# SafeRL-Kit: Evaluating Efficient Reinforcement Learning Methods for Safe Autonomous Driving

Linrui Zhang [*1] Qin Zhang [*1] Li Shen [2] Bo Yuan [1] Xueqian Wang [1]

## Abstract

Safe reinforcement learning (RL) has achieved significant success on risk-sensitive tasks and shown promise in autonomous driving (AD) as well. Considering the distinctiveness of this community, efficient and reproducible baselines are still lacking for safe AD. In this paper, we release SafeRL-Kit to benchmark safe RL methods for AD-oriented tasks. Concretely, SafeRL-Kit contains several latest algorithms specific to zero-constraint-violation tasks, including Safety Layer, Recovery RL, off-policy Lagrangian method, and Feasible Actor-Critic. In addition to existing approaches, we propose a novel first-order method named Exact Penalty Optimization (EPO) and sufficiently demonstrate its capability in safe AD. All algorithms in SafeRL-Kit are implemented (i) under the off-policy setting, which improves sample efficiency and can better leverage past logs; (ii) with a unified learning framework, providing off-the-shelf interfaces for researchers to incorporate their domain-specific knowledge into fundamental safe RL methods. Conclusively, we conduct a comparative evaluation of the above algorithms in SafeRL-Kit and shed light on their efficacy for safe autonomous driving. The source code is available at this https URL.

## 1. Introduction

Reinforcement Learning (RL) has achieved superhuman performance in 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 has natural advantages in mastering complex skills for highly nonlinear systems like autonomous vehicles (Kiran et al., 2021).

Nevertheless, concerns about the systematic safety limit the widespread use of standard RL in real-world applications (Amodei et al., 2016). As an alternative, safe RL takes safety requirements as hard constraints and optimizes policies in the feasible domain. In recent years, it has been deemed as a practical solution to resource allocation (Liu et al., 2021), robotic locomotion (Yang et al., 2022), etc.

There have also been studies introducing safe RL into autonomous driving (AD) (Isele et al., 2018; Chen et al., 2021; Li et al., 2022). Despite those ongoing efforts, a unified benchmark is of great relevance to facilitate further research on safe AD. We notice some risk-sensitive simulated environments (Li et al., 2021; Herman et al., 2021) have been proposed, but an efficient safe RL toolkit is still absent for this community. Considering the distinctiveness of AD-oriented tasks, common code-bases (Ray et al., 2019; Yuan et al., 2021) lack the following pivotal characteristics:

**(1) Being safety-critical.** The agent must maintain zero cost-return as much as possible since any inadmissible behavior in autopilot leads to catastrophic failures. Instead, the previous code-base is built for a general-purpose with trajectory-based constraints and non-zero thresholds.

**(2) Being sample-efficient.** Off-policy algorithms can better leverage past logs and human demonstrations, which is crucial for AD. By contrast, the previous code-base requires tens of millions of interactions due to its on-policy algorithms, like CPO and PPO-L (Ray et al., 2019).

**(3) Being up-to-date.** There has been a fast-growing body of RL-based safe control. Nevertheless, the previous code-base merely contains elder baselines (Achiam et al., 2017; Chow et al., 2017) and lacks the latest advances.

**(4) Being easy-to-use.** Most work on learning-based safe AD tends to incorporate domain-specific knowledge into fundamental safe RL. Thus the toolkit is supposed to provide off-the-shelf interfaces for extended studies. However, the modules of the previous code-base are highly coupled and are implemented with the deprecated TensorFlow version.

[*]Equal contribution [1]Tsinghua Shenzhen International Graduate School, Tsinghua University, Beijing, China. [2]JD Explore Academy, Beijing, China. Correspondence to: Xueqian Wang <wang.xq@sz.tsinghua.edu.cn>.

*The 1st Workshop on Safe Learning for Autonomous Driving (SL4AD) in the 39th International Conference on Machine Learning (ICML)*, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

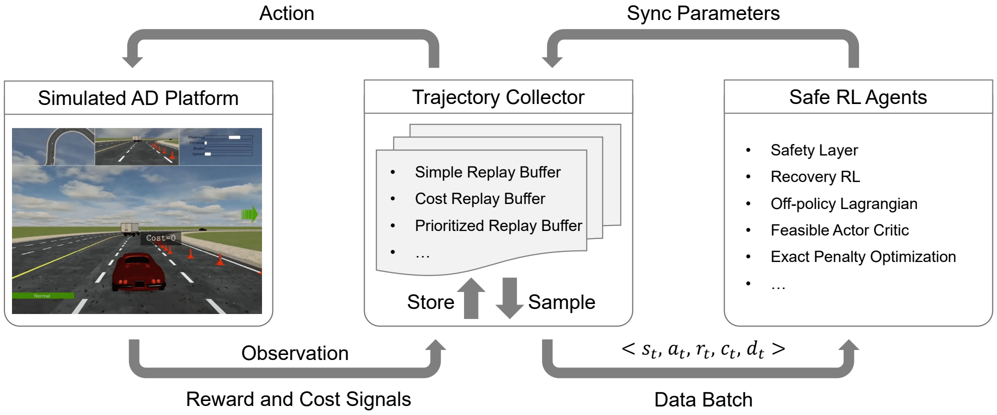


*Figure 1.* The overall framework of SafeRL-Kit. The trajectory collector interacts with specified AD environments (e.g., MetaDrive (Li et al., 2021)) and stores transitions in the memory. SafeRL-Kit contains several safe RL agents that efficiently learn from past experiences, including Safety Layer, Recovery RL, Off-policy Lagrangian, Feasible Actor Critic, and newly proposed Exact Penalty Optimization.

