Evaluating Model-Free Reinforcement Learning toward Safety-Critical Tasks

Linrui Zhang[1]*, Qin Zhang[1], Li Shen[2†], Bo Yuan[3], Xueqian Wang[1†], Dacheng Tao[2]

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University
[2] JD Explore Academy, JD.com, Inc.
[3] Qianyuan Institute of Sciences
{zlr20,zhangqin21}@mails.tsinghua.edu.cn, {mathshenli,dacheng.tao}@gmail.com,
boyuan@ieee.org, wang.xq@sz.tsinghua.edu.cn

Abstract

Safety comes first in many real-world applications involving autonomous agents. Despite a large number of reinforcement learning (RL) methods focusing on safety-critical tasks, there is still a lack of high-quality evaluation of those algorithms that adheres to safety constraints at each decision step under complex and unknown dynamics. In this paper, we revisit prior work in this scope from the perspective of state-wise safe RL and categorize them as projection-based, recovery-based, and optimization-based approaches, respectively. Furthermore, we propose Unrolling Safety Layer (USL), a joint method that combines safety optimization and safety projection. This novel technique explicitly enforces hard constraints via the deep unrolling architecture and enjoys structural advantages in navigating the trade-off between reward improvement and constraint satisfaction. To facilitate further research in this area, we reproduce related algorithms in a unified pipeline and incorporate them into SafeRL-Kit, a toolkit that provides off-the-shelf interfaces and evaluation utilities for safety-critical tasks. We then perform a comparative study of the involved algorithms on six benchmarks ranging from robotic control to autonomous driving. The empirical results provide an insight into their applicability and robustness in learning zero-cost-return policies without task-dependent handcrafting. The project page is available at https://sites.google.com/view/saferlkit.

Introduction

Model-free reinforcement learning (RL) has achieved superhuman performance on many decision-making problems (Mnih et al. 2015; Vinyals et al. 2019). Typically, the agent learns from trial and error and requires minimal prior knowledge of the environment. Such a paradigm features significant advantages in mastering essential skills for complex systems, but concerns about the systematic safety limit the extent of adoption of model-free RL in real-world applications, such as human-robot collaboration (Villani et al. 2018) and autonomous driving (Kiran et al. 2021).

Penalizing unsafe transitions via the reward function is straightforward but sometimes cumbersome to navigate the trade-off between performance and safety. A trivial penalty term may fail to obtain a risk-averse policy, whereas an excessive punishment may make the agent too conservative to explore the environment. Alternatively, incorporating safety into RL via constraints (Altman 1999) is widely adopted since the strength of constraints reflects the human-specified safety requirement, and the agent is desired to optimize its behavior within the constrained policy search space.

In this paper, we explore model-free reinforcement learning methods that adhere to state-wise safety constraints. To better understand how this study is developed, two points deserve further clarification. First, our work is aimed at learning a stationary safe policy under the general model-free settings, instead of refusing any safety violations during the training. The latter is more related to optimal control and relies on the known dynamic model (Cheng et al. 2019) or a carefully designed energy function (Zhao, He, and Liu 2021). Second, we focus on the state-wise constraint at every decision-making step and demonstrate that this type of constraint is more strict and practical toward safety-critical tasks theoretically and empirically. Our contributions in this paper are summarized as follows:

1. We revisit model-free RL following state-wise safety constraints and present SafeRL-Kit, a toolkit that implements prior work in this scope under a unified off-policy framework. Specifically, SafeRL-Kit contains projection-based Safety Layer (Dalal et al. 2018), recovery-based Recovery RL (Thananjeyan et al. 2021), optimization-based Off-policy Lagrangian (Ha et al. 2020), Feasible Actor-Critic (Ma et al. 2021), and the new method proposed in this paper.
2. We propose Unrolling Safety Layer (USL), a novel approach that combines safety projection and safety optimization. USL unrolls gradient-based corrections to the jointly optimized actor-network and thus explicitly enforces the constraints. The proposed method is simple-yet-effective and outperforms state-of-the-art algorithms in learning risk-averse policies.
3. We perform a comparative study based on SafeRL-Kit and evaluate the related algorithms on six different tasks. We further demonstrate their applicability and robustness in safety-critical tasks with the universal binary cost indicator and a constant constraint threshold.

*This work was done during Linrui Zhang's internship at JD Explore Academy, JD.com, Inc.
†Correspondence to: Xueqian Wang and Li Shen

Related Work

Safe RL Algorithms. Safe RL optimizes policies under episodic or instantaneous constraints. The most common approach to solving the episodic constraint is Lagrangian relaxation (Chow et al. 2017; Tessler at al. 2017; Stooke et al. 2020). Other works (Achiam et al. 2017; Yang et al. 2020) approximate the constrained policy iteration with a quadratic constrained optimization. Recently, first-order methods (Liu, Ding, and Liu 2020; Zhang et al. 2022; Yang et al. 2022a) start to gain attractions as the objective is efficient to optimize and easy to handle multiple constraints. For the instantaneous constraint, Lagrangian relaxation is also a feasible solution (Bohez et al. 2019). Dalal et al. (2018) perform quadratic programming to project actions back to the safe set. Other works model the instantaneous cost as Gaussian Processes and plan in the safety-proven neighboring states (Wachi et al. 2018; Wachi and Sui 2020). In this paper, we formulate safe RL following state-wise safety constraints, which are slightly different from the above genres.

Safe RL Benchmarks. There have already been some safety-critical benchmarks to evaluate the efficacy of safe RL methods, including traditional MuJoCo tasks (Achiam et al. 2017; Zhang, Vuong, and Ross 2020), navigation in the cluttered environment (Ray, Achiam, and Amodei 2019), safe robotic control task (Yuan et al. 2021) and safe autonomous driving (Li et al. 2021; Herman et al. 2021). However, a comprehensive study on learning a zero-cost-return policy with model-free methods is still absent.

