Reachability Constrained Reinforcement Learning

Dongjie Yu [*] [1] **Haitong Ma** [*] [1 2] **Shengbo Eben Li** [1] **Jianyu Chen** [3 4]

Abstract

Constrained reinforcement learning (CRL) has gained significant interest recently, since safety constraints satisfaction is critical for real-world problems. However, existing CRL methods constraining discounted cumulative costs generally lack rigorous definition and guarantee of safety. In contrast, in the safe control research, safety is defined as persistently satisfying certain state constraints. Such persistent safety is possible only on a subset of the state space, called feasible set, where an optimal largest feasible set exists for a given environment. Recent studies incorporate feasible sets into CRL with energy-based methods such as control barrier function (CBF), safety index (SI), and leverage prior conservative estimations of feasible sets, which harms the performance of the learned policy. To deal with this problem, this paper proposes the reachability CRL (RCRL) method by using reachability analysis to establish the novel self-consistency condition and characterize the feasible sets. The feasible sets are represented by the safety value function, which is used as the constraint in CRL. We use the multi-time scale stochastic approximation theory to prove that the proposed algorithm converges to a local optimum, where the largest feasible set can be guaranteed. Empirical results on different benchmarks validate the learned feasible set, the policy performance, and constraint satisfaction of RCRL, compared to CRL and safe control baselines.

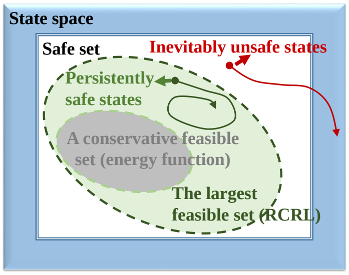

State space
Safe set
Inevitably unsafe states
Persistently safe states
A conservative feasible set (energy function)
The largest feasible set (RCRL)

Figure 1. The intuitive relationship among the state space, the safe states, feasible sets and the largest feasible set.

1. Introduction

Constrained reinforcement learning (CRL) has gained growing attention due to the safety requirements in the practical applications of RL. The safety specifications in common CRL methods are expected discounted cumulative costs (Altman, 1999; Achiam et al., 2017; Tessler et al., 2019; Yang et al., 2020). However, the main deficiency of expected costs is averaging the potential danger at a state to the whole trajectory. For example, the autonomous vehicle should always keep a safe distance to other traffic participants but not keep the cumulative or average distance during a period when the collision might happen at a single time step. Therefore, constraints imposed on expected cumulative costs lack rigorous definition and guarantee of safety.

Meanwhile, the persistent state constraint satisfaction in the safe control research clarifies the safety of states with rigorous definitions (Liu & Tomizuka, 2014; Ames et al., 2019; Choi et al., 2021). Rich theoretical and practical techniques for ensuring safety in such settings are provided, where an important fact is that only a subset of states can be guaranteed safe persistently, called the feasible set. Outside of the feasible set, even temporally safe states will violate the constraints inevitably in the future, no matter what policies to choose, as shown in Figure 1. For example, if a vehicle with high speed is too close to a front obstacle, it is doomed to crash since the deceleration capability is limited. Therefore, accurately identifying feasible sets in CRL can significantly affect the performance and safety of the learned policy.

Some recent RL studies adopt energy-based methods to handle persistent safety and characterize feasible sets. Representatives include control barrier function (CBF) (Ames

[*] Equal contribution [1] School of Vehicle and Mobility, Tsinghua University, Beijing, China [2] John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts, USA. This work was conducted during Haitong's graduate study at Tsinghua University. [3] Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China [4] Shanghai Qizhi Institute, Shanghai, China. Correspondence to: Shengbo Eben Li <lishbo@tsinghua.edu.cn>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

et al., 2019; Ma et al., 2021a) and safety index (SI) (Liu & Tomizuka, 2014; Ma et al., 2022). However, these methods rely on prior formulation of the energy function, which results in conservative feasible sets (as Figure 1 shows), causing unsatisfying performance sacrifice. Hamilton-Jacobi (HJ) reachability analysis is another branch in the safe control research, which identifies the theoretical largest feasible set (Lygeros et al., 1999; Mitchell et al., 2005; Bansal et al., 2017). Recently, some pioneering studies migrated HJ reachability analysis to model-free RL (Fisac et al., 2019; Hsu et al., 2021). However, these works obtain only the safest policies, leaving the performance criterion (e.g., reward optimization) unconsidered. This safety-only design significantly limits broader applications of HJ reachability analysis in RL.

This paper proposes reachability constrained reinforcement learning (RCRL), which learns the optimal safe policy satisfying persistent safety within the identified largest feasible set. We leverage reachability analysis to establish the novel self-consistency condition and characterize the feasible sets. The feasible sets are represented by the safety value function. Intuitively, the function describes the worst constraint-violation in the long term, and its sub-zero level set is the feasible set. We use the multi-time scale stochastic approximation theory (Borkar, 2009; Chow et al., 2017) to prove that the proposed algorithm converges to a local optimum. Empirical results on low-dimensional problems validate the correctness of the learned feasible sets. Further experiments conducted on complex benchmarks such as safe-control-gym (Yuan et al., 2021) and Safety-Gym (Achiam & Amodei, 2019) indicate that RCRL achieves competitive performance while maintaining constraint-satisfaction. Our main contributions are:

- We are the first to introduce reachability constraints into CRL, which is critical for learning a nearly optimal and persistently safe policy upon its corresponding feasible set. Compared to other feasible set characterization methods, RCRL enlarges the feasible sets and reduces the policy conservativeness.
- We use the multi-time scale stochastic approximation theory to prove that RCRL converges to a locally optimal policy, which also persistently satisfies the state constraints across the entire largest feasible set if the initialization of states is general.
- Comprehensive experiments demonstrate that the proposed RCRL method outperforms CRL and safe control baselines in terms of final performance and constraint satisfaction.

2. Related Work

Constrained reinforcement learning (CRL) problems are usually formulated as constrained Markov decision process (CMDP) (Altman, 1999; Brunke et al., 2021). Constrained optimization approaches are adopted to solve CRL problems: (1) penalty function (Guan et al., 2022); (2) Lagrangian methods (Tessler et al., 2019; Chow et al., 2017; Duan et al., 2021b; Ma et al., 2021b); (3) trust-region methods (Achiam et al., 2017; Yang et al., 2020) and (4) other approaches such as conservative updates (Bharadhwaj et al., 2021). CMDP relies on the expected discounted cumulative costs and a hand-crafted threshold to improve the safety of policies. However, a proper threshold relies on engineering intuitions and varies in different tasks (Qin et al., 2021).

Characterizing feasible sets is a critical and open problem in safe control research (Brunke et al., 2021). Feasible sets, also called recoverable sets (Thomas et al., 2021), are usually represented by safety certificates. Representative safety certificates include energy functions such as CBF (Ma et al., 2021a; Choi et al., 2021; Luo & Ma, 2021) and SI (Liu & Tomizuka, 2014). The core idea of the energy function is that the energy of a dynamical system dissipates when it is approaching the safer region (Ames et al., 2019). Nevertheless, energy-based methods suffer from conservative or inaccurate feasible sets (Ma et al., 2022). HJ reachability analysis is a promising way towards general and rigorous derivation of feasible sets (Lygeros et al., 1999; Mitchell et al., 2005). However, it is quite difficult to obtain the largest feasible set because it is represented by a non-trivial partial differentiable equation, whose analytical solution is often intractable (Bansal et al., 2017). Machine learning, especially RL approaches, is adopted to deal with this problem (Fisac et al., 2019; Bansal & Tomlin, 2021; Hsu et al., 2021). However, most of the existing reachability studies only care about safety while ignoring other metrics, especially the optimality criterion, limiting reachability analysis approaches from broader applications. Thananjeyan et al. (2021) pretrain a feasible set indicator for switching between the optimal policy and a back-up safe controller during training. A recent CRL study utilizes reachability analysis to learn a purely safe back-up policy (Chen et al., 2021). Then the agent switches between the safe and optimal policies when interacting with the environment. Different from these switching-based methods, we only learn one policy tackling safety and optimality simultaneously.

3. When RL Meets Feasible Sets

3.1. Notation

We formulate the CRL problem as an MDP with a deterministic dynamic (a reasonable assumption in safe control problems), defined by the tuple $\langle \mathcal{S}, \mathcal{A}, P, r, h, c, \gamma \rangle$ where (1) the state space $\mathcal{S}$ and the action space $\mathcal{A}$ are bounded (possibly continuous); (2) unknown transition probability $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \{0, 1\}$ represents the dynamics; (3) $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function; (4) $h : \mathcal{S} \mapsto \mathbb{R}$

is the state constraint. c is called the cost signal where $c(s) = \mathbb{1}_{h(s)>0}$, indicating we get 1 if $h(s) \leq 0$ is violated and otherwise 0. (5) $\gamma \in (0, 1)$ is the discount factor. A deterministic policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ chooses action a_t at state s_t at time t. The initial state distribution is denoted as $d_0(s)$ while $d_\pi(s, a)$ is the state-action marginals following π. We denote the initial state set as $\mathcal{S}_0 \triangleq \{s \mid d_0(s) > 0\}$.