To provide a such as toolkit for safe RL algorithms and understand which of them are best suited for AD-oriented tasks, our contributions in this work are summarized as the following three-folds:

- We release SafeRL-Kit, which contains the latest advances in safe RL (Dalal et al., 2018; Ha et al., 2020; Thananjeyan et al., 2021; Ma et al., 2021). All algorithms are implemented efficiently under off-policy settings and with a unified training framework.
- We propose a novel first-order method coined Exact Penalty Optimization (EPO) and incorporate it into SafeRL-Kit. EPO utilizes a single penalty factor and a ReLU operator to construct an equivalent unconstrained objective. Empirical results show the simple technique is surprisingly effective and robust for AD-oriented tasks.
- We benchmark SafeRL-Kit in a representative toy environment and a simulated platform with realistic vehicle dynamics. To the best of our knowledge, this paper is the first to provide unified off-policy safe RL baselines and a fair comparison of them specific to AD.

## 2. Related Work

### 2.1. Safe RL Algorithms

A number of works tackle RL-based safe control for autonomous agents, and we divide them into three genres. The first type of method, coined as safe policy optimization, incorporates safety constraints into the standard RL objective and yields a constrained sequential optimization problem (Chow et al., 2017; Achiam et al., 2017; Zhang et al., 2020; Ma et al., 2021; Zhang et al., 2022). The second type of method, coined as safety correction, projects initial unsafe behaviors to the feasible region (Dalal et al., 2018; Zhao et al., 2021). The third type of method, coined as safety recovery, learns an additional pair of safe actor-critic to take over control when encountering potential risks (Thananjeyan et al., 2021; Yang et al., 2022).

There have also been studies on safe RL specific to AD-oriented tasks. Isele et al. (2018) utilizes a prediction module to generate masks on dangerous behaviors, which merely works in discrete action spaces. Wen et al. (2020) extend Constrained Policy Optimization (CPO) (Achiam et al., 2017) to AD and employ synchronized parallel actors to accelerate the convergence speed for on-policy CPO. Chen et al. (2021) take the ego-camera view as input and train an additional recovery policy via a heuristic objective based on Hamilton-Jacobi reachability. Li et al. (2022) propose a human-in-loop approach to learn safe driving efficiently.

### 2.2. Safe RL Benchmarks

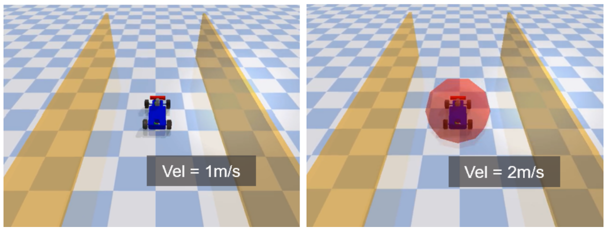
For general scenarios, a set of benchmarks are commonly used to evaluate the efficacy of safe RL algorithms. The classic environments[1] include Robot with Limit Speed (Zhang et al., 2020), Circle and Gather (Achiam et al., 2017), etc. Safety-gym[2] (Ray et al., 2019) contains several tasks (goal, button, push) and agents (point, car, doggo) that are representative in robot control problems. Meanwhile, the authors provide popular baselines[3], including CPO and some on-policy Lagrangian methods. Safe-control-gym[4] (Yuan et al., 2021) bridges the gap between control and RL communities. The authors also developed an open-sourced toolkit supporting both model-based and data-driven control techniques.

[1] https://github.com/SvenGronauer/Bullet-Safety-Gym

[2] https://github.com/openai/safety-gym

[3] https://github.com/openai/safety-starter-agents

[4] https://github.com/utiasDSL/safe-control-gym



(a) Cost Signal = 0 (b) Cost Signal = 1

*Figure 2.* SpeedLimit Benchmark. The vehicle is rewarded for driving along the avenue, but receives a cost signal if $vel > 1.5m/s$.

For AD-oriented tasks, there have been some existing environments for safe driving. Li et al. (2021) release Metadrive[5] that benchmarks reinforcement learning algorithms for vehicle autonomy, including safe exploitation and exploration. Herman et al. (2021) propose Learn-to-Race[6] that focuses on safe control in high speed. Nevertheless, it still lacks a set of strong baselines specific to the AD community considering its distinctiveness depicted above in Section 1. To our best knowledge, this paper is the first to provide unified off-policy safe RL baselines and a fair comparison of them for the purpose of autonomous driving.

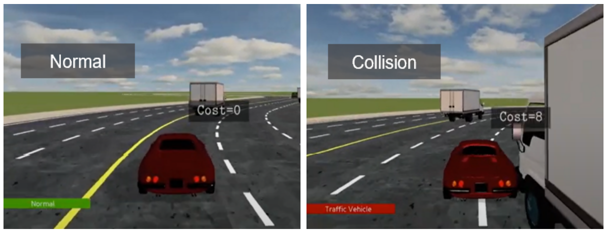
## 3. Preliminaries

A Markov Decision Process (MDP) (Sutton & Barto, 1998) is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mu, \gamma)$. $\mathcal{S}$ 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. $\mu : \mathcal{S} \mapsto [0, 1]$ is the initial state distribution. $\gamma$ is the discount factor for future reward. A stationary policy $\pi : S \mapsto P(A)$ maps the given states to probability distributions over action space. The goal of standard RL is to find the optimal policy $\pi^*$ that maximizes the expected discounted return $J_R(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$, where $\tau = \{(s_t, a_t)\}_{t\geq 0}$ is a sample trajectory and $\tau \sim \pi$ accounts for the distribution over trajectories depending on $s_0 \sim \mu, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)$.