Preliminaries

This study lies in the context of constrained Markov Decision Process (CMDP) (Altman 1999), which extends standard MDP (Sutton and Barto 1998) as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{C}, \mu, \gamma)$. $\mathcal{S}$ and $\mathcal{A}$ denote the state space and the action space, respectively. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ is the transition probability function to describe the dynamics of the system. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function. $\mathcal{C} : \mathcal{S} \times \mathcal{A} \mapsto [0, +\infty]$ is the cost function and reflects the safety violation. $\mu : \mathcal{S} \mapsto [0, 1]$ is the initial state distribution, and γ is the discount factor for future reward and cost. A stationary policy $\pi : S \mapsto P(A)$ maps the given states to probability distributions over action space and the expected discounted return of the policy is $J_R(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$, where $\tau \sim \pi$ accounts for the stochastic trajectory distribution sampled on $s_0 \sim \mu, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)$. The goal of safe RL is to find the optimal policy:

$$\max_\pi J_R(\pi) \quad \text{s.t. } \pi \text{ is feasible.} \tag{1}$$

In a CMDP, the agent is typically constrained by the cost function in two ways. One is the *Episodic Constraint*. This type of formulation requires the cost-return in the whole trajectory be within a certain threshold, namely $J_C(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t c_t\right] \leq d$, which is suitable for total energy consumption, resource over-utilization, etc. The other is the *Instantaneous Constraint*. This type of formulation requires the selected actions to enforce the constraint at every decision-making step, namely $\forall t, \mathbb{E}_\pi[c_t|s_t] \leq \epsilon$, which is indispensable in accident and damage avoidance.

Revisit RL toward Safety-critical Tasks

In many safety-critical scenarios, the final policy is assumed to maintain the zero-cost return since any inadmissible behavior could lead to catastrophic failure in the execution. Prior constrained learning paradigms have fatal flaws under this premise. For the episode constraint, with a threshold d close to 0, the agent often either fails to improve policy or receives a cost-return more significant than 0. The underlying reason is that such a constraint is easy to violate, especially when the time horizon contains thousands of steps. If the algorithm handles the episodic formula directly, it is cumbersome to identify ill actions as well as tweak the policy parameters and the corresponding action sequence. For another, the instantaneous constraint is tighter since it is a sufficient but not necessary condition for the episodic constraint with $\epsilon = (1 - \gamma)d$. Nevertheless, it is problematic to directly enforce $C(s_t, a_t) \leq \epsilon$ at each single decision-making step in complicated dynamics, since some actions have a long-term effect on future visited states and the infeasible states might also be intractable to recover in a single time-step. A more reasonable solution is to prevent the current state from falling into the unsafe set in a certain planning span, which is inspired by model predictive control. Consequently, we define the long-term return for cost as $Q^\pi_c(s_t, a_t) = \mathbb{E}_\pi\left[\sum_{t'=t}^{\infty} \gamma^t c_{t'}|s_t, a_t\right]$, which is similar to $Q^\pi(s, a)$ for reward in standard RL by substituting r_t with c_t. Next, we present the formal definitions related to state-wise safe reinforcement learning.

Definition 1 (State-wise safety constraints)**.** *In the whole trajectory, the agent is required to adhere to the following long-term constraint at every visited state*

$$Q^\pi_c(s_t, a_t) \leq \delta, \forall t \geq 0. \tag{2}$$

Definition 2 (Optimal state-wise safe policy)**.** *In any feasible state, the optimal action is*

$$a^* = \arg\max_a \left[Q^\pi(s, a)\right] \quad \text{s.t. } Q^\pi_c(s, a) \leq \delta. \tag{3}$$

Using the cumulative constraint to enhance state-wise safety is not novel (Srinivasan et al. 2020; Ma et al. 2021; Yu et al. 2022). Nevertheless, the choice of δ is tricky since the relationship between (2) and the desired instantaneous constraint is not straightforward. In this paper, we give a theoretical bound of cost limit δ as follows.

Proposition 1. *If $\delta \leq \epsilon \cdot \gamma^T$, any policy $\pi(\cdot|s)$ satisfying* (2) *fulfills $\mathbb{E}_\pi[c_t|s_t] \leq \epsilon$ within the planning span T.*

The above proposition, to some extent, alleviates the concern that decreasing δ leads to overly conservative policy. For example, if a racing car is able to slow down and avoid the obstacle in 20 steps ahead, setting $T > 20$ will not change the optimal sequence for $a^*_{t-t'}$, where $t' > 20$. In our experiments, we keep $\delta = 0.1$ across different safety-critical tasks, which equals the safe planning span of at least 100 steps with a universal binary cost indicator. Empirical results demonstrate that safe RL methods adhering to state-wise safety constraints are robust to this value.

In this paper, we explore model-free RL that adheres to state-wise safety constraints in continuous state-action spaces and unknown dynamics. We classify the most related approaches into the following three categories:

Safety Correction. This type of methods corrects tl tial unsafe decision by projecting it back to the safe se projection can be constructed by the control barrier tion (Cheng et al. 2019) with known dynamics, ir safety index (Zhao, He, and Liu 2021) with hand-c energy function, or parametric linear model (Dalal 2018) learned from past experiences. However, tho proaches are sometimes under-performed regarding lative rewards since the correction only guarantees tl sibility but lacks the equivalence with optimality.

Safety Recovery. This type of methods is especiall comed in the field of robotics (Thananjeyan et al. 2021 et al. 2022b) and autonomous driving (Chen et al. 2021 critical idea behind Recovery RL is introducing a dedicated policy that recovers unsafe states, whereas the task policy is trained by the standard RL to achieve the original goal. However, those approaches struggle for a rational recovery policy since it tends to be overly conservative in preventing risky exploration. Furthermore, the decisions between two policies may conflict with each other, which makes the agents stuck near the boundaries of safe regions easily.