The objective of standard RL is to find a policy maximizing the expected return (discounted cumulative rewards) $\mathcal{J}(\pi) = \mathbb{E}_{s_t,a_t\sim d_\pi} \sum_t[\gamma^t r(s_t, a_t)]$. A value function $V^\pi(s) \triangleq \mathbb{E}_{s_t,a_t\sim d_\pi} \sum_t[\gamma^t r(s_t, a_t)|s_0 = s]$ represents the potential return in the future from state s, satisfying $V^\pi(s) = r(s, \pi(s)) + \gamma\mathbb{E}_{s'\sim P}[V^\pi(s')]$. One can easily find that $\mathcal{J}(\pi) = \mathbb{E}_{s\sim d_0(s)}[V^\pi(s)]$. In CRL, one can define discounted cumulative costs as cost return $\mathcal{J}_c(\pi) = \mathbb{E}_{s_t,a_t\sim d_\pi} \sum_t[\gamma^t c(s_t)]$ and cost value function $V_c^\pi(s) = c(s) + \gamma\mathbb{E}_{s'\sim P}[V^\pi(s')]$ similarly.

We need a few extended notations beyond standard ones. We denote $(s_t^\pi \mid s_0 = s, \pi), t \in \mathbb{N}$ as the state trajectory $\{s_0^\pi, s_1^\pi, \cdots \mid s_0 = s, \pi\}$ induced by π from $s_0 = s$. Let $h(s_t^\pi \mid s_0 = s), t \in \mathbb{N}$ specify the state constraint sequence of the trajectory $\{h(s_0^\pi), h(s_1^\pi), h(s_2^\pi), \cdots \mid s_0 = s, \pi\}$. We also denote $h(s_t^\pi \mid s_0 = s) \leq 0, t \in \mathbb{N}$ as persistently satisfying the constraint, i.e. $h(s_t^\pi) \leq 0, \forall t \in \mathbb{N}$.

3.2. Definition of Feasible Sets

Generally speaking, a state is considered *safe* if it satisfies the state constraint $h(s) \leq 0$, such as keeping distance from obstacles. The safe set is defined by the set of all safe states:

Definition 3.1 (Safe set)**.**

$$\mathcal{S}_c \triangleq \{s \mid h(s) \leq 0\}.$$

However, as stated in Section 1, some safe states would go dangerous no matter what policy we choose, such as a high-speed vehicle close to a front obstacle. Therefore, what really matters for meaningful safety guarantee is not the *temporary* safety but the *persistent* safety, i.e., $h(s_t^\pi \mid s_0 = s) \leq 0, t \in \mathbb{N}$. Otherwise, the system will be dangerous sooner or later. In other words, we need to characterize those states starting from which the policy π is able to keep the system constraint-satisfactory. We define the feasible set as the set of all the states which are able to be safe persistently.

Definition 3.2 (Feasible set)**.** The feasible set of *a specific* policy π can be defined as

$$\mathcal{S}_f^\pi \triangleq \{s \in \mathcal{S} \mid h(s_t^\pi \mid s_0 = s) \leq 0, t \in \mathbb{N}\}.$$

A policy π is feasible if $\mathcal{S}_f^\pi \neq \emptyset$ and otherwise it is infeasible. The largest feasible set $\mathcal{S}_f$ is a subset of $\mathcal{S}$ composed of states from which there exists *at least one* policy that keeps the system satisfying the constraint, i.e.,

$$\mathcal{S}_f \triangleq \{s \in \mathcal{S} \mid \exists\pi, h(s_t^\pi \mid s_0 = s) \leq 0, t \in \mathbb{N}\}.$$

To guarantee that all the states in the trajectory $\{s_t^\pi \mid s_0 = s, \pi\}, t \in \mathbb{N}$ are safe, we only need to guarantee that the worst-case i.e., the *maximum* violation in the trajectory is below zero, which brings the following definition:

Definition 3.3 (Safety value function)**.**

$$V_h^\pi(s) \triangleq \max_{t\in\mathbb{N}} h(s_t^\pi \mid s_0 = s), \tag{1}$$

is the worst constraint violation in the long term.

The safety value of a given state s varies when the policy π changes. The best value we can get is the one where we choose the policy minimizing the constraint violation and we call it the optimal safety value function:

$$V_h^*(s) \triangleq \min_\pi \max_{t\in\mathbb{N}} h(s_t^\pi \mid s_0 = s).$$

One can easily observe that the (largest) feasible set is the sub-zero level set of the (optimal) safety value function, i.e.,

$$\mathcal{S}_f^\pi = \{s \mid V_h^\pi(s) \leq 0\}, \quad \mathcal{S}_f = \{s \mid V_h^*(s) \leq 0\}.$$

and clearly $\mathcal{S}_f^\pi \subseteq \mathcal{S}_f$ for $\forall\pi$.

Safety problems in reality are more about the worst-case through time other than the cumulative or average costs (Fisac et al., 2019), where the latter is often the case in previous CRL. The safety value function measures the safety of the most dangerous state on the trajectory generated by π. Specifically, if $V_h^\pi(s) \leq 0$, the most dangerous state is safe, so the safety of the system can be guaranteed. Otherwise, the policy π would definitely cause state constraint violations in the future. Therefore, once $V_h^\pi(s) \leq 0$, which we call that the *reachability constraint* is fulfilled, the agent is guaranteed to be inside the feasible set since the state constraint could be satisfied persistently.

3.3. Computation of the Safety Value Function

Although the safety value function does not use the discounted cumulative formulation like the common value functions in RL, we can still use the temporal difference learning technique to get it. Fisac et al. (2019) proposes the following lemma about the optimal safety value function:

Lemma 3.4 (Safety Bellman equation (SBE))**.**

$$V_h^*(s) = \max\left\{h(s), \min_{a\in\mathcal{A}} V_h^*(s')\right\} \tag{2}$$

holds for $\forall s \in \mathcal{S}$, where s' is the successive state of state s.

We extend Lemma 3.4 to a general form applicable to any policy, which is called the self-consistency condition:

Theorem 3.5 (Self-consistency condition of the safety value function)**.**

$$V_h^\pi(s) = \max\{h(s), V_h^\pi(s')\} \tag{3}$$

holds for $\forall s \in \mathcal{S}$ and $\forall \pi$, where s' is the successive state at state s following π.

Proof. See Appendix B.1. □

Remark 3.6 (Optimality). The largest feasible set can be obtained by solving SBE (2) for the optimal safety value function. However, this will lead to a policy always pursuing the lowest constraint violation, i.e., a purely safe policy. The purely safe policy does not tackle the optimality specifications. For example, a robotic arm should catch the objects as *quickly* as possible with only *bounded* torques. We do not need to choose the safest action at states which are interior points of the feasible set (non-safety-critical) (Asayesh et al., 2021). Intuitively, the safest action has to be taken only at states on the boundary of the feasible set (safety-critical states). To address this issue, some studies design the switching rules between the purely safe policy and optimal policy (Chen et al., 2021). In contrast, this paper learns only one policy tackling safety and optimality simultaneously. Theorem 3.5 could compute the safety value for this unified policy.

Remark 3.7 (Scalability). Besides characterizing the persistent safety of agents, reachability constraints are also scalable to constraints on cumulative quantities (i.e., safe budget) similar to conventional CRL, enabling it to be a general constraint formulation. A safe budget for the whole trajectory can be seen as the remaining budget constraint on any state during the trajectory. Let the budget constraint be $\sum_t c(s_t|s) - \eta(s) \leq 0$, where $c(s_t)$ is the consumption of one step and $\eta(s)$ is the remaining budget at state s. We define $h(s) = -\eta(s)$ and thus $V_h^\pi(s)$ equals the worst-case over-budget whose being greater than 0 is unacceptable.

4. Reachability Constrained Reinforcement Learning

In this section, we formally propose the novel RCRL problem which guarantees the persistent safety of the policy. Furthermore, if the initialization of the state covers the largest feasible set, the feasible set solved by RCRL equals the largest one defined in Definition 3.2. Then the RCRL algorithm with a convergence guarantee will be devised.

4.1. Problem Statement

Given an MDP defined in Section 3.1 and an initial state distribution d_0, RCRL aims to find the optimal policy π^* to the following optimization problem:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{s\sim d_0(s)}[V^\pi(s)\cdot \mathbb{1}_{s\in\mathcal{S}_f} - V_h^\pi(s)\cdot \mathbb{1}_{s\notin\mathcal{S}_f}] \\ \text{subject to} \quad & V_h^\pi(s) \leq 0, \forall s \in \mathcal{S}_f \cap \mathcal{S}_0, \end{aligned} \tag{RCRL}$$

where $\mathbb{1}_A = 1$ holds when the event A is `true` and otherwise $\mathbb{1}_A = 0$. Intuitively, **for initial states inside the largest feasible set**, i.e. $s \in \mathcal{S}_0 \cap \mathcal{S}_f$, (RCRL) aims to maximize the expected return and ensure the persistent safety when following this policy. However, **for initial states outside the largest feasible set**, i.e., $s \in \mathcal{S}_0 \setminus \mathcal{S}_f \triangleq \{s \mid s \in \mathcal{S}_0, s \notin \mathcal{S}_f\}$, the state constraint $h(s) \leq 0$ will be violated sooner or later and it is impossible to satisfy the reachability constraint. Thus, it is meaningless to optimize the return of these infeasible states, and we only try to find the safest actions by minimizing the safety value functions.