A Constrained Markov Decision Process (CMDP) (Altman, 1999) extends MDP to $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{C}, \mu, \gamma)$. The cost function $\mathcal{C} : \mathcal{S} \times \mathcal{A} \mapsto [0, +\infty]$ reflects the violation of systematic safety. The goal of safe RL is to find

$$\pi^* = \arg\max_\pi J_R(\pi) \quad \text{s.t.} \quad \{a_t\}_{t\geq 0} \text{ is feasible.}$$

In a CMDP, the cost function is typically constrained in the following two ways. The first is *Instantaneous Constrained MDP*. This type of Safe RL formualtion requires the selected actions enforce the constraint at every decision-making step, namely $C(s_t, a_t) \leq \epsilon$. The second is *Cumulative Constrained MDP*. This type of Safe RL formualtion requires the discounted sum of cost signals in the whole trajectory within a certain threshold, namely $J_C(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t C(s_t, a_t)\right] \leq d$.



(a) Cost Signal = 0 (b) Cost Signal = 1

*Figure 3.* MetaDrive Benchmark. The vehicle aims to reach virtual markers, but receives a cost signal if it collides with obstacles and other vehicles or it is out of the road.

## 4. Problem Setup

In this paper, we develop SafeRL-Kit to evaluate efficient RL algorithms for safe autonomous driving on existing benchmarks. We simplify the cost function as the following risk-indicator:

$$C(s, a) = \begin{cases} 1, & \text{if the transition is unsafe} \\ 0, & \text{otherwise} \end{cases}. \tag{1}$$

This formulation is generalizable to different AD-oriented tasks without cumbersome reward and cost shaping. The goal of the autonomous vehicle is to reach the destination as fast as possible while adhering to zero cost signals at every time steps. Specifically, we conduct comparative evaluations on a representative toy environment and a simulated platform with realistic vehicle dynamics respectively.

### 4.1. SpeedLimit Benchmark

The task is inspired by Zhang et al. (2020), as illustrated in Figure 2. In SpeedLimit task, the agent is a four-wheeled race-car whose observation is ego position, velocity and yaw. The selected action controls the Revolution Per Minute (RPM) and steering of wheels. The agent is rewarded for approaching $x_{dest} = +\infty$ and the cost function is

$$C(s, a) = \begin{cases} 1, & \text{if vehicle's velocity} > 1.5m/s \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

The toy environment is simple yet representative since speed control is a classic problem in vehicle autonomy. Besides, the speed limit is easy to reach and thus undesirable algorithms may violate the safety constraint at almost every time step. That is, the toy environment enables us to see which

[5] https://github.com/metadriverse/metadrive

[6] https://github.com/learn-to-race/l2r

*Table 1.* Comparison of different safe reinforcement learning algorithms for AD-oriented tasks.

| Algorithm | Constraint Type | | Policy Type | |
|---|---|---|---|---|
| | Cumulative/Instantaneous | State-wise/Trajectory-wise | Deterministic | Stochastic |
| CPO (Ray et al., 2019) | Cumulative | Trajectory-wise | × | √ |
| PPO-L (Ray et al., 2019) | Cumulative | Trajectory-wise | × | √ |
| TRPO-L (Ray et al., 2019) | Cumulative | Trajectory-wise | × | √ |
| Safety Layer | Instantaneous | State-wise | √ | × |
| Recovery RL | Cumulative | State-wise | √ | √ |
| Off-policy Lagrangian | Cumulative | Trajectory-wise | √ | √ |
| Feasible Actor-Critic | Cumulative | State-wise | √ | √ |
| Exact Penalty Optimization | Cumulative | Both | √ | √ |

algorithms can effectively degrade the dense cost return and are best suited for safe AD tasks.

### 4.2. MetaDrive Benchmark

This task is inspired by Li et al. (2021), as illustrated in Figure 3. Metadrive is a compositional, lightweight and realistic platform for vehicle autonomy. Most importantly, it provides pre-defined environments for safe policy learning in autopilots. Concretely, the observation is encoded by a vector containing ego-state, navigation information and surrounding information detected by the Lidar. We control the speed and steering of the car to hit virtual land markers for rewards, and the cost function is defined as

$$C(s,a) = \begin{cases} 1, & \text{if collides or out of the road} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

It worth mentioning that we set the traffic density twice than the original paper to construct a more challenging scenario.

## 5. Efficient Safe RL Algorithms

### 5.1. Overall Implementation

The current version of SafeRL-Kit contains some latest RL-based methods, including *Safety Layer* (Dalal et al., 2018), *Recovery RL* (Thananjeyan et al., 2021), *Off-policy Lagrangian* (Ha et al., 2020), *Feasible Actor-Critic* (Ma et al., 2021) and newly proposed *Exact Penalty Optimization*. We compare above methods along with some existing on-policy baselines (Ray et al., 2019) in Table 1.

Before diving into algorithmic details, we first explain the overall implementation of SafeRL-Kit and its benefits:

**(1)** The adopted algorithms address safe policy learning from different perspectives (Safety Layer for safety correction; Recovery RL for safety recovery; Lagrangian, FAC, and EPO for constrained optimization). Thus, users can combine AD-specific knowledge with the proper type of safe RL baselines in their studies.