Safety Optimization. This type of methods incorporates safety constraints into the RL objective and yields a constrained sequential optimization task. These approaches employ different optimization objectives to guide the updates of parametric policies, which can be tackled by Lagrangian relaxation (Ha et al. 2020; Ma et al. 2021) or the penalty method (Zhang et al. 2022). Unfortunately, the "soft" loss function in the sample-based learning does not consistently enforce the "hard" constraint in practice and barely leads to zero-cost-return policies even at convergence.

Unrolling Safety Layer: A Novel Approach

Orthogonal to existing algorithms, we propose a novel approach referred to as **U**nrolling **S**afety **L**ayer (USL) in this paper, which is inspired by the complementary of safety projection and safety optimization. For projection-based approaches, the correction (even if tractable) only enforces the feasibility but lacks the equivalence to the optimal maximum return. For optimization-based approaches, most of them tend to find the optimal solution to the constrained problem with the help of neural networks. However, the forward computing lacks explicit restrictions on the output actions, and thus the "soft" loss function often fails to fully satisfy "hard" constraints. Recently, Deep Constraint Completion and Correction (DC3) (Donti, Rolnick, and Kolter 2021) shows potential to achieve optimal objective values while preserving feasibility, which is of independent interest to general constrained problems. As illustrated in Figure 1, we employ a similar joint architecture that combines safety optimization (serves as the first stage's approximate solver) and safety projection (serves as the second stage's iterative correction). To the best of our knowledge, this is the first study to introduce deep unrolling optimization into safe RL.

Stage 1: Policy Network as Approximate Solver

We train a parametric neural network in the first stage as the approximate solver to problem (3), which aims to output sub-optimal actions via naive forward computing. Different from Donti et al. (2021) that simply applies ℓ_2 regular term to the objective function, we use the exact penalty function (Zhang et al. 2022) as the alternative. The merit is that one can construct an equivalent counterpart for problem (3) with a finite penalty factor κ as

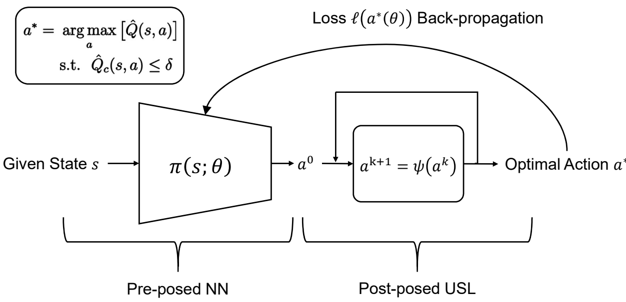

Figure 1: The deep unrolling architecture for safe RL. At each decision-making step, the pre-posed policy network outputs near-optimal action a_0; the post-posed unrolling safety layer (USL) takes a_0 as the initial solution and iteratively performs gradient-based corrections to enforce the hard constraint. Back-propagation of the state-wise constrained objective guides the policy optimization.

$$\ell(\pi) = \mathbb{E}_{\mathcal{D}}\big[-Q^{\pi}(s,\pi(s)) + \kappa \cdot \max\{0, Q_c^{\pi}(s,\pi(s)) - \delta\}\big]. \quad (4)$$

Proposition 2. *Let $\mathcal{L}(\pi,\lambda)$ denote the Lagrangian function $\mathbb{E}_{\mathcal{D}}\big[-Q^{\pi}(s,\pi(s)) + \lambda(s)\big(Q_c^{\pi}(s,\pi(s)) - \delta\big)\big]$. Assume that the optimal π^* and λ^* exist for the dual problem $\max_{\lambda\geq 0}\min_{\pi}\mathcal{L}(\pi,\lambda)$. If $\kappa \geq ||\lambda(s)||_{\infty}$, it holds that*

$$\min \ell(\pi) \iff \max_{\lambda\geq 0}\min_{\pi}\mathcal{L}(\pi,\lambda)$$

We use a fixed κ as a hyper-parameter in the practical implementation and find that a large constant ($\kappa = 5$ in our experiments) is empirically effective across different tasks even if the supremum of Lagrange multipliers is intractable to estimate. Moreover, if the actual $\lambda(s)$ tends to be positively infinitive for some critically dangerous states, there would be numerical issues for optimization-based approaches. On the contrary, the objective function (4) under those circumstances can be regarded as a penalty method by adding regularization terms and only gives a sub-optimal initial solution for the next stage. Fortunately, the proposed two-stage architecture does not require an optimal solution in the first stage, and the joint training and inference process with post-projection can tackle this problem to some extent.

Stage 2: Gradient-based Projection

The approximate solver in stage 1 may still output infeasible actions for the following reasons: (a) The supremum of Lagrangian multipliers in Proposition 2 is hard to obtain, and we only settle κ as a fixed, large but sub-optimal hyper-parameter. (b) The inherent issues of safe RL, such as the approximation in the modeling, sample-based learning, distributional shift, etc., make it possible for the end-to-end actor to violate hard constraints in the policy execution.