The formulation in (RCRL) is different from the common CRL formulations in (Achiam et al., 2017; Tessler et al., 2019; Yang et al., 2020) where the constraint is imposed on the expectation of cost return:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{s\sim d_0(s)}[V^\pi(s)] \\ \text{subject to} \quad & \mathbb{E}_{s\sim d_0(s)}[V_c^\pi(s)] \leq \eta, \end{aligned} \tag{4}$$

where η is the cost threshold, V_c^π is the cost value function. However, as stated in Section 2, choosing η is tricky and it is hard to migrate the expectation-based CRL formulation to CRL problems with state constraints, such as (RCRL).

Specially, if the the initial states cover the largest feasible set, the solution to (RCRL) also has the largest feasible set, which is given by the following proposition:

Proposition 4.1 (The largest feasible set)**.** *Assuming $\mathcal{S}_0 \cap \mathcal{S}_f \neq \emptyset$, for any feasible π of problem* (RCRL)*, we have $\mathcal{S}_f^\pi = \mathcal{S}_f$ if $\mathcal{S}_f \subseteq \mathcal{S}_0$.*

Proof. See Appendix B.2. □

Overall, (RCRL) has three significant advantages: (1) Compared to conventional CRL approaches, (RCRL) considers the vital persistent safety of the system because every single time step in the future is guaranteed to be safe when reachability constraints are satisfied. (2) Compared to HJ reachability analysis studies, RCRL considers performance optimality besides safety. (3) Compared to other RL methods with feasible sets, any feasible policy of (RCRL) renders the largest feasible set if the initial states cover the feasible sets. The largest feasible sets brings less conservativeness and better performance because the policy could drive the system towards states with higher return.

4.2. Lagrangian-based Algorithm with Statewise Constraints

We leverage the Lagrange multiplier method to solve problem (RCRL), which is a common approach to CRL (Chow et al., 2017; Achiam & Amodei, 2019; Tessler et al., 2019). The key idea of Lagrangian-based methods is to descend in π and ascend in the multiplier λ using the gradients of the Lagrangian $\mathcal{L}(\pi, \lambda)$ w.r.t. π and λ and to finally reach the optimal policy which satisfies the constraints. Notably, constraints in (RCRL) are imposed on *each state* in $\mathcal{S}_0 \cap \mathcal{S}_f$, which is significantly different from typical ones on only the expectation among the states in (Achiam et al., 2017; Chow et al., 2017; Tessler et al., 2019). We call these type of constraints as *statewise constraints*. In this case, the multiplier is not longer a scalar but a vector (infinite-dimension in infinite states cases). Some preliminary studies discuss statewise constraints on the density of the state distribution (Chen & Ames, 2019; Qin et al., 2021). A more general Lagrangian-based solution for statewise constraints with *an approximation to the multipliers* is discussed recently in (Ma et al., 2021b; 2022). Without loss of generality, we denote the statewise multiplier as a function $\lambda : \mathcal{S} \mapsto [0, +\infty) \cup \{+\infty\}$. Then the Lagrangian of (RCRL) can be formulated as:

$$\begin{aligned}\mathcal{L}(\pi, \lambda) =& \mathbb{E}_{s\sim d_0}\left[-V^\pi(s)\cdot \mathbb{1}_{s\in\mathcal{S}_f} + V_h^\pi(s)\cdot\mathbb{1}_{s\notin\mathcal{S}_f}\right]\\ &+\int_{\mathcal{S}_f\cap\mathcal{S}_0}\lambda(s)V_h^\pi(s)\mathrm{d}s\end{aligned} \tag{5}$$

The most significant issue when we are tackling (5) is that we cannot obtain the largest feasible set $\mathcal{S}_f$ in advance, which means $\mathcal{S}_f \cap \mathcal{S}_0$ is unknown. However, the initial distribution d_0 is usually accessible and we propose a surrogate Lagrangian in the form of expectation w.r.t. d_0 instead:

$$\hat{\mathcal{L}}(\pi, \lambda) = \mathbb{E}_{s\sim d_0}[-V^\pi(s) + \lambda(s)V_h^\pi(s)] \tag{6}$$

A common choice of the initial distribution is the uniform distribution in the safe set $\mathcal{S}_c$ which covers the largest feasible set. However, it is inevitable that there are initial states outside $\mathcal{S}_f$. The constraint $V_h^\pi(s) \le 0$ can never be satisfied for those states outside $\mathcal{S}_f$. Thus, each corresponding multiplier $\lambda(s)$ will go to $+\infty$ because $\lambda(s)$ tries to maximize the product of itself and a positive scalar $V_h^\pi(s)$, resulting in the divergence of (6). We can set a large upper bound $\lambda_{\max}$ to the statewise multiplier to avoid the divergence. We claim that when $\lambda_{\max} \to +\infty$, solving the surrogate Lagrangian is equivalent to solving the original one:

Proposition 4.2 (Equivalent Lagrangian)**.** *Assume both* (5) *and* (6) *have the unique optimal solution. If we denote* $\pi^* = \arg\min_\pi \max_\lambda \mathcal{L}$*,* $\hat{\pi}^* = \arg\min_\pi \max_\lambda \hat{\mathcal{L}}$*, we have* $\lim_{\lambda_{\max}\to+\infty} \hat{\pi}^* = \pi^*$.

Proof. See Appendix B.3. □

Remark 4.3. The surrogate Lagrangian can be regarded as an expected weighted sum of $-V^\pi(s)$ and $V_h^\pi(s)$ using weights $\lambda(s)$. We separate the surrogate Lagrangian into two parts here, the expectations on feasible and infeasible initial states. For those feasible initial states, $\lambda(s)$ is finite and the surrogate Lagrangian calculates the expected weighted sums of $-V^\pi(s)$ and $V_h^\pi(s)$. For those infeasible initial states, $\lambda(s) \to +\infty$, so $-V^\pi(s)$ is ignored in the weighted sums and the surrogate Lagrangian is only dominated by the expected $V_h^\pi(s)$.

Algorithm 1 Template for actor-critic RCRL

```
Input: MDP $M$ with constraint $h(\cdot)$, critic and safety value function learning rate $\beta_1(k)$, actor learning rate $\beta_2(k)$, multiplier learning rate $\beta_3(k)$
Initialization: q-function parameters $\omega = \omega_0$, safety q-function parameters $\phi = \phi_0$, policy parameters $\theta = \theta_0$, multiplier parameters $\xi = \xi_0$
for $k = 0, 1, \dots$ do
  Initialize state $s_0 \sim d_0$.
  for $t = 0$ to $T - 1$ do
    Select action $a_t = \pi_\theta(s_t)$, observe next state $s_{t+1}$, reward $r_t$ and constraint $h(s_t)$
    Critic update $\omega_{k+1} = \omega_k - \beta_1(k)\hat{\nabla}_\omega \mathcal{J}_Q(\omega)$
    Safety value update
      $\phi_{k+1} = \phi_k - \beta_1(k)\hat{\nabla}_\phi \mathcal{J}_{Q_h}(\phi)$
    Actor update $\theta_{k+1} = \Gamma_\Theta\left(\theta_k - \beta_2(k)\hat{\nabla}_\theta \mathcal{J}_\pi(\theta)\right)$
    Multiplier update
      $\xi_{k+1} = \Gamma_\Xi\left(\xi_k + \beta_3(k)\hat{\nabla}_\xi \mathcal{J}_\lambda(\xi)\right)$
  end for
end for
return parameters $\omega, \phi, \theta, \xi$
```

Hence, solving problem (RCRL) can be approximated by finding the saddle point of the surrogate (6):

$$\min_\pi \max_\lambda \hat{\mathcal{L}}(\pi, \lambda). \tag{7}$$

Consider the common actor-critic framework with state-action value functions. We have the state-action (safety) value $Q(s, \pi(s)) = V^\pi(s)$ and $Q_h(s, \pi(s)) = V_h^\pi(s)$ due to the deterministic dynamics and policy. We also adopt parameterized Q-function $Q(s, a; \omega)$, safety Q-value function $Q_h(s, a; \phi)$, a policy $\pi(s; \theta)$, and the statewise multiplier $\lambda(s; \xi)$. The Lagrangian thus becomes $\hat{\mathcal{L}}(\theta, \xi)$. Note that sometimes we use $Q_\omega, \pi_\theta, \lambda_\xi$ for short. Now we derive the objectives of the parameterized functions.

The Q-value function update is the standard one in popular RL (Sutton & Barto, 2018), which can be seen in Appendix A. The safety Q-value function of the current policy is updated according to the self-consistency condition in Theorem 3.5, i.e., minimizing the mean squared error:

$$\mathcal{J}_{Q_h}(\phi) = \mathbb{E}_{s\sim\mathcal{D}}\left[1/2\left(Q_h(s, a; \phi) - \hat{Q}_h(s, a)\right)^2\right], \tag{8}$$

where

$$\begin{aligned}\hat{Q}_h(s, a) =& (1-\gamma)h(s)\\ &+\gamma\mathbb{E}_{s'\sim P}[\max\{h(s), Q_h(s', \pi(s'); \phi)\}],\end{aligned} \tag{9}$$

$\mathcal{D}$ is the distribution of previously sampled states and actions (i.e., d_π), or a replay buffer, and a is the action taken at s. Note that the discounted version of self-consistency condition is for convergence in Appendix B.4, as in (Fisac et al., 2019).