**(2)** All the algorithms are implemented under the off-policy Actor-Critic architecture. Thus, they enjoy better sample efficiency and can leverage human demonstration if needed.

**(3)** All the algorithms are implemented with a unified training framework. By default, all networks are MLPs with (256,256) hidden layers activated via the ReLU function. The essential updates of backbone networks follow TD3 (Fujimoto et al., 2018) without pre-training processes. Thus, we can conduct a fair comparison to see which of them are best suited for AD-oriented tasks.

### 5.2. Safety Layer

Safety Layer, added on top of the original policy network, conducts a quadratic-programming-based constrained optimization to find the ”nearest” action to the feasible region.

Specifically, Safety Layer utilizes a parametric linear model

$$C(s_t, a_t) \approx g(s_t;\omega)^\top a_t + c_{t-1} \tag{4}$$

to approximate the single-step cost function with supervised training and yields the following QP problem

$$\begin{aligned} a_t^* = \; & \arg\min_a \frac{1}{2}||a - \mu_\theta(s)||^2 \\ & \text{s.t.} \quad g(s_t;\omega)^\top a_t + c_{t-1} \leq \epsilon, \end{aligned} \tag{5}$$

which projects the unsafe action back to the feasible region.

Since there is only one compositional cost signal in our problem, the closed-form solution of problem (5) is

$$a_t^* = \mu_\theta(s_t) - \left[\frac{g(s_t;\omega)^\top \mu_\theta(s) + c_{t-1} - \epsilon}{g(s_t;\omega)^\top g(s_t;\omega)}\right]^+ g(s_t;\omega) \tag{6}$$

Thus, Safety Layer is the type of method that addresses state-wise, instantaneous constraints.

By the way, the $g_\omega$ is trained from offline data in Dalal et al. (2018). SafeRL-Kit instead learns the linear model with the policy network synchronously, considering the side-effect of distribution shift. We employ a warm-up in the training process to avoid meaningless, inaccurate corrections.

*Table 2.* Hyper-parameters of different safety-aware algorithms in SafeRL-Kit.

| Hyper-parameter | Safety Layer | Recovery RL | Lagrangian | FAC | EPO |
|---|---|---|---|---|---|
| Cost Limit | 0.02 | 0.1 | 0.1 | 0.1 | 0.1 |
| Reward Discount | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |
| Cost Discount | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |
| Warm-up Ratio | 0.2 | 0.2 | N/A | N/A | N/A |
| Batch Size | 256 | 256 | 256 | 256 | 256 |
| Critic LR | 3E-4 | 3E-4 | 3E-4 | 3E-4 | 3E-4 |
| Actor LR | 3E-4 | 3E-4 | 3E-4 | 3E-4 | 3E-4 |
| Safe Critic LR | 3E-4 | 3E-4 | 3E-4 | 3E-4 | 3E-4 |
| Safe Actor LR | N/A | 3E-4 | N/A | N/A | N/A |
| Multiplier LR | N/A | N/A | 1E-5 | 1E-5 | N/A |
| Multiplier Init | N/A | N/A | 0.0 | N/A | N/A |
| Policy Delay | 2 | 2 | 2 | 2 | 2 |
| Multiplier Delay | N/A | N/A | N/A | 12 | N/A |
| Penalty Factor | N/A | N/A | N/A | N/A | 5 |

### 5.3. Recovery RL

The critical insight behind Recovery RL is to introduce an additional policy that recovers potential unsafe states. Consequently, it trains two independent RL agents instead of solving a cumbersome constrained optimization problem.

Specifically, Recovery RL learns a safe critic to estimate the future probability of constraint violation as

$$Q^{\pi}_{\text{risk}}(s_t, a_t) = c_t + (1 - c_t)\gamma \mathbb{E}_{\pi} Q^{\pi}_{\text{risk}}(s_{t+1}, a_{t+1}). \quad (7)$$

This formulation is slightly different from the standard Bellman equation since it assumes the episode terminates when the agent receives a cost signal. We found in experiments that such an early stopping makes it intractable for agents to master desirable skills in AD. Thus, we remove the early-stopping condition but still preserve the original formulation of $Q^{\pi}_{\text{risk}}$ in (7) since it limits the upper bound of the safe critic and eliminates the over-estimation in Q-learning.

In the phase of policy execution, the recovery policy takes over the control when the predicted value of the safe critic exceeds the given threshold, as

$$a_t = \begin{cases} \pi_{\text{task}}(s_t), & \text{if } Q^{\pi}_{\text{risk}}\big(s_t, \pi_{\text{task}}(s_t)\big) \le \epsilon \\ \pi_{\text{risk}}(s_t), & \text{otherwise} \end{cases} \quad (8)$$

The optimization objective of $\pi_{\text{task}}$ is to maximize the cumulative rewards, whereas the goal of $\pi_{\text{risk}}$ is to minimize $Q^{\pi}_{\text{risk}}$, namely to degrade the potential risk of the agent.

It is important to store $a_{\text{task}}$ and $a_{\text{risk}}$ simultaneously in the replay buffer, and utilize them to train $\pi_{\text{task}}$ and $\pi_{\text{risk}}$ respectively in Recovery RL. This technique ensures that $\pi_{\text{task}}$ can learn from the new MDP, instead of proposing same unsafe actions continuously.

Similar to Safety Layer, Recovery RL in SafeRL-Kit also has a warm-up stage where $Q^{\pi}_{\text{risk}}$ is trained but is not utilized.