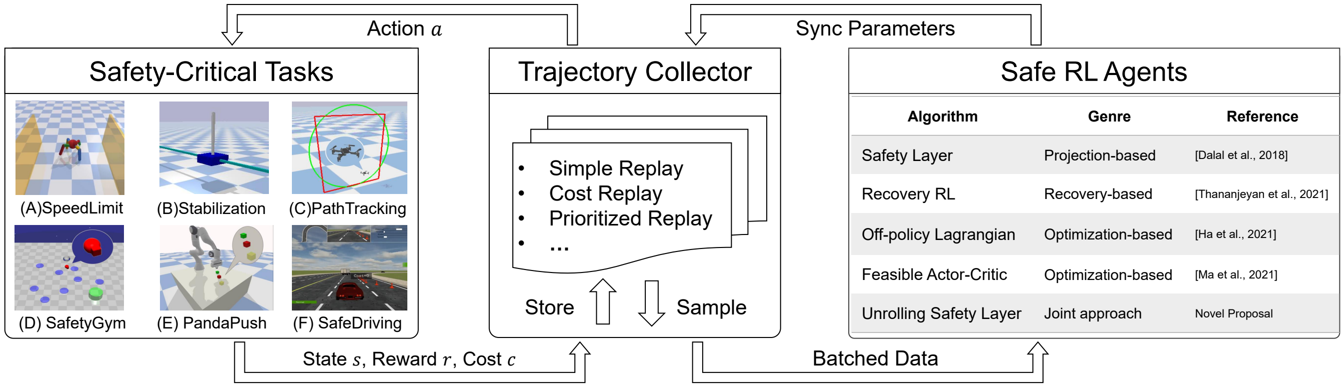

Figure 2: The schema of SafeRL-Kit. All the algorithms are implemented under off-policy settings and evaluated on six safety-critical tasks with the universal binary cost indicator and a constant constraint threshold.

To address the above issue, the post-posed projection performs gradient steps to rectify the hard constraint from the initial iteration point $a^0 \sim \pi(\cdot|s)$. $\psi(\cdot)$ is the differentiable operator that takes intermediate action a^k at k_{th} iteration as input and performs a gradient descent towards the constraint-violation wrapped with ReLU function:

$$\psi(a^k) = a^k - \frac{\eta}{\mathcal{Z}_k} \cdot \frac{\partial}{\partial a^k}\big[Q_c(s, a_k) - \delta\big]^+. \tag{5}$$

Here $\mathcal{Z}_k = ||\frac{\partial}{\partial a^k}\big[Q_c(s,a_k) - \delta\big]^+||_\infty$ is the normalization factor that rescales the gradients on a_k, and therefore the hyper-parameter η determines the maximum step size of the action change.

Notably, the iterative executions of $\psi(\cdot)$ do not always converge to global (or even local) optima for the primal constrained optimization problem (3). Nevertheless, such a method is highly effective in practice if the initial iteration point is close to the optimal solution (Panageas, Piliouras, and Wang 2019; Donti, Rolnick, and Kolter 2021). This fact emphasizes the necessity for training a pre-posed policy network via the exact penalty regularization, which provides a non-pathological initialization for USL. By means of minimizing the objective function (4), the output of $\pi(\cdot|s)$ may be still infeasible sometimes but already close to the optimal action a^*. Thus, the sequence of $a^{k+1} = \psi(a^k)$ is expected to converge to a^* when $k \to +\infty$. However, letting $k \to +\infty$ is not practical in use, and thus we set an upper limit K as the maximum iterations of USL. Note that the value of K needs to match the gradient step-size factor η. For example, we set $\eta = 0.05$ and $K = 20$ to enable USL to degrade the normalized action from 1 to 0 within maximum iterations.

SafeRL-Kit: A Systematic Implementation

To facilitate further research in this area, we release SafeRL-Kit[1], a reproducible and open-source safe RL toolkit as shown in Figure 2. In brief, SafeRL-Kit contains a list of representative algorithms that address safe learning from different perspectives. Potential users can also incorporate domain-specific knowledge into appropriate baselines to build more competent algorithms for their tasks of interest. Furthermore, SafeRL-Kit is implemented in an off-policy training pipeline, which provides unified and efficient interfaces for fair comparisons among different algorithms on different benchmarks.

Safety-critical Benchmarks

SafeRL-Kit includes six safety-critical benchmarks, ranging from basic robotic control to autonomous driving, which are well-explored in recent literature (Yuan et al. 2021; Ray, Achiam, and Amodei 2019; Li et al. 2021). A short description of the benchmarks is presented below:

(A) SpeedLimit. The four-legged ant runs along the avenue and receives a cost signal when exceeding the velocity limit.

(B) Stabilization. The cart pole is rewarded for keeping itself upright while being constrained by angular velocity.

(C) PathTracking. The quadrotor tracks the green circular trajectory and receives a cost signal if it leaves the area allowed to fly bounded within the red rectangular.

(D) SafetyGym-PG. The mass point moves to the green goal and is required to get rid of blue hazards.

(E) PandaPush. The robotic arm pushes the green cube to the destination while avoiding collisions with the red cube.

(F) SafeDriving. The autonomous vehicle learns to reach the navigation land markers as quickly as possible but is not allowed to collide with other vehicles or be out of the road.

Safe Learning Algorithms

SafeRL-Kit includes five safe learning methods addressing safety-critical tasks from different perspectives.

Safety Layer (Dalal et al. 2018) for safety projection

$$a^* = \arg\min_a \frac{1}{2}||a - \pi_\theta(s)||^2 \quad \text{s.t. } g_\omega(s)^\top a + \bar{c}(s) \le \delta.$$

Recovery RL (Thananjeyan et al. 2021) for safety recovery

$$a_t = \begin{cases} \pi_{\text{task}}(s_t), & \text{if } Q^\pi_{\text{risk}}\big(s_t, \pi_{\text{task}}(s_t)\big) \le \delta \\ \pi_{\text{risk}}(s_t), & \text{otherwise} \end{cases}.$$

