daniel J Mankowitz, Michael Neunert, Steven Bohez, Yuval Tassa, Nicolas Heess, Martin Riedmiller, et al. A constrained multi-objective reinforcement learning framework. In *Conference on Robot Learning*, pages 883–893. PMLR, 2022.

David Isele, Alireza Nakhaei, and Kikuo Fujimura. Safe reinforcement learning on autonomous vehicles. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–6. IEEE, 2018.

Jiaming Ji, Borong Zhang, Xuehai Pan, Jiayi Zhou, Juntao Dai, and Yaodong Yang. Safety-gymnasium. *GitHub repository*, 2023.

Vanshaj Khattar, Yuhao Ding, Bilgehan Sel, Javad Lavaei, and Ming Jin. A cmdp-within-online framework for meta-safe reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Dohyeong Kim and Songhwai Oh. Efficient off-policy safe reinforcement learning using trust region conditional value at risk. *IEEE Robotics and Automation Letters*, 7(3):7644–7651, 2022.

Dohyeong Kim, Kyungjae Lee, and Songhwai Oh. Trust region-based safe distributional reinforcement learning for multiple constraints. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Yunho Kim, Hyunsik Oh, Jeonghyun Lee, Jinhyeok Choi, Gwanghyeon Ji, Moonkyu Jung, Donghoon Youm, and Jemin Hwangbo. Not only rewards but also constraints: Applications on legged robot locomotion. *arXiv preprint arXiv:2308.12517*, 2023b.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Jinning Li, Xinyi Liu, Banghua Zhu, Jiantao Jiao, Masayoshi Tomizuka, Chen Tang, and Wei Zhan. Guided online distillation: Promoting safe reinforcement learning by offline demonstration. *arXiv preprint arXiv:2309.09408*, 2023.

Shengbo Eben Li. Deep reinforcement learning. In *Reinforcement Learning for Sequential Decision and Optimal Control*, pages 365–402. Springer, 2023.

Qingkai Liang, Fanyu Que, and Eytan Modiano. Accelerated primal-dual policy optimization for safe reinforcement learning. *arXiv preprint arXiv:1802.06480*, 2018.

Haohong Lin, Wenhao Ding, Zuxin Liu, Yaru Niu, Jiacheng Zhu, Yuming Niu, and Ding Zhao. Safety-aware causal representation for trustworthy offline reinforcement learning in autonomous driving. *IEEE Robotics and Automation Letters*, 2024.

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

Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, et al. Datasets and benchmarks for offline safe reinforcement learning. *arXiv preprint arXiv:2306.09303*, 2023a.

Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constrained decision transformer for offline safe reinforcement learning. *arXiv preprint arXiv:2302.07351*, 2023b.

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

Zengyi Qin, Kaiqing Zhang, Yuxiao Chen, Jingkai Chen, and Chuchu Fan. Learning safe multi-agent control with decentralized neural barrier certificates. *arXiv preprint arXiv:2101.05436*, 2021.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7, 2019.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

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

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. In *International Conference on Machine Learning*, pages 9767–9779. PMLR, 2021.

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

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

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3): 4915–4922, 2021.

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

Akifumi Wachi, Yunyue Wei, and Yanan Sui. Safe policy optimization with local generalized linear function approximations. *Advances in Neural Information Processing Systems*, 34:20759–20771, 2021.

Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning: a hierarchical bayesian approach. In *Proceedings of the 24th international conference on Machine learning*, pages 1015–1022, 2007.

Mengdi Xu, Zuxin Liu, Peide Huang, Wenhao Ding, Zhepeng Cen, Bo Li, and Ding Zhao. Trustworthy reinforcement learning against intrinsic vulnerabilities: Robustness, safety, and generalizability. *arXiv preprint arXiv:2209.08025*, 2022.

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

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

Tsung-Yen Yang, Tingnan Zhang, Linda Luu, Sehoon Ha, Jie Tan, and Wenhao Yu. Safe reinforcement learning for legged locomotion. *arXiv preprint arXiv:2203.02638*, 2022.

Yongxin Yang and Timothy M Hospedales. Trace norm regularised deep multi-task learning. *arXiv preprint arXiv:1606.04038*, 2016.

Yihang Yao, Zuxin Liu, Zhepeng Cen, Jiacheng Zhu, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constraint-conditioned policy optimization for versatile safe reinforcement learning. *arXiv preprint arXiv:2310.03718*, 2023.

Ming Yu, Zhuoran Yang, Mladen Kolar, and Zhaoran Wang. Convergent policy optimization for safe reinforcement learning. *arXiv preprint arXiv:1910.12156*, 2019.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33: 5824–5836, 2020.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018.

Hongchao Zhang, Junlin Wu, Yevgeniy Vorobeychik, and Andrew Clark. Exact verification of relu neural control barrier functions. *arXiv preprint arXiv:2310.09360*, 2023.

Richard Zhang and Daniel Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *International Conference on Machine Learning*, pages 11096–11105. PMLR, 2020.

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

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

Weiye Zhao, Tairan He, and Changliu Liu. Probabilistic safeguard for reinforcement learning using safety index guided gaussian process models. In *Learning for Dynamics and Control Conference*, pages 783–796. PMLR, 2023.

Liyuan Zheng, Yuanyuan Shi, Lillian J Ratliff, and Baosen Zhang. Safe reinforcement learning of control-affine systems with vertex networks. In *Learning for Dynamics and Control*, pages 336–347. PMLR, 2021.