### 5.4. Off-policy Lagrangian

Lagrangian Relaxation is commonly-used to address constrained optimization problem. Safe RL as well can be formulated as a constrained sequential optimization problem

$$\begin{aligned} &\max_{\pi} \mathop{\mathbb{E}}_{s\sim\mu} V^{\pi}_0(s) \\ &\text{s.t.} \mathop{\mathbb{E}}_{s\sim\mu} U^{\pi}_0(s) \le \epsilon, \end{aligned} \quad (9)$$

where $V^{\pi}_0(s) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r_t \middle| s_0 = s\right]$ and $U^{\pi}_0(s) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t c_t \middle| s_0 = s\right]$.

Strong duality holds for primal problem (9) (Paternain et al., 2022), thus it can be tackled via the dual problem

$$\max_{\lambda\ge 0}\min_{\pi} \mathop{\mathbb{E}}_{s\sim\mu} -V^{\pi}_0(s) + \lambda\big(U^{\pi}_0(s) - \epsilon\big). \quad (10)$$

The off-policy objective of problem (10) in the parametric space (Ha et al., 2020) can be formulated as

$$\max_{\lambda\ge 0}\min_{\theta} \mathbb{E}_{\mathcal{D}} - Q^{\pi}(s, \pi_\theta(s)) + \lambda\big(Q^{\pi}_c(s, \pi_\theta(s)) - \epsilon\big). \quad (11)$$

Stochastic primal-dual optimization (Luenberger et al., 1984) is applied here to update primal and dual variables alternatively, which follows as

$$\begin{cases} \theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathbb{E}_{\mathcal{D}}\big(-Q^{\pi}(s, \pi_\theta(s)) + \lambda Q^{\pi}_c(s, \pi_\theta(s))\big) \\ \lambda \leftarrow \big[\lambda + \eta_\lambda \mathbb{E}_{\mathcal{D}}\big(Q^{\pi}_c(s, \pi_\theta(s)) - \epsilon\big)\big]^+ \end{cases} \quad (12)$$

Notably, the timescale of primal variable updates is required to be faster than the timescale of Lagrange multipliers. Thus, we set $\eta_\theta \gg \eta_\lambda$ in SafeRL-Kit.

### 5.5. Feasible Actor-Critic

The constraint of Off-policy Lagrangian in Section 5.4 is based on the expectation of whole trajectories, which inevitably allows some unsafe roll-outs. Ma et al. (2021) introduce a new concept, namely state-wise constraints for cumulative cost-return which follows as

$$\begin{aligned} &\max_{\pi} \mathop{\mathbb{E}}_{s\sim\mu} V_0^\pi(s) \\ &\text{s.t. } U_0^\pi(s) \le \epsilon, \forall s \in \mathcal{I}_\mathcal{F}. \end{aligned} \tag{13}$$

Here $s \in \mathcal{I}_\mathcal{F}$ stands for all "feasible" initial states. Also, their theoretical results show that problem (13) is a stricter version than problem (9), which provides strong safety guarantees for state-wise safe control.

The dual problem of (13) is derived by rescaling the state-wise constraints and follows as

$$\max_{\lambda\ge 0}\min_{\pi} \mathop{\mathbb{E}}_{s\sim\mu} -V_0^\pi(s) + \lambda(s)\big(U_0^\pi(s) - \epsilon\big). \tag{14}$$

The distinctiveness of problem (14) is there are infinitely many Lagrangian multipliers that are state-dependent. In SafeRL-Kit, we employ an neural network $\lambda(s;\xi)$ activated by *Softplus* function to map the given state $s$ to its corresponding Lagrangian multiplier $\lambda(s)$.

The primal-dual ascents of policy network is similar to (12); the updates of multiplier network is given by

$$\xi \leftarrow \xi + \eta_\xi \nabla_\xi \mathbb{E}_\mathcal{D} \lambda(s;\xi)\big(Q_c^\pi(s, \pi_\theta(s)) - \epsilon\big). \tag{15}$$

Besides, SafeRL-Kit also sets a different interval schedule $m_\pi$ (for $\pi_\theta$ delay steps) and $m_\lambda$ (for $\lambda_\xi$ delay steps) to stabilize the training process (Ma et al., 2021).

### 5.6. Exact Penalty Optimization

In this paper, we propose a simple-yet-effective approach motivated by the exact penalty method.

**Theorem 5.1.** *Considering the following two problems*

$$\min f(x) \;\text{ s.t. } g_i(x) \le 0, i = 1, 2, ... \tag{P}$$

$$\min f(x) + \kappa \cdot \sum_i \max\{0, g_i(x)\} \tag{Q}$$

*Suppose $\lambda^*$ is the optimal Lagrange multiplier vector of problem (P). If the penalty factor $\kappa \ge ||\lambda^*||_\infty$, problem (P) and problem (Q) share the same optimal solution set.*

*Proof.* See our recent work (Zhang et al., 2022). □

The above theorem enables us to construct an equivalent function whose unconstrained minimizing points also solve the previous constrained problem. Meanwhile, the unconstrained problem can tackle multiple constraints with exactly one consistent penalty factor.