[1]Project page: https://sites.google.com/view/saferlkit

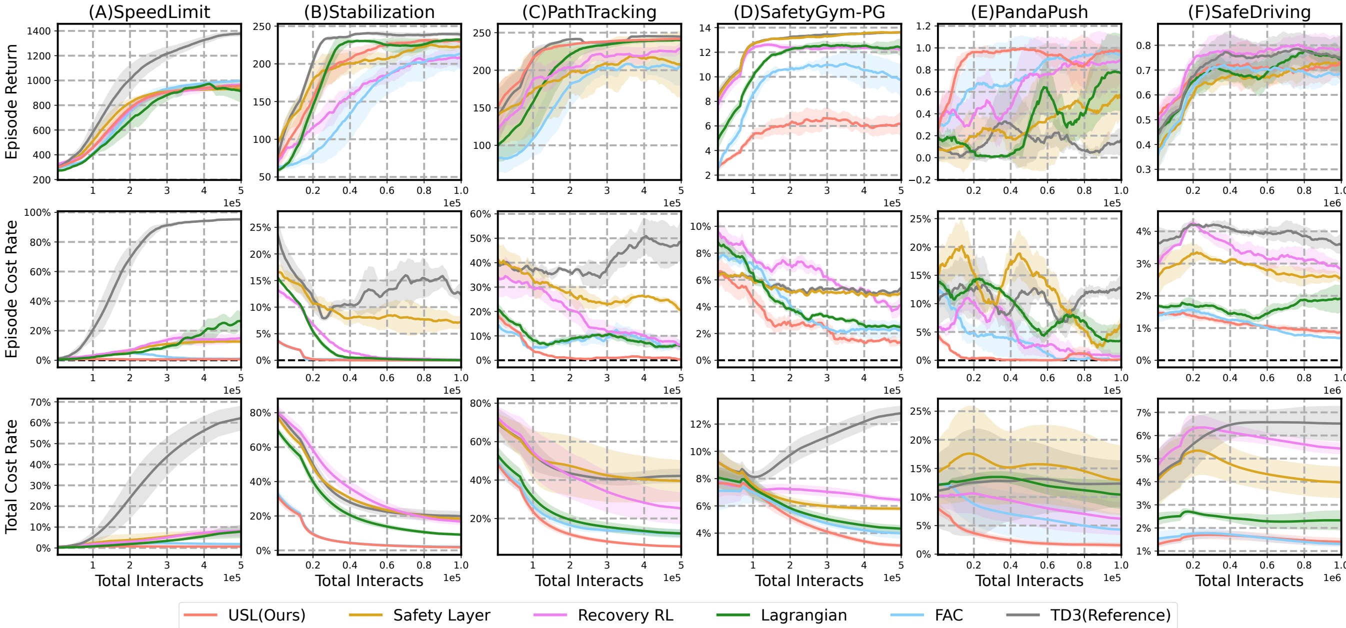

Figure 3: Learning curves of different algorithms on safety-critical tasks. The x-axis is the number of interactions. The y-axis represents episodic return (top line), episodic cost rate (middle line) and total cost rate (bottom line), respectively. The dashed line is the expected zero-cost limit in the test time.

Off-Policy Lagrangian Method (Ha et al. 2020) for safety optimization

$$\max_{\lambda \geq 0} \min_{\theta} \mathbb{E}_{\mathcal{D}} - Q^{\pi}(s, \pi_\theta(s)) + \lambda(Q_c^{\pi}(s, \pi_\theta(s)) - \epsilon).$$

Feasible Actor Critic (FAC) (Ma et al. 2021) for safety optimization

$$\max_{\xi} \min_{\theta} \mathbb{E}_{\mathcal{D}} - Q^{\pi}(s, \pi_\theta(s)) + \lambda_\xi(s)(Q_c^{\pi}(s, \pi_\theta(s)) - \epsilon).$$

Unrolling Safety Layer (USL). A joint approach proposed in this paper combining safety projection and optimization.

All the above algorithms in SafeRL-Kit are implemented under the off-policy Actor-Critic architecture. Although these model-free algorithms may inevitably encounter cost signals in the training process, they still enjoy better sample efficiency with fewer unsafe transitions compared with on-policy implementations (Ray, Achiam, and Amodei 2019), and can better leverage human demonstration if needed. The essential updates of backbone networks uniformly follow TD3 (Fujimoto, Hoof, and Meger 2018), and thus we can perform a fair evaluation to see which of them are best suited for safety-critical tasks.

Cost Function and Evaluation Metrics

Without loss of generality, we uniformly designate the cost function as a binary indicator (1 for unsafe transitions; 0 for other cases), and our experiments aim to obtain stationary policies that adhere to zero cost signals. It is worth noting that some works define cost more prospectively, for example, using the distance from the car to the road boundary in autonomous driving scenarios (Chen et al. 2021). We omit any task-dependent hand-crafting in our comparative study since they overly rely on domain-specific knowledge and are sometimes intractable on complex tasks. Instead, receiving an instantaneous cost signal is much more straightforward and can be generalized to related tasks.

Considering the properties of safety-critical tasks and the definition of the cost function, we employ the following metrics for the joint evaluation:

Episodic Return $\triangleq$ sum of rewards in the test time. It indicates how well the agent finishes the original task.

Episodic Cost Rate $\triangleq \frac{\text{number of cost signals}}{\text{length of the episode}}$. It indicates how safe the agent is in the test time.

Total Cost Rate $\triangleq \frac{\text{total number of cost signals}}{\text{total number of training steps}}$. It indicates how safe the agent is in the whole training process.

Experiments

In this section, we empirically evaluate model-free RL toward safety-critical tasks based on SafeRL-Kit and investigate the following research questions.

Q1: How applicable and robust are the algorithms regarding state-wise constraints?

We plot the learning curves of each algorithm on different tasks over five random seeds in Figure 3 and report their mean performance at convergence in Table 1.