As aforementioned, the purpose of policy π_θ is to descend the Lagrangian while the multiplier tries to ascend it:

$$\begin{aligned} &\mathcal{J}_\pi(\theta) = \mathcal{J}_\lambda(\xi) \\ &\quad= \mathbb{E}_{s\sim\mathcal{D}}[-Q(s,\pi_\theta(s);\omega) + \lambda_\xi(s) Q_h(s,\pi_\theta(s);\phi)]. \end{aligned} \tag{10}$$

Algorithm 1 provides the pseudo-code of an actor-critic version of RCRL. The algorithm alternates between interacting with the environment and updating the parameter vectors with stochastic gradients $\hat{\nabla}_\omega \mathcal{J}_Q(\omega)$, $\hat{\nabla}_\theta \mathcal{J}_\pi(\theta)$, $\hat{\nabla}_\theta \mathcal{J}_\pi(\theta)$, and $\hat{\nabla}_\xi \mathcal{J}_\lambda(\xi)$, whose derivation can be seen in Appendix A. In the algorithm, the $\Gamma_\Psi(\psi)$ operator projects a vector $\psi \in \mathbb{R}^\kappa$ to the closet point in a compact and convex set $\Psi \subseteq \mathbb{R}^\kappa$, i.e., $\Gamma_\Psi(\psi) = \arg\min_{\hat{\psi}\in\Psi} \|\hat{\psi} - \psi\|^2$ where ψ is denoted as any one of θ, ξ. These projection operators are necessary for the convergence of the actor-critic algorithm (Chow et al., 2017). A policy-gradient version of RCRL is designed similarly in Algorithm 2.

5. Convergence Analysis

Under moderate assumptions, we can provide a convergence guarantee of Algorithm 1. The convergence analysis follows heavily from the convergence proof of multi-time scale stochastic approximation algorithms (Chow et al., 2017). We also utilize theorems of combining the ODE (ordinary differential equation) viewpoint and stochastic approximation from (Borkar, 2009). We first introduce the necessary assumptions.

Assumption 5.1 (Finite MDP)**.** The MDP is finite (finite state and action space, i.e., $|\mathcal{S}| < \infty, |\mathcal{A}| < \infty$), and $\mathcal{S}$ and $\mathcal{A}$ are both bounded. The first-hitting time of the MDP $T_{\pi,s}$ is bounded almost surely over all policy π and all initial states $s \in \mathcal{S}_0$. We refer the upper bound as T. The reward function and constraint value of a single step are bounded by $r_{\max}$ and $h_{\max}$, respectively. Hence, the value function is upper bounded by $r_{\max}/(1-\gamma)$.

Assumption 5.2 (Strict Feasibility)**.** There exists a policy $\pi(\cdot;\theta)$ such that $V_h^{\pi_\theta}(s) \leq 0, \forall s \in \mathcal{S}_0 \cap \mathcal{S}_f \neq \emptyset$.

Assumption 5.3 (Differentiability)**.** For any state-action pair (s,a), $Q(s,a;\omega)$ and $Q_h(s,a;\phi)$ are continuously differentiable in ω and ϕ, respectively. Moreover, $\nabla_a Q(s,a;\omega)$ and $\nabla_a Q_h(s,a;\phi)$ are Lipschitz functions in a, for $\forall s \in \mathcal{S}, \forall \omega \in \Omega$, and $\forall \phi \in \Phi$. For $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}$, $\nabla_a Q(s,a;\omega)$ is a Lipschitz function in ω and $\nabla_a Q_h(s,a;\phi)$ is a Lipschitz function in ϕ. For any state s, $\pi(s;\theta)$ is continuously differentiable in θ and $\nabla_\theta \pi(s;\theta)$ is a Lipschitz function in θ. For any state s, $\lambda(s;\xi)$ is continuously differentiable in ξ and $\nabla_\xi \lambda(s;\xi)$ is a Lipschitz function in ξ.

Assumption 5.4 (Step Sizes)**.** The step size schedules $\{\beta_1(k)\}, \{\beta_2(k)\}$, and $\{\beta_3(k)\}$ satisfy

$$\begin{aligned} &\sum_k \beta_1(k) = \sum_k \beta_2(k) = \sum_k \beta_3(k) = \infty \\ &\sum_k \beta_1(k)^2, \sum_k \beta_2(k)^2, \sum_k \beta_3(k)^2 < \infty \\ &\beta_3(k) = o\left(\beta_2(k)\right), \beta_2(k) = o\left(\beta_1(k)\right). \end{aligned}$$

These step-size schedules satisfy the standard conditions for stochastic approximation algorithms, and ensure that the critic update is on the fastest time scale $\{\beta_1(k)\}$, the policy update is on the intermediate time scale $\{\beta_2(k)\}$, and the multiplier is on the slowest one $\{\beta_3(k)\}$. Now we come to the position where the convergence of actor-critic RCRL can be provided.

Theorem 5.5. *Under Assumption 5.1 to 5.4, the policy sequence updated in Algorithm 1 converges almost surely to a locally optimal policy for the reachability constrained policy optimization problem* (RCRL)*.*

Proof. See Appendix B.4. □

The conditions for convergence may be strict and ideal such that we have to make some simplification and approximation to make the RCRL algorithm tractable and scalable for high-dimensional and continuous problems. We discuss the gap between the necessary assumptions and the practical situation in Appendix C.1. More details about implementation can be found in Appendix C.

6. Experiments

We aim to answer the following through our experiments:

- Can RCRL learn the largest feasible sets using neural networks approximation of safety value functions?
- Does RCRL outperform CRL methods based on cost value constraints in terms of fewer violations?
- Does RCRL perform better than methods based on energy function with respect to performance optimality benefiting from the largest feasible set?

Benchmarks. We implement both on- and off-policy RCRL and compare them with different CRL baselines. Experiments include that: (1) use double-integrator (Fisac et al., 2019) which has an analytical solution to check the correctness of feasible set learned by RCRL; (2) validate the scalability of RCRL to nonlinear control problems, specifically, a 2D quadrotor trajectory tracking task in safe-control-gym

(Yuan et al., 2021), and (3) classical safe learning benchmark Safety-Gym (Achiam & Amodei, 2019). Details about each benchmark will be introduced per subsection.

Baseline Algorithms. Details about algorithms can be seen in Appendix C. Besides RCRL, we test following baselines: (1) **Lagrangian**-based algorithms whose constraint is about the discounted cumulative costs (an implementation of RCPO (Tessler et al., 2019)); (2) **Reward shaping** method with a fixed coefficient penalty added to the reward; (3) **CBF**-based algorithms whose constraint is about $B(s) \triangleq \dot{h}(s) + \mu h(s) \leq 0$ where $\mu \in (0, 1)$ is a hyperparameter; and (4) **SI**-based methods that defines an SI $\varphi(s) = \sigma - (-h(s))^n + k\dot{h}(s)$ and sets constraints

$$\varphi(s') - \max\{\varphi(x) - \eta_D, 0\} \leq 0, \tag{11}$$

where σ, n, k, η_D are hyperparameters.

6.1. Double Integrator: Comparison to Ground Truth

We demonstrate that RCRL can learn the largest feasible sets when controlling the double integrator. The reason why we choose the double integrator is that it is a simple dynamical system where we can use numerical solution by the level set toolbox to obtain the ground truth about the largest feasible sets (also called *HJ viability kernels*) (Mitchell, 2008). Double integrator is a 2D dynamical system, where the system states and the dynamics are denoted as

$$s = [x_1, x_2]^T, \quad \dot{s} = [x_2, a], \tag{12}$$

where the control limits of action a is $a \in [-0.5, 0.5]$. The safety constraint is $\|s\|_\infty \leq 5$. The reward is designed as $r_t = \|s\|^2 + a_t^2$.

Baselines. In addition to the ground truth using the level set toolbox (Mitchell, 2008), we introduce two discrete approximations of feasible sets using model predictive control (MPC) utilizing CBF constraints and terminal constraints, respectively (Ma et al., 2021a; Mayne et al., 2000). These two MPC baselines are named as MPC-CBF and MPC-Terminal, respectively.

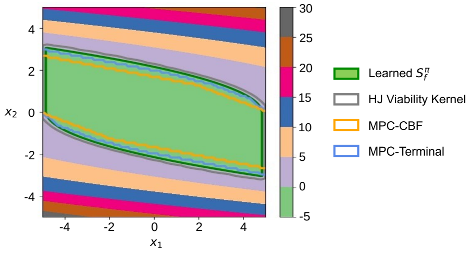


Figure 2. Learned $\mathcal{S}_f^\pi$ of the double integrator.