**Algorithm 1** State-wise Exact Penalty Optimization

```
Require: deterministic policy network $\pi(s;\theta)$; critic networks $\hat{Q}(s,a;\phi)$ and $\hat{Q}_c(s,a;\varphi)$
for t in $1, 2, ...$ do
    $a_t = \pi(s_t;\theta) + \epsilon, \;\; \epsilon \sim \mathcal{N}(0, \sigma)$.
    Apply $a_t$ to the environment.
    Store the transition $(s_t, a_t, s_{t+1}, r_t, c_t, d_t)$ in $\mathcal{B}$.
    Sample a mini-batch of $N$ transitions from $\mathcal{B}$.
    $\varphi \leftarrow \arg\min_\varphi \mathbb{E}_\mathcal{B}\big[\hat{Q}_c(s,a;\varphi) - \big(c + \gamma_c(1 - d)\hat{Q}_C(s', \pi(s';\theta);\varphi)\big)\big]^2$.
    $\phi \leftarrow \arg\min_\phi \mathbb{E}_\mathcal{B}\big[\hat{Q}(s,a;\phi) - \big(r + \gamma(1 - d)\hat{Q}(s', \pi(s';\theta);\phi)\big)\big]^2$.
    $\theta \leftarrow \arg\min_\theta \mathbb{E}_\mathcal{B}\big[ - \hat{Q}(s, \pi(s;\theta);\phi) + \kappa \cdot \max\{0, \hat{Q}_c(s, \pi(s;\theta);\varphi) - \delta\}\big]$.
end for
```

Thus, we simplify Lagrangian-based methods (i.e., Off-policy Lagrangian and FAC) with this technique, considering that the single-constrained optimization problem (9) and the multi-constrained optimization problem (13) are suited for exact penalty method in Theorem 5.1. In this way, we can employ a single minimization on primal variables with fixed penalty terms instead of cumbersome min-max optimization over both primal and dual variables.

Below we merely summarize the state-wise Exact Penalty Optimization (EPO) in Algorithm 1 as an alternative to FAC, since FAC provides stricter safety guarantees but suffers from the oscillation and instability of the multiplier network. The off-policy surrogate objective of state-wise EPO follows as

$$\ell(\theta) = \mathbb{E}_\mathcal{D} - Q^\pi(s, \pi_\theta(s)) + \kappa\big[Q_c^\pi(s, \pi_\theta(s)) - \epsilon\big]^+, \tag{16}$$

where $\kappa$ is a fixed, sufficiently large hyper-parameter.

## 6. Empirical Analysis

We benchmark RL-based algorithms on SpeedLimit task (Zhang et al., 2020) and MetaDrive platform (Li et al., 2021). Below, we give a comparative evaluation according to the empirical results.

**Unconstrained Reference.** We utilize TD3 (Fujimoto et al., 2018) as the unconstrained reference for upper bounds of reward performance and constraint violations. For the SpeedLimit task (500 max_episode_horizon), TD3 exceeds the velocity threshold at almost every step with a near 100% cost rate. For the MetaDrive environment (1000 max_episode_horizon), the agent receives sparse cost signals

*Table 3.* Mean performance with normal 95% confidence for safety-aware algorithms on benchmarks.

| Environments | | Safety Layer | Recovery RL | Lagrangian | FAC | EPO |
|---|---|---|---|---|---|---|
| SpeedLimit | Ep-Reward | $651.59 \pm 10.70$ | $623.67 \pm 99.58$ | $565.50 \pm 69.29$ | $631.55 \pm 34.92$ | $\mathbf{684.86 \pm 3.19}$ |
| | Ep-Cost | $76.30 \pm 9.07$ | $187.14 \pm 96.50$ | $7.28 \pm 3.11$ | $7.83 \pm 5.23$ | $\mathbf{5.44 \pm 0.53}$ |
| | CostRate | $0.33 \pm 0.01$ | $0.43 \pm 0.06$ | $0.06 \pm 0.01$ | $0.07 \pm 0.01$ | $\mathbf{0.02 \pm 0.01}$ |
| MetaDrive | SuccessRate | $0.73 \pm 0.05$ | $\mathbf{0.78 \pm 0.06}$ | $0.74 \pm 0.05$ | $0.68 \pm 0.04$ | $0.73 \pm 0.05$ |
| | Ep-Cost | $12.91 \pm 1.10$ | $14.18 \pm 1.92$ | $9.23 \pm 4.88$ | $\mathbf{3.29 \pm 0.50}$ | $4.29 \pm 0.71$ |
| | CostRate | $0.04 \pm 0.001$ | $0.05 \pm 0.001$ | $0.02 \pm 0.01$ | $\mathbf{0.01 \pm 0.01}$ | $0.01 \pm 0.01$ |

when it collides with obstacles or is out of the road. Besides, the cost signals are encoded into the reward function; otherwise, it would be too hard to learn desirable behaviors (Li et al., 2021). Consequently, TD3 with reward-shaping (TD3-RS) would not have that high cumulative costs as it does in the toy environment.

**Overall Performance.** The mean performances are summarized in Table 2 and the learning curves over five seeds are shown in Figure 4 and 5. We conclude that Safety Layer and Recovery RL are less effective in degrading cost return. They still have around 10% safety violations in SpeedLimit, and the safety improvement in MetaDrive is also limited. As 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. As for Recovery RL, the estimation error of $Q_{\text{risk}}$ is probably the biggest problem affecting the recovery effects. By contrast, Off-policy Lagrangian and FAC have significantly lower cumulative costs. However, the Lagrangian-based methods have the inherent problems from primal-dual ascents. For one thing, the Lagrangian multiplier tuning causes oscillations of learning curves. For another thing, those algorithms are susceptible to Lagrangian multipliers' initialization and learning rate. We conclude that constrained optimization still outperforms safety correction and recovery if the hyper-parameters are appropriately settled. At last, we find that the newly proposed EPO is surprisingly effective for learning safe AD. In SpeedLimit, it converges to a high plateau quickly while adhering to an almost zero cost return. In MetaDrive, it is still competitive with SOTA baselines. We regard the underlying reason as that EPO is an equivalent form to FAC but reduces state-dependent Lagrangian multipliers to one fixed hyper-parameter. The consistent loss function stabilizes the training process compared with primal-dual optimization.