TD3 (Fujimoto, Hoof, and Meger 2018) is used as the unconstrained reference for the upper bounds of reward performance and cost signals. On the *SpeedLimit* task, the safety

Tasks		USL(ours)	Safety Layer	Recovery RL	Lagrangian	FAC	TD3(ref)
(A)Speedlimit	Ep Return	965.97 ± 5.09	935.69 ± 9.11	965.73 ± 3.52	897.32 ± 77.80	$\mathbf{978.77 \pm 12.12}$	1382.16 ± 19.65
	Ep CostRate(%)	$\mathbf{0.63 \pm 0.55}$	12.73 ± 2.71	15.10 ± 4.07	$27,17 \pm 14.00$	1.07 ± 0.58	95.34 ± 1.03
	Tot CostRate(%)	$\mathbf{0.63 \pm 0.06}$	8.00 ± 0.75	8.25 ± 2.74	8.01 ± 3.28	1.85 ± 0.40	62.56 ± 5.89
(B)Stabilization	Ep Return	228.18 ± 3.88	222.80 ± 15.22	204.23 ± 3.92	$\mathbf{231.61 \pm 2.12}$	214.20 ± 19.38	238.47 ± 3.99
	Ep CostRate(%)	$\mathbf{0.00 \pm 0.00}$	6.98 ± 2.96	0.10 ± 0.04	0.03 ± 0.06	0.02 ± 0.03	12.89 ± 4.08
	Tot CostRate(%)	$\mathbf{1.77 \pm 0.23}$	18.27 ± 2.06	16.63 ± 1.98	9.05 ± 0.85	2.07 ± 0.19	20.01 ± 2.87
(C)Pathtracking	Ep Return	$\mathbf{241.74 \pm 0.88}$	205.93 ± 46.73	229.05 ± 7.16	240.50 ± 2.03	218.28 ± 28.99	248.80 ± 0.54
	Ep CostRate(%)	$\mathbf{0.17 \pm 0.18}$	18.79 ± 5.52	6.22 ± 3.91	5.77 ± 2.27	6.33 ± 3.55	48.40 ± 9.49
	Tot CostRate(%)	$\mathbf{5.36 \pm 0.60}$	39.44 ± 10.66	25.08 ± 8.27	12.18 ± 2.10	11.92 ± 2.43	42.22 ± 3.74
(D)Safetygym	Ep Return	6.36 ± 0.90	$\mathbf{13.64 \pm 0.05}$	12.24 ± 0.26	12.28 ± 0.87	9.66 ± 1.24	13.65 ± 0.12
	Ep CostRate(%)	$\mathbf{1.49 \pm 0.74}$	5.17 ± 0.47	4.06 ± 1.68	2.47 ± 0.41	$1.84 \pm .88$	5.41 ± 0.16
	Tot CostRate(%)	$\mathbf{3.07 \pm 0.15}$	5.79 ± 0.16	6.40 ± 0.22	4.31 ± 0.27	3.97 ± 0.42	12.80 ± 0.55
(E)Pandapush	Ep Return	$\mathbf{0.96 \pm 0.04}$	0.55 ± 0.22	0.89 ± 0.19	0.77 ± 0.35	0.92 ± 0.08	0.16 ± 0.13
	Ep CostRate(%)	$\mathbf{0.17 \pm 0.17}$	6.28 ± 4.44	0.66 ± 0.57	6.28 ± 4.44	3.48 ± 2.55	12.56 ± 4.70
	Tot CostRate(%)	$\mathbf{1.56 \pm 0.43}$	12.89 ± 6.11	6.17 ± 2.87	10.37 ± 2.52	4.28 ± 2.84	12.35 ± 4.06
(F)Safedriving	Ep Return	0.73 ± 0.04	0.73 ± 0.05	$\mathbf{0.78 \pm 0.06}$	0.74 ± 0.05	0.69 ± 0.04	0.80 ± 0.06
	Ep CostRate(%)	0.85 ± 0.14	2.59 ± 0.22	2.83 ± 0.38	1.85 ± 0.98	$\mathbf{0.66 \pm 0.10}$	3.81 ± 0.51
	Tot CostRate(%)	$\mathbf{1.30 \pm 0.17}$	3.96 ± 0.68	5.42 ± 0.25	2.33 ± 0.44	1.30 ± 0.13	6.22 ± 0.82

Table 1: Mean performance at convergence with 95% confidence interval for different algorithms on safety-critical tasks.

Tasks	Speedlimit			Pathtracking		
USL Models	Ep Return	Ep CostRate(%)	Tot CostRate(%)	Ep Return	Ep CostRate(%)	Tot CostRate(%)
Stage 1 + Stage 2	965.97 ± 5.09	0.63 ± 0.55	0.63 ± 0.06	241.74 ± 0.80	0.09 ± 0.09	5.35 ± 0.60
Stage 1 only	1016.03 ± 29.17	5.05 ± 2.69	2.53 ± 0.39	242.32 ± 0.27	0.49 ± 0.09	8.74 ± 0.72
Stage 2 only	989.12 ± 96.62	38.87 ± 15.50	13.25 ± 7.82	211.88 ± 18.28	0.62 ± 0.15	18.24 ± 1.66
Unconstrained	1382.16 ± 19.65	95.34 ± 1.03	62.56 ± 5.89	$244.80 \pm 0.54s$	24.20 ± 4.75	$42.22 \pm 3.74s$

Table 2: Ablation study for USL on two representative tasks.

constraint ($velocity < 1.5m/s$) is easily violated since the ant is able to move much faster for higher rewards. Thus, we can clearly observe that the TD3 agent achieves over 95% episodic cost rate at convergence while other risk-aware algorithms in SafeRL-kit suppress the explosion of cost rate. On other tasks, the cost signals are more sparse, such as the accidental collisions with obstacles in autonomous driving. Nevertheless, the evaluated algorithms still effectively reduce the likelihood of safety violations.