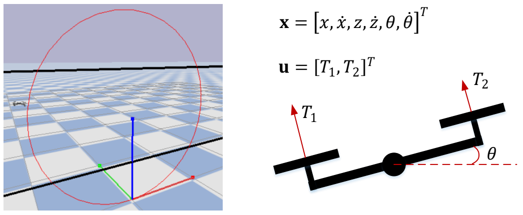
The learned result is shown in Figure 2. It depicts that the learned $\mathcal{S}_f^\pi$ exactly approximate the largest feasible set or the HJ viability kernel given by the level set toolbox. MPC-Terminal also identifies the feasible sets with some discretization error. MPC-CBF has a smaller feasible set since the conservativeness of energy-based methods.

6.2. Safe-control-gym: Quadrotor Trajectory Tracking

The 2D quadrotor trajectory tracking task comes from safe-control-gym (Yuan et al., 2021) and is shown in Figure 3(a), where the circle trajectory is marked as red and the constraint for the quadrotor is to keep itself between the two black lines. Schematics of the 2D quadrotor are shown in Figure 3(b), where (x, z) and $(\dot{x}, \dot{z})$ are the position and velocity of the COM (center of mass) of the quadrotor in the xz-plane, and θ and $\dot{\theta}$ are the pitch angle and pitch angle rate, respectively. The task for the quadrotor is to track the moving waypoint on the circle trajectory by controlling the normalized thrusts while maintaining its altitude z between $[0.5, 1.5]$, i.e.,

$$h(s) = \max\{0.5 - s^{(2)}, s^{(2)} - 1.5\}.$$

Details about the state and action space, reward function can be seen in Appendix D.1.


$\mathbf{x} = [x, \dot{x}, z, \dot{z}, \theta, \dot{\theta}]^T$
$\mathbf{u} = [T_1, T_2]^T$
T_1
T_2
θ

(a) Snapshot of environment, where red line is the reference and black lines are constraint boundaries.

(b) Schematics, state and input of the 2D quadrotor

Figure 3. safe-control-gym environment

Baselines. We implement an off-policy version of RCRL in safe-control-gym based on SAC (Haarnoja et al., 2018), forming our Reachable Actor Critic (RAC). Other off-policy baselines are all implemented based on SAC for fairness, including: (1) **SAC-Lagrangian**, (2) **SAC-Reward Shaping**, (3) **SAC-CBF**, and (4) **SAC-SI**.

Figure 4 demonstrates performance with respect to the average return and constraint violation rate of the five algorithms. RAC (blue line) learns a zero-violation policy and reaches near-optimal tracking accuracy. In contrast, though SAC-CBF does not violate the constraint as well, the tracking error is quite large because it just moves horizontally due to a conservative policy. The constraints of SAC-SI require the SI to be below zero and to decrease when it is beyond zero, which explains that it makes the quadrotor fly beyond

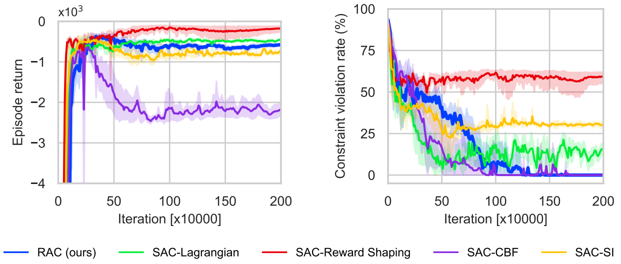


Figure 4. Performance of algorithms on safety-control-gym. The first two figures are training curves on the quadrotor trajectory tracking task. All results are averaged on 5 independent runs and the shaded regions are the 95% confidence intervals.

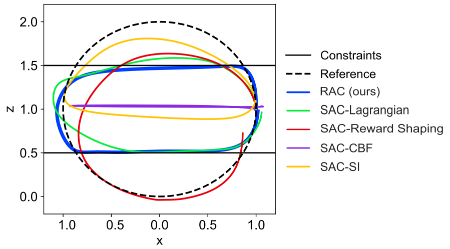
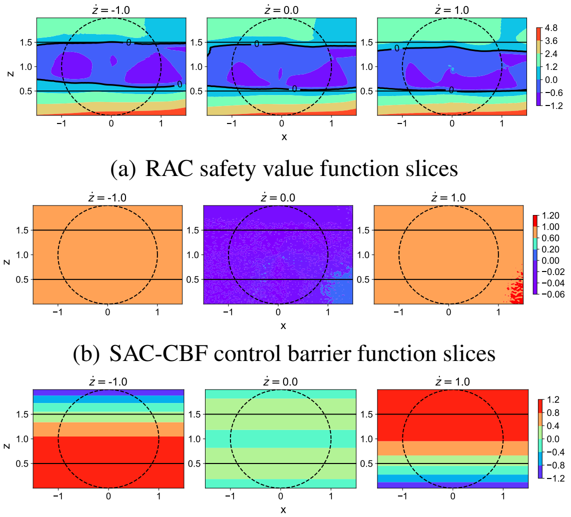


Figure 5. Trajectories of final policies trained by different algorithms in a run.

the upper bound and move horizontally in the safe set, corresponding to the feasible set in Figure 6(c). The other two algorithms reach higher returns at the cost of unacceptable constraint violation. In this task, keeping z between $[0.5, 1.5]$ will definitely bring tracking error, which leads to a lower return. Trajectories in Figure 5 indicate that RAC keeps the quadrotor in the safe set strictly and tracks the trajectory accurately inside the safe set while others fail to.



(a) RAC safety value function slices

(b) SAC-CBF control barrier function slices

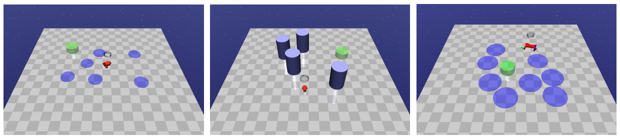
(c) SAC-SI safety index slices

Figure 6. The learned feasible set slices on the 2D xz-plane with different $\dot{z}$. Values below zero mean that the states are feasible.

Figure 6 shows the slices on xz-plane of the feasible sets learned by RAC, SAC-CBF, and SAC-SI with $\dot{z} = -1, 0, 1$. The sub-zero level set of each constrained function represents the learned feasible sets, i.e. $\{s \mid F^{\pi}(s, a; \phi) \leq 0\}$ where $F = Q_h, B$, or φ. In Figure 6(a), the feasible sets of RAC (inside the zero-contour) is smaller than $\mathcal{S}_c$ because when the quadrotor at the boundary of $\mathcal{S}_c$ with a velocity pointing outside, it is doomed to fly out of the space, leading to constraint violation. Thus, such states are supposed to be potentially dangerous and must have a super-zero safety value. Characterizing the feasible set helps RAC track the trajectory accurately inside the safe set (*optimality*) and satisfy the constraint strictly (*safety*). In contrast, an energy function like CBF or SI relies on prior knowledge about the dynamical system. When we choose empirical hyperparameters, the algorithms will possibly learn the wrong feasible set, which leads to either constraint violation or poor performance. As shown in Figure 6(b) and 6(c), when $\dot{z} \neq 0$, the whole safe set is considered unsafe because of conservativeness. SAC-CBF considers $\dot{z} = 0$ as safe, leading to a horizontal-moving policy while SAC-SI leans a wrong feasible set when $\dot{z} = 0$, leading to its poor performance.

6.3. Safety-Gym: Moving with Sensor Inputs

All previous tasks have full knowledge of the exact system states in the observations. In this section, we demonstrate the effectiveness of RCRL in complex safe control tasks with only high-dimensional sensor inputs, such as Lidar, on **Safety-Gym**. **Safety-Gym** is a widely used safe RL/CRL benchmark. In Figure 7, `point`, `car` or `doggo` agents (red) are controlled to reach `goal` (green) while avoiding `hazards` (blue, non-physical) or `pillars` (purple, physical). The tasks are named as {`Agent`}-{`Obstacles`}. The first two tasks have 76D observations space and the last task has 112D observation space and 12D action space.



(a) Car-Hazards (b) Point-Pillars (c) Doggo-Hazards

Figure 7. Safety-Gym tasks.

Baselines. We name the *on-policy* version of RCRL as Reachability Constrained Optimization **(RCO)**. Baseline algorithms in Safety-Gym tasks include traditional CRL baseline **PPO-Lagrangian** (Schulman et al., 2017; Achiam & Amodei, 2019) and two constrained version of PPO with energy-function based constraints (named as **PPO-CBF** and **PPO-SI**), and unconstrained baseline **PPO**. The distance and relative speed are approximated from the Lidar and speedometer sensors. For multiple obstacles, we select the one with the closest distance (in RCO) or the safety energy decreases (in PPO-SI and PPO-CBF) to compute the safety value functions or energy functions used for constraints.