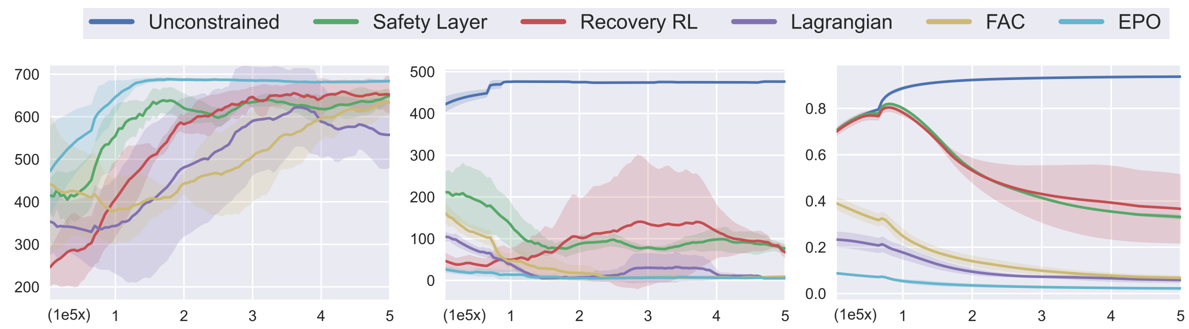
**Sensitivity Analysis.** In this paper, we study the sensitivity to hyper-parameters of Lagrangian-based methods and EPO in Figure 6 and Figure 7 respectively. We found that Lagrangian-based methods are susceptible to the learning rate of the Lagrangian multiplier(s) in stochastic primal-dual optimization. First, the oscillating $\lambda$ causes non-negligible deviations in the learning curves. Besides, the increasing $\eta_\lambda$ may degrade the performance dramatically. The phenomenon is especially pronounced in FAC, which has a multiplier network to predict the state-dependent $\lambda(s; \xi)$. Thus, we suggest $\eta_\lambda \ll \eta_\theta$ in practice. As for EPO, we find if the penalty factor $\kappa$ is too small, the cost return may fail to converge. Nevertheless, if $\kappa$ is sufficiently large, the learning curves are robust and almost identical. Thus, we suggest $\kappa > 5$ in experiments and a grid search for better performance.

**Sample Complexity.** Considering the difficulty of the above two tasks, we run $5 \times 10^5$ and $1 \times 10^6$ interactive steps respectively to obtain admissible results. Notably, previous on-policy codebases require significantly more samples for convergence; for example, Ray et al. (2019) run $1 \times 10^7$ interactive steps even for toy environments. Thus, SafeRL-Kit with off-policy implementations is much more sample-efficient compared to theirs, emphasizing the applicability of SafeRL-Kit to data-expensive AD-oriented tasks.

## 7. Further Discussion

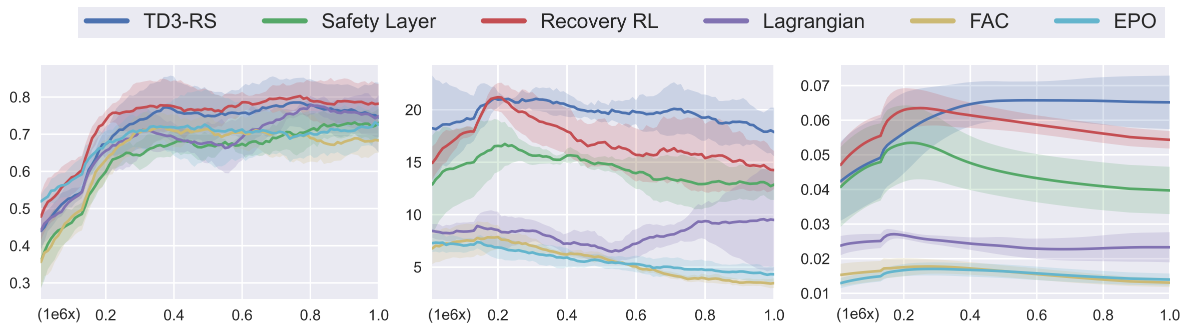
The released SafeRL-kit contains several SOTA off-policy safe RL methods that are suited for safety-critical autonomous driving. We conduct the comparative evaluation of those baselines over one representative toy environment and one simulated AD platform, respectively. The proposed Exact Penalty Optimization in this paper is easy-to-implement and surprisingly effective on AD-oriented tasks. We think future work on SafeRL-kit from two aspects:

- The off-policy implementation of SafeRL-Kit can naturally leverage offline data, including past logs and human demonstrations, which are commonly used and highly effective for AD-oriented tasks.
- We only benchmark safe RL methods with vector input (ego-state, navigation information, Lidar signals, etc.) in this paper. Nevertheless, vision-based AD is still less studied in the current version of SafeRL-Kit.