The empirical results reveal that Safety Layer and Recovery RL are comparatively ineffective in reducing the cost return. For Safety Layer, the main reasons are that the linear approximation to the cost function brings about non-negligible errors, and the single-step correction is myopic for future risks. For Recovery RL, the estimation error of Q_{risk} is the major factor affecting its efficacy.

By contrast, Off-policy Lagrangian and FAC have significantly lower cumulative costs. However, Lagrangian-based methods may suffer from the inherent issues due to primal-dual ascents. For one thing, the Lagrangian multiplier tuning causes oscillations in learning curves. For another thing, the performance may heavily depend on the Lagrangian multipliers' initialization and learning rate. According to the sensitivity analysis, we find that Lagrangian-based methods are susceptible to the learning rate of the Lagrangian multiplier(s) in stochastic primal-dual optimization. First, the oscillating λ causes non-negligible deviations in the learning curves. Second, the increasing η_λ may degrade the performance dramatically. The phenomenon is especially pronounced in FAC, which has a multiplier network to predict the state-dependent $\lambda(s;\xi)$. Consequently, we suggest to ensure $\eta_\lambda \ll \eta_\theta$ in practice.

In summary, the proposed USL is clearly effective for learning constraint-satisfying policies. First, USL achieves higher or competitive returns while adhering to (almost) zero cost return across different tasks. Second, USL features minor standard deviations and oscillations, which demonstrates its robustness. At last, USL generally converges with fewer interactions, which is crucial in sample-expensive risky environments. The underlying reason is that the optimization stage is equivalent to FAC but the state-dependent Lagrangian multipliers are reduced to a single fixed hyperparameter. Meanwhile, the consistent loss function stabilizes the training process compared with primal-dual optimiza-

Metrics	USL(ours)	Recovery	Safety Layer	Lagrangian	FAC	TD3(ref)
Normalized Inference Time	5.0	1.2	1.6	1.0	1.0	1.0
Average Inference Time (s)	25E-4	6E-4	8E-4	5E-4	5E-4	5E-4
Max Control Frequency (Hz)	400	1666	1250	2000	2000	2000

Table 3: Computational efficiency for different algorithms ($K = 20, \eta = 0.05$ for USL).

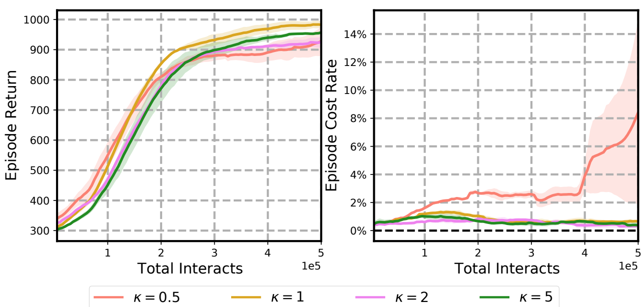

Figure 4: Sensitivity analysis on penalty factor κ.

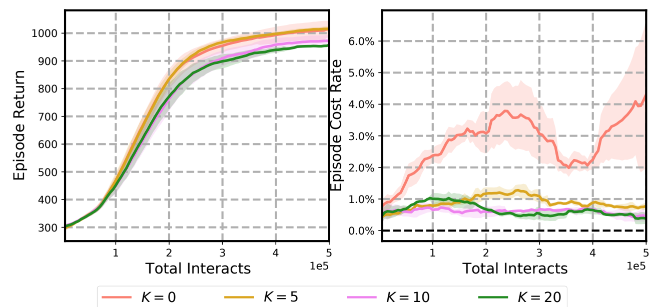

Figure 5: Sensitivity analysis on iterative number K.

tion. Furthermore, the projection stage explicitly enforces the state-wise constraints intractable in naive forward computing.

Q2: How to account for the importance of the two stages in USL?

To better understand the importance of the two stages in our approach, we perform an ablation study as shown in Table 2 and confirm that the two stages must work jointly to achieve the desired performance. An intuitive example is that the solution derived by standard RL may be far away from the desired optimal safe action on tasks such as *SpeedLimit*. Thus, directly post-optimizing over Q_c^π may not necessarily converge to a^*, and the agent still has a 38% cost rate. Instead, if the initial solution from Stage 1 is close to a^*, it can serve as a valid candidate and the cost rate goes down to 0.63%. Note that, when the unconstrained action is not that far from the safe set, such as on the *PathTracking* task, both the optimization and projection parts can effectively degrade the cost rate from 24% to less than 1%. However, using only Stage 2 is inferior on episodic return, which is an inherent flaw of the projection-based method.

Q3: How sensitive is USL to its hyper-parameters and how to tune them?

We study the impacts of two pivotal hyper-parameters in USL, namely the penalty factor κ in the training objective and the maximum iterative number K in the post-projection, on the *SpeedLimit* task. For κ, Figure 4 shows that final policies are insensitive to its value, and the learning curves are almost identical for sufficiently large κ values. By contrast, a small κ value may degenerate the first stage of USL into a "soft" regularization method. In our experiments, we find $\kappa = 5$ generally achieves good performance across different tasks. For K, Figure 5 shows that USL can enforce the hard constraint within five iterations at most decision-making steps, indicating the possibility of navigating the trade-off between constraint satisfaction and computational efficiency. We set $K = 0$ in the sensitivity study and show that the single optimization in Stage 1 can not lead to zero cost return without the aid of the post-posed projection.

Q4: How is the computational efficiency of USL with the additional iterative steps?

The two-stage architecture of USL inevitably brings concerns on computational feasibility in real-world applications. We compare different algorithms on a mainstream computing platform (Intel Core i7-9700K, NVIDIA GeForce RTX 2070). Table 3 shows that USL takes around 4-5 times the inference time of the unconstrained TD3 but still achieves an admissible 400 Hz control frequency. Having said that, we will leave the efforts on improving the time efficiency of USL to accelerate inference speed as future work.