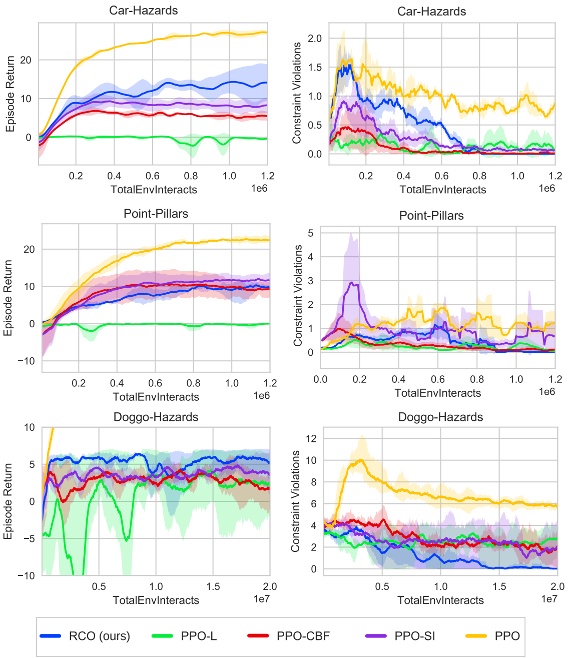


Figure 8. Training results of Safety-Gym tasks.

The training results are shown in Figure 8. The violation rate figures indicate that RCO has the best constraint satisfaction performance, while energy-based baselines PPO-CBF, PPO-SI oscillate around zero violation, indicating RCO learns the feasible sets more exactly. On the contrary, the prior parameters of SI and CBF fail to characterize the exact feasible sets. Meanwhile, RCO has comparable or better return performance, further verifying that RCO reduces the conservativeness compared to energy-based methods.

7. Concluding Remarks

We study the novel reachability constraint in CRL, where the safety value function is constrained, guaranteeing persistent constraint-satisfaction. We establish the self-consistency condition of the safety value function, which enables us to characterize the largest feasible set in CRL and consider performance optimality besides safety, avoiding the safety-oriented policies in prior HJ reachability analysis methods. We also prove the convergence of the proposed RCRL with multi-time scale stochastic approximation theory under mild assumptions. Experiments on common benchmarks indicate that RCRL is able to capture the approximate feasible sets, which further guarantees the persistent safety and the competitive performance w.r.t. baselines.

Although empirical results show that RCRL generates persistently safe agents after the convergence of training, like many other Lagrangian-based methods such as (Tessler et al., 2019; Ma et al., 2021b), RCRL focuses on safety *after convergence* rather than that *during learning*, while the latter is significant for real-world application of RL algorithms. Furthermore, RCRL explores the boundary of the feasible sets instead of conservatively staying inside them, leading to more violations during the early training stage, which can be seen in Figure 4 and 8. We are working on improving RCRL with different approaches such as model-based methods for *safer* exploration and learning.

Acknowledgements

The authors would like to thank Dr. Jingliang Duan and Wenjun Zou for their valuable suggestions about the problem formulation and the writing of this paper; anonymous reviewers and meta-reviewers for their insightful comments. This work was supported in part by NSF China, under Grant No. U20A201622, U20A20334 and 52072213. This work was also supported by the Ministry of Science and Technology of the People's Republic of China, the 2030 Innovation Megaprojects "Program on New Generation Artificial Intelligence" (Grant No. 2021AAA0150000). The authors want to thank support from the Tsinghua University-Toyota Joint Research Center for AI Technology of Automated Vehicle and Horizon Robotics.

References

Achiam, J. and Amodei, D. Benchmarking safe exploration in deep reinforcement learning. 2019.

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.

Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, pp. 3420–3431. IEEE, 2019.

Asayesh, S., Chen, M., Mehrandezh, M., and Gupta, K. Toward observation based least restrictive collision avoidance using deep meta reinforcement learning. *IEEE Robotics and Automation Letters*, 6(4):7445–7452, 2021. doi: 10.1109/LRA.2021.3098332.

Bansal, S. and Tomlin, C. J. Deepreach: A deep learning approach to high-dimensional reachability. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1817–1824. IEEE, 2021.

Bansal, S., Chen, M., Herbert, S., and Tomlin, C. J. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2242–2253. IEEE, 2017.

Bertsekas, D. P. *Nonlinear programming*. Athena Scientific, 2016.

Bharadhwaj, H., Kumar, A., Rhinehart, N., Levine, S., Shkurti, F., and Garg, A. Conservative safety critics for exploration. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=iaO86DUuKi.

Borkar, V. S. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., and Schoellig, A. P. Safe learning in robotics: From learning-based control to safe reinforcement learning, 2021.

Chen, B., Francis, J., Oh, J., Nyberg, E., and Herbert, S. L. Safe Autonomous Racing via Approximate Reachability on Ego-vision, November 2021. URL http://arxiv.org/abs/2110.07699. arXiv: 2110.07699.

Chen, Y. and Ames, A. D. Duality between density function and value function with applications in constrained optimal control and markov decision process. *arXiv preprint arXiv:1902.09583*, 2019.

Choi, J. J., Lee, D., Sreenath, K., Tomlin, C. J., and Herbert, S. L. Robust control barrier-value functions for safety-critical control. *arXiv preprint arXiv:2104.02808*, 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.

Dawson, C., Qin, Z., Gao, S., and Fan, C. Safe nonlinear control using robust neural lyapunov-barrier functions. *arXiv preprint arXiv:2109.06697*, 2021.

Duan, J., Guan, Y., Li, S. E., Ren, Y., Sun, Q., and Cheng, B. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021a. doi: 10.1109/TNNLS.2021.3082568.

Duan, J., Liu, Z., Li, S. E., Sun, Q., Jia, Z., and Cheng, B. Adaptive dynamic programming for nonaffine nonlinear optimal control problem with state constraints. *Neurocomputing*, 2021b. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.04.134. URL https://www.sciencedirect.com/science/article/pii/S0925231221015848.

Fisac, J. F., Lugovoy, N. F., Rubies-Royo, V., Ghosh, S., and Tomlin, C. J. Bridging hamilton-jacobi safety analysis and reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8550–8556, 2019. doi: 10.1109/ICRA.2019.8794107.

Fujimoto, S., van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Guan, Y., Duan, J., Li, S. E., Li, J., Chen, J., and Cheng, B. Mixed policy gradient. *arXiv preprint arXiv:2102.11513*, 2021.

Guan, Y., Ren, Y., Sun, Q., Li, S. E., Ma, H., Duan, J., Dai, Y., and Cheng, B. Integrated decision and control: Toward interpretable and computationally efficient driving intelligence. *IEEE Transactions on Cybernetics*, pp. 1–15, 2022. doi: 10.1109/TCYB.2022.3163816.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 10–15 Jul 2018.

Hsu, K.-C., Rubies-Royo, V., Tomlin, C. J., and Fisac, J. F. Safety and liveness guarantees through reach-avoid reinforcement learning. *arXiv preprint arXiv:2112.12288*, 2021.

Khalil, H. K. and Grizzle, J. *Nonlinear systems*. Prentice hall Upper Saddle River, 2002.

Kim, H., Papamakarios, G., and Mnih, A. The lipschitz constant of self-attention. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5562–5571. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kim21i.html.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. In *ICLR (Poster)*, 2016. URL http://arxiv.org/abs/1509.02971.

Liu, C. and Tomizuka, M. Control in a safe set: Addressing safety in human-robot interactions. In *Dynamic Systems and Control Conference*, volume 46209, pp.

V003T42A003. American Society of Mechanical Engineers, 2014.

Luo, Y. and Ma, T. Learning barrier certificates: Towards safe reinforcement learning with zero training-time violations. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 25621–25632. Curran Associates, Inc., 2021. URL https://openreview.net/forum?id=K4Su8BIivap.

Lygeros, J., Tomlin, C., and Sastry, S. Controllers for reachability specifications for hybrid systems. *Automatica*, 35 (3):349–370, 1999.

Ma, H., Chen, J., Li, S., Lin, Z., Guan, Y., Ren, Y., and Zheng, S. Model-based constrained reinforcement learning using generalized control barrier function. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4552–4559, 2021a. doi: 10.1109/IROS51168.2021.9636468.

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*, 2021b.

Ma, H., Liu, C., Li, S. E., Zheng, S., and Chen, J. Joint synthesis of safety certificate and safe control policy using constrained reinforcement learning. In Firoozi, R., Mehr, N., Yel, E., Antonova, R., Bohg, J., Schwager, M., and Kochenderfer, M. (eds.), *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168 of *Proceedings of Machine Learning Research*, pp. 97–109. PMLR, 23–24 Jun 2022.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

Mitchell, I. The flexible, extensible and efficient toolbox of level set methods. *Journal of Scientific Computing*, 35: 300–329, 2008. doi: 10.1007/s10915-007-9174-4.

Mitchell, I., Bayen, A., and Tomlin, C. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005. doi: 10.1109/TAC.2005.851439.

Qin, Z., Chen, Y., and Fan, C. Density constrained reinforcement learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8682–8692. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/qin21a.html.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *International conference on machine learning*, pp. 387–395. PMLR, 2014.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkfrvsA9FX.

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. doi: 10.1109/LRA.2021.3070252.

Thomas, G., Luo, Y., and Ma, T. Safe reinforcement learning by imagining the near future. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 13859–13869. Curran Associates, Inc., 2021.

Yang, T.-Y., Rosca, J., Narasimhan, K., and Ramadge, P. J. Projection-based constrained policy optimization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rke3TJrtPS.

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, J., He, T., Sra, S., and Jadbabaie, A. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJgnXpVYwS.