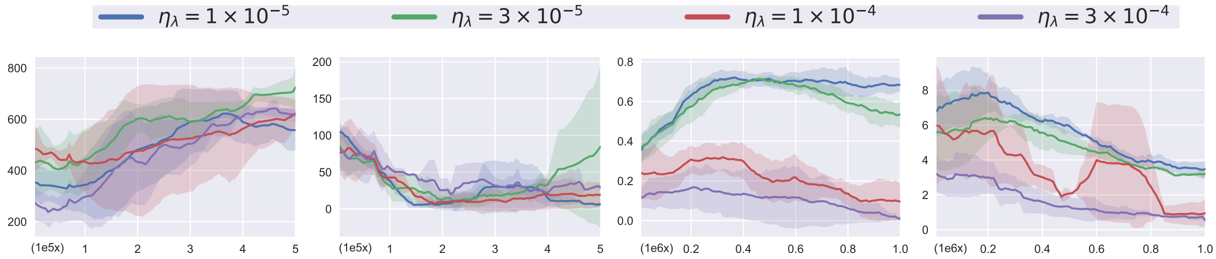
(a) Eval Episode Reward (b) Eval Episode Cost (c) Training Cost rate

*Figure 4.* Learning curves on the SpeedLimit benchmark. The x-axis is the number of interactions with the simulator (500,000 total).



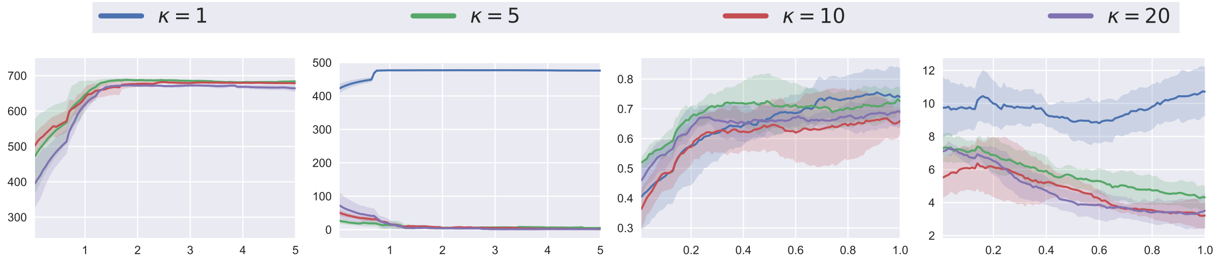
(a) Eval Success Rate (b) Eval Episode Cost (c) Training Cost rate

*Figure 5.* Learning curves on the MetaDrive Benchmark. The x-axis is the number of interactions with the simulator (1,000,000 total).



(a) Reward-Lag-SpeedLimit (b) Cost-Lag-SpeedLimit (c) Reward-FAC-MetaDrive (d) Cost-FAC-MetaDrive

*Figure 6.* Sensitivity studies of Lagrangian-based methods. The first two figures are reward and cost plots of Off-policy Lagrangian on SpeedLimit task with different $\lambda$ learning rates. The last two figures are success rate and cost plots of Feasible Actor-Critic on MetaDrive benchmark with different $\lambda(s; \xi)$ learning rates.



(a) Eval Episode Reward (b) Eval Episode Cost (c) Training Cost rate (d) Training Cost rate

*Figure 7.* Sensitivity studies of Exact Penalty Optimization. The first two figures are reward and cost plots of EPO on the SpeedLimit task with different penalty factors $\kappa$. The last two figures are the success rate and cost plots of EPO on the MetaDrive benchmark with different penalty factors $\kappa$.

## References

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

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

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Chen, B., Francis, J., Nyberg, J. O. E., and Herbert, S. L. Safe autonomous racing via approximate reachability on ego-vision. *arXiv preprint arXiv:2110.07699*, 2021.

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

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

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

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

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

Isele, D., Nakhaei, A., and Fujimura, K. Safe reinforcement learning on autonomous vehicles. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–6. IEEE, 2018.

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

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

Li, Q., Peng, Z., and Zhou, B. Efficient learning of safe driving policy via human-ai copilot optimization. *arXiv preprint arXiv:2202.10341*, 2022.

Liu, Y., Ding, J., and Liu, X. Resource allocation method for network slicing using constrained reinforcement learning. In *2021 IFIP Networking Conference (IFIP Networking)*, pp. 1–3. IEEE, 2021.

Luenberger, D. G., Ye, Y., et al. *Linear and nonlinear programming*, volume 2. Springer, 1984.

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

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. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Paternain, S., Calvo-Fullana, M., Chamon, L. F., and Ribeiro, A. Safe policies for reinforcement learning via primal-dual methods. *IEEE Transactions on Automatic Control*, 2022.

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

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

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

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

Wen, L., Duan, J., Li, S. E., Xu, S., and Peng, H. Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–7. IEEE, 2020.

Yang, T.-Y., Zhang, T., Luu, L., Ha, S., Tan, J., and Yu, W. Safe reinforcement learning for legged locomotion. *arXiv preprint arXiv:2203.02638*, 2022.

Yuan, Z., Hall, A. W., Zhou, S., Brunke, L., Greeff, M., Panerati, J., and Schoellig, A. P. safe-control-gym: a unified benchmark suite for safe learning-based control and reinforcement learning. *arXiv preprint arXiv:2109.06325*, 2021.

Zhang, L., Shen, L., Yang, L., Chen, S.-Y., Yuan, B., Wang, X., and Tao, D. Penalized proximal policy optimization for safe reinforcement learning. *arXiv preprint arXiv:2205.11814*, 2022.

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

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