Conclusions

In this paper, we perform a comparative study on model-free reinforcement learning toward safety-critical tasks following state-wise safety constraints. We revisit and evaluate related algorithms from the perspective of safety projection, recovery, and optimization, respectively. Furthermore, we propose Unrolling Safety Layer (USL) and demonstrate its efficacy in improving the episodic return and enhancing the safety-constraint satisfaction with an admissible computational complexity. We also present the open-sourced SafeRL-Kit and invite researchers and practitioners to incorporate domain-specific knowledge into the baselines to build more competent algorithms for their tasks.

Acknowledgments

This work is supported by the National Key R&D Program of China (2022YFB4701400/4701402), the National Natural Science Foundation of China (No. U21B6002, U1813216, 52265002), and the Science and Technology Innovation 2030 – "Brain Science and Brain-like Research" key Project (No. 2021ZD0201405).

References

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

Altman, E. 1999. *Constrained Markov decision processes*, volume 7. CRC Press.

Bohez, S.; Abdolmaleki, A.; Neunert, M.; Buchli, J.; Heess, N.; and Hadsell, R. 2019. Value constrained model-free continuous control. *arXiv preprint arXiv:1902.04623*.

Chen, B.; Francis, J.; Nyberg, J. O. E.; and Herbert, S. L. 2021. Safe Autonomous Racing via Approximate Reachability on Ego-vision. *arXiv preprint arXiv:2110.07699*.

Cheng, R.; Orosz, G.; Murray, R. M.; and Burdick, J. W. 2019. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3387–3395.

Dalal, G.; Dvijotham, K.; Vecerik, M.; Hester, T.; Paduraru, C.; and Tassa, Y. 2018. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*.

Donti, P. L.; Rolnick, D.; and Kolter, J. Z. 2021. Dc3: A learning method for optimization with hard constraints. *arXiv preprint arXiv:2104.12225*.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, 1587–1596. PMLR.

Ha, S.; Xu, P.; Tan, Z.; Levine, S.; and Tan, J. 2020. Learning to walk in the real world with minimal human effort. *arXiv preprint arXiv:2002.08550*.

Herman, J.; Francis, J.; Ganju, S.; Chen, B.; Koul, A.; Gupta, A.; Skabelkin, A.; Zhukov, I.; Kumskoy, M.; and Nyberg, E. 2021. Learn-to-race: A multimodal control environment for autonomous racing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9793–9802.

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*.

Li, Q.; Peng, Z.; Xue, Z.; Zhang, Q.; and Zhou, B. 2021. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *arXiv preprint arXiv:2109.12674*.

Liu, Y.; Ding, J.; and Liu, X. 2020. Ipo: Interior-point policy optimization under constraints. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 4940–4947.

Ma, H.; Guan, Y.; Li, S. E.; Zhang, X.; Zheng, S.; and Chen, J. 2021. Feasible actor-critic: Constrained reinforcement learning for ensuring statewise safety. *arXiv preprint arXiv:2105.10682*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Panageas, I.; Piliouras, G.; and Wang, X. 2019. First-order methods almost always avoid saddle points: The case of vanishing step-sizes. *Advances in Neural Information Processing Systems*, 32.

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

Srinivasan, K.; Eysenbach, B.; Ha, S.; Tan, J.; and Finn, C. 2020. Learning to be safe: Deep rl with a safety critic. *arXiv preprint arXiv:2010.14603*.

Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT press.

Thananjeyan, B.; Balakrishna, A.; Nair, S.; Luo, M.; Srinivasan, K.; Hwang, M.; Gonzalez, J. E.; Ibarz, J.; Finn, C.; and Goldberg, K. 2021. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3): 4915–4922.

Villani, V.; Pini, F.; Leali, F.; and Secchi, C. 2018. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55: 248–266.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Wachi, A.; and Sui, Y. 2020. Safe reinforcement learning in constrained Markov decision processes. In *International Conference on Machine Learning*, 9797–9806. PMLR.

Wachi, A.; Sui, Y.; Yue, Y.; and Ono, M. 2018. Safe exploration and optimization of constrained mdps using gaussian processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Yang, L.; Ji, J.; Dai, J.; Zhang, L.; Zhou, B.; Li, P.; Yang, Y.; and Pan, G. 2022a. Constrained Update Projection Approach to Safe Policy Optimization. In *36th Conference on Neural Information Processing Systems*.

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

Yang, T.-Y.; Zhang, T.; Luu, L.; Ha, S.; Tan, J.; and Yu, W. 2022b. Safe Reinforcement Learning for Legged Locomotion. *arXiv preprint arXiv:2203.02638*.

Yu, D.; Ma, H.; Li, S.; and Chen, J. 2022. Reachability Constrained Reinforcement Learning. In *International Conference on Machine Learning*, 25636–25655. PMLR.

Yuan, Z.; Hall, A. W.; Zhou, S.; Brunke, L.; Greeff, M.; Panerati, J.; and Schoellig, A. P. 2021. safe-control-gym: a Unified Benchmark Suite for Safe Learning-based Control and Reinforcement Learning. *arXiv preprint arXiv:2109.06325*.

Zhang, L.; Shen, L.; Yang, L.; Chen, S.; Wang, X.; Yuan, B.; and Tao, D. 2022. Penalized Proximal Policy Optimization for Safe Reinforcement Learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 3744–3750.

Zhang, Y.; Vuong, Q.; and Ross, K. W. 2020. First order constrained optimization in policy space. *arXiv preprint arXiv:2002.06506*.

Zhao, W.; He, T.; and Liu, C. 2021. Model-free safe control for zero-violation reinforcement learning. In *5th Annual Conference on Robot Learning*.