Zhao, W., He, T., and Liu, C. Model-free safe control for zero-violation reinforcement learning. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=UGp6FDaxB0f.

A. Loss Function and Gradients Derivation

The Q-value loss is the mean square error between the approximated Q function and its target (Lillicrap et al., 2016):

$$\mathcal{J}_Q(\omega) = \mathbb{E}_{s\sim\mathcal{D}}[1/2(Q(s,a;\omega) - \hat{Q}(s,a))^2] \tag{13}$$

where

$$\hat{Q}(s,a) = r(s,a) + \gamma \mathbb{E}_{s'\sim P}[Q_{(}s', \pi(s');\omega)].$$

Therefore, we are able to derive the stochastic gradients of each objective w.r.t. the parameter vectors in the approximate function. A sample (s_t, a_t, s_{t+1}) at time step t is leveraged to compute the stochastic gradients. First, (13) and (8) can be optimized with stochastic gradients descent (SGD):

$$\begin{aligned} \hat{\nabla}_\omega \mathcal{J}_Q(\omega) =& \nabla_\omega Q(s_t, a_t;\omega) \cdot \left[Q_\omega(s_t,a_t) - (r(s_t,a_t) + \gamma Q(s_{t+1}, a_{t+1};\omega))\right], \\ \hat{\nabla}_\phi \mathcal{J}_{Q_h}(\phi) =& \nabla_\phi Q_h(s_t,a_t;\phi) \cdot \left[Q_h(s_t,a_t;\phi) - ((1-\gamma)h(s) + \gamma \max\{h(s), Q_h(s_{t+1},a_{t+1};\phi)\})\right], \end{aligned} \tag{14}$$

where $a_{t+1} = \pi(s_{t+1})$.

Combining results from (Silver et al., 2014; Lillicrap et al., 2016), the estimated deterministic policy gradient (DPG) is

$$\hat{\nabla}_\theta \mathcal{J}_\pi(\theta) = \nabla_a \left[-Q_\omega(s_t,a_t) + \lambda_\xi(s_t) Q_h(s_t,a_t;\phi)\right]|_{a=a_t} \cdot \nabla_\theta \pi_\theta(s_t). \tag{15}$$

Then the stochastic gradient of the multiplier is used to ascend (10):

$$\hat{\nabla}_\xi \mathcal{J}_\lambda(\xi) = Q_h(s_t, \pi_\theta(s_t);\phi) \nabla_\xi \lambda_\xi(s_t). \tag{16}$$

B. Proofs

B.1. Proof of Theorem 3.5

We only prove the self-consistency condition because the SBE can be proven similarly. From the definition of the safety value function, we know that

$$\begin{aligned} V_h^\pi(s) &= \max_{t\in\mathbb{N}} h(s_t^\pi \mid s_0 = s) \\ &= \max\{h(s), \max_{t\in\mathbb{N}^+} h(s_t^\pi | s_0 = s)\} \\ &= \max\{h(s), \max_{t\in\mathbb{N}^+} h(s_t^\pi | s_1 = s')\} \\ &= \max\{h(s), \max_{t\in\mathbb{N}} h(s_t^\pi | s_0 = s')\} \\ &= \max\{h(s), V_h^\pi(s')\}, \end{aligned} \tag{17}$$

where $s' \sim P(\cdot \mid s, \pi(s))$.

B.2. Proof of Proposition 4.1

It is obvious that $\mathcal{S}_f^\pi \subseteq \mathcal{S}_f$, we only need to prove that $\mathcal{S}_f \subseteq \mathcal{S}_f^\pi$. When $\mathcal{S}_f \subseteq \mathcal{S}_0$, we have $\mathcal{S}_f \cap \mathcal{S}_0 = \mathcal{S}_f$. Therefore, the constraint in (RCRL) becomes

$$V_h^\pi(s) \leq 0, \forall s \in \mathcal{S}_f. \tag{18}$$

Thus we have $s \in \mathcal{S}_f^\pi$ by Definition 3.2. In other words, we can conclude that if $s \in \mathcal{S}_f$, we will get $s \in \mathcal{S}_f^\pi$, which means $\mathcal{S}_f \subseteq \mathcal{S}_f^\pi$.

B.3. Proof of Proposition 4.2

We start from decomposing the surrogate Lagrangian (6):

$$
\begin{aligned}
&\min_{\pi}\max_{\lambda}\hat{\mathcal{L}}(\pi,\lambda)\\
=&\min_{\pi}\max_{\lambda}\mathbb{E}_{s\sim d_0}[-V^{\pi}(s)+\lambda(s)V_h^{\pi}(s)]\\
=&\min_{\pi}\max_{\lambda}\left\{\mathbb{E}_{s\sim d_0}\left[(-V^{\pi}(s)+\lambda(s)V_h^{\pi}(s))\cdot\mathbb{1}_{s\in\mathcal{S}_f}\right]+\mathbb{E}_{s\sim d_0}\left[(-V^{\pi}(s)+\lambda(s)V_h^{\pi}(s))\cdot\mathbb{1}_{s\notin\mathcal{S}_f}\right]\right\}\\
=&\min_{\pi}\max_{\lambda}\mathbb{E}_{s\sim d_0}\left[(-V^{\pi}(s)+\lambda(s)V_h^{\pi}(s))\cdot\mathbb{1}_{s\in\mathcal{S}_f}\right]\\
&\qquad+\min_{\pi}\max_{\lambda}\mathbb{E}_{s\sim d_0}\left[(-V^{\pi}(s)+\lambda(s)\underbrace{V_h^{\pi}(s)}_{>0})\cdot\mathbb{1}_{s\notin\mathcal{S}_f}\right]\\
=&\underbrace{\min_{\pi}\max_{\lambda}\mathbb{E}_{s\sim d_0}\left[(-V^{\pi}(s)+\lambda(s)V_h^{\pi}(s))\cdot\mathbb{1}_{s\in\mathcal{S}_f}\right]}_{\text{Part 1}}\\
&\qquad+\underbrace{\min_{\pi}\mathbb{E}_{s\sim d_0}\left[(-V^{\pi}(s)+\lambda_{\max}V_h^{\pi}(s))\cdot\mathbb{1}_{s\notin\mathcal{S}_f}\right]}_{\text{Part 2}}.
\end{aligned}
\tag{19}
$$

Note that from line 3 to line 4 the $\min\max$ can be decomposed into two parts because the policy (or the multiplier) is statewise and the results $\pi(s)$ (or $\lambda(s)$) of states inside and outside $\mathcal{S}_f$ are independent.

One the one hand, when $\lambda_{\max}\to+\infty$, Part 2 will be dominated by $V_h^{\pi}(s)>0$. Thus, $\lim_{\lambda_{\max}\to+\infty}\hat{\pi}^*$ tries to minimize the expected safety value function of initial states outside the largest feasible set. On the other hand, Part 1 is to find the saddle point of the lagrangian of the constrained optimization problem: maximizing the expected return while satisfying the reachability constraint for all initial states inside $\mathcal{S}_f$. In other words,

$$
\min_{\pi}\max_{\lambda}\mathbb{E}_{s\sim d_0}\left[(-V^{\pi}(s)+\lambda(s)V_h^{\pi}(s))\cdot\mathbb{1}_{s\in\mathcal{S}_f}\right]\iff\begin{array}{rl}\max_{\pi} & \mathbb{E}_{s\sim d_0(s)}[V^{\pi}(s)\cdot\mathbb{1}_{s\in\mathcal{S}_f}]\\ \text{subject to} & V_h^{\pi}(s)\le 0,\forall s\in\mathcal{S}_f\cap\mathcal{S}_0\end{array}.
$$

Overall, $\lim_{\lambda_{\max}\to+\infty}\hat{\pi}^*$ aims to (1) maximize the expected return while satisfy reachability constraints for initial states inside $\mathcal{S}_f$ and (2) minimize the safety value function of initial states outside $\mathcal{S}_f$. This is exactly what problem (RCRL) does. Therefore, we have $\lim_{\lambda_{\max}\to+\infty}\hat{\pi}^*=\pi^*$.

B.4. Proof of Theorem 5.5

The proof borrows heavily from (Chow et al., 2017) and (Ma et al., 2022) which both follow the convergence proof of multi-time scale stochastic approximation algorithms in (Borkar, 2009). A high level overview of the proof structure is shown as follows.

1. By utilizing policy evaluation techniques, we show the critic and safety value function update converge (in the fastest time scale) almost surely to a fixed point solution (ω^*,ϕ^*).
2. Then, with convergence properties of multi-time scale discrete stochastic approximation algorithms, we show that each update (θ_k,ξ_k) converges almost surely to a stationary point (θ^*,ξ^*) of the corresponding continuous time system but with different speeds.
3. By using Lyapunov analysis, we show that the continuous time system is locally asymptotically stable at the stationary point (θ^*,ξ^*).
4. Since the Lyapunov function used in the above analysis is the Lagrangian function $\mathcal{L}(\theta,\xi)$, we conclude that the stationary point (θ^*,ξ^*) is a local saddle point. Finally by the local saddle point theorem, we deduce that θ^* is a locally optimal solution for the reachability constrained RL problem.

Time scale 1 (Convergence of ω- and ϕ-updates) The step size Assumption 5.4 tells that $\{\omega_k\}$ and $\{\phi_k\}$ converges on a faster time scale than $\{\theta_k\}$ and $\{\xi_k\}$. According to (Borkar, 2009, Lemma 1, Chapter 6), we can treat (θ,ξ) as arbitrarily fixed quantities during updating $\{\omega_k\}$ and $\{\phi_k\}$. Therefore, we take $(\theta, \xi) = (\theta_k, \xi_k)$, which means the policy and the multiplier are fixed and we are performing policy evaluation to compute $Q^{\pi_{\theta_k}}(s,a)$ and $Q_h^{\pi_{\theta_k}}(s,a)$. With the standard policy evaluation convergence results in (Sutton & Barto, 2018), one can easily know that $Q(s,a;\omega_k) \to Q(s,a;\omega^*) = Q^{\pi_{\theta_k}}(s,a)$ as $k \to \infty$ because the operator $\mathcal{B}$ defined by

$$\mathcal{B}[Q(s,a)] = r(s,a) + \gamma \mathbb{E}_{s' \sim P}[Q(s', \pi(s'))]$$

is a γ contraction mapping. Thus, all we need to do is to prove that the safety value evaluation in (9) is a γ-contraction mapping as well, which is stated in the following Lemma.

Lemma B.1 (γ-contraction Mapping)**.** *Under Assumption 5.1, the operator $\mathcal{B}_h$ introduced by $\mathcal{B}_h[Q_h(s,a)] = (1-\gamma)h(s) + \gamma \max\{h(s), \mathbb{E}_{s'\sim P}[Q_h(s', \pi(s'))]\}$ is a γ-contraction mapping.*

Proof. We study the supremum norm of $\mathcal{B}_h$. For any Q_h and $\hat{Q}_h$, the following holds:

$$\begin{aligned}
\|\mathcal{B}_h[Q_h(s,a)] - \mathcal{B}_h[\hat{Q}_h(s,a)]\|_\infty &= \gamma \| \max\{h(s), \mathbb{E}_{s'\sim P}[Q_h(s', \pi(s'))]\} - \max\{h(s), \mathbb{E}_{s'\sim P}[\hat{Q}_h(s', \pi(s'))]\}\|_\infty \\
&\leq \gamma \|\mathbb{E}_{s'\sim P}[Q_h(s', \pi(s'))] - \mathbb{E}_{s'\sim P}[\hat{Q}_h(s', \pi(s'))]\|_\infty \\
&= \gamma \|\mathbb{E}_{s'\sim P}[Q_h(s', \pi(s')) - \hat{Q}_h(s', \pi(s'))]\|_\infty \\
&\leq \gamma \|Q_h(s,a) - \hat{Q}_h(s,a)\|_\infty .
\end{aligned}$$

□

According to (Bertsekas, 2016, Proposition A.26), we can conclude that $Q_h(s,a;\phi_k)$ will converge to $Q_h(s,a;\phi^*) = Q_h^{\pi_{\theta_k}}(s,a)$ as $k \to \infty$. Hence, both ω_k and ϕ_k converge to ω^* and ϕ^*, respectively and the convergence of **Time scale 1** is proved.

Time scale 2 (Convergence of θ-update) Due to the faster convergence speed of θ_k than ξ_k, we can take $\xi = \xi_k$ when updating θ according to (Borkar, 2009, Lemma. 1, Chapter 6). Furthermore, since ω and ϕ converge on a faster speed than θ, we have $\|Q(s,a;\omega_k) - Q^{\pi_{\theta_k}}(s,a)\| \to 0$ and $\|Q_h(s,a;\phi_k) - Q_h^{\pi_{\theta_k}}(s,a)\| \to 0$ almost surely. Assume that the sample distribution is the same as $\mathcal{D}$. The θ-update from (15) is

$$\theta_{k+1} = \Gamma_\Theta \left[\theta_k - \beta_2(k) \nabla_a (-Q_{\omega_k}(s_t, a) + \lambda_{\xi_k}(s_t) Q_h(s_t, a; \phi_k))|_{a=a_t} \cdot \nabla_\theta \pi_\theta(s_t)|_{\theta=\theta_k}\right]. \tag{20}$$

(20) can also be rewritten as:

$$\theta_{k+1} = \Gamma_\Theta \left[\theta_k + \beta_2(k)(-\nabla_\theta \mathcal{L}(\theta, \xi)|_{\theta=\theta_k} + \delta\theta_{k+1} + \delta\theta_\epsilon)\right].$$

where

$$\begin{aligned}
\delta\theta_{k+1} =& \mathbb{E}_{s\sim\mathcal{D}} \left[\nabla_a(-Q_{\omega_k}(s,a) + \lambda(s;\xi_k) Q_h(s,a;\phi_k))|_{a=\pi(s;\theta_k)} \nabla_\theta \pi(s;\theta)|_{\theta=\theta_k}\right] \\
&- \nabla_a(-Q_{\omega_k}(s_t,a) + \lambda_{\xi_k}(s_t) Q_h(s_t,a;\phi_k))|_{a=a_t} \cdot \nabla_\theta \pi(s_t;\theta)|_{\theta=\theta_k}
\end{aligned}$$

and

$$\begin{aligned}
\delta\theta_\epsilon = \mathbb{E}_{s\sim\mathcal{D}}[&-\nabla_a(-Q(s,a;\omega_k) + \lambda(s;\xi_k) Q_h(s,a;\phi_k))|_{a=\pi(s;\theta_k)} \nabla_\theta \pi(s;\theta)|_{\theta=\theta_k} \\
&+ \nabla_a(-Q^{\pi_{\theta_k}}(s,a) + \lambda(s;\xi_k) Q_h^{\pi_{\theta_k}}(s,a))|_{a=\pi(s;\theta_k)} \nabla_\theta \pi(s;\theta)|_{\theta=\theta_k}]
\end{aligned}$$

1. We will show that $\delta\theta_{k+1}$ is square integrable first, specifically,

   $$\mathbb{E}[\|\delta\theta_{k+1}\|^2 \mid \mathcal{F}_{\theta,k}] \leq 4 \cdot \left[\|\nabla_\theta \pi_\theta(s)|_{\theta=\theta_k}\|_\infty^2 \times (\|\nabla_a Q(s,a;\omega_k)\|_\infty^2 + \|\lambda(s;\xi_k)\|_\infty^2 \cdot \|\nabla_a Q_h(s,a;\phi_k)\|_\infty^2)\right].$$

   Assumption 5.3 implies that

   $$\begin{aligned}
   \|\nabla_\theta \pi_\theta(s)|_{\theta=\theta_k}\|_\infty^2 &\leq K_1(1 + \|\theta_k\|_\infty^2), \\
   \|\nabla_a Q(s,a;\omega_k)\|_\infty^2 &\leq K_2(1 + \max_{a\in\mathcal{A}} \|a\|_\infty^2), \\
   \|\nabla_a Q_h(s,a;\phi_k)\|_\infty^2 &\leq K_3(1 + \max_{a\in\mathcal{A}} \|a\|_\infty^2).
   \end{aligned}$$

   where K_1, K_2, K_3 is three sufficiently large scalars. Furthermore, $\lambda(s;\xi_k)$ can be bounded by $\lambda_{\max}$ due to the multiplier upper bound. Because of the aforementioned conditions, we can conclude that $\mathbb{E}[\|\delta\theta_{k+1}\|^2 \mid \mathcal{F}_{\theta,k}] \leq 4K_1(1 + \|\theta_k\|_\infty^2)[K_2(1 + \max_{a\in\mathcal{A}} \|a\|_\infty^2) + \lambda_{\max}(K_3(1 + \max_{a\in\mathcal{A}} \|a\|_\infty^2)] < \infty$. Thus $\delta\theta_{k+1}$ is square integrable.

2. Second, we will show $\delta\theta_\epsilon \to 0$. Specifically,

$$\begin{aligned}\delta\theta_\epsilon &= \mathbb{E}_{s\sim\mathcal{D}}[\nabla_a(-Q^{\pi_{\theta_k}}(s,a) + Q(s,a;\omega_k) + \lambda(s;\xi_k)(Q_h^{\pi_{\theta_k}}(s,a) - Q_h(s,a;\phi_k)))|_{a=\pi(s;\theta_k)}\nabla_\theta\pi(s;\theta)|_{\theta=\theta_k}] \\ &= \mathbb{E}_{s\sim\mathcal{D}}[\nabla_a(-Q(s,a;\omega^*) + Q(s,a;\omega_k) + \lambda(s;\xi_k)(Q_h(s,a;\phi^*) - Q_h(s,a;\phi_k)))|_{a=\pi(s;\theta_k)}\nabla_\theta\pi(s;\theta)|_{\theta=\theta_k}] \\ &\leq \mathbb{E}_{s\sim\mathcal{D}}[\nabla_\theta\pi(s;\theta)|_{\theta=\theta_k}]\cdot(K_4\|\omega_k-\omega^*\|_\infty + \lambda_{max}K_5\|\phi_k-\phi^*\|_\infty) \to 0\end{aligned}$$

   where K_4, K_5 is the Lipschitz constant. The limit comes from the convergence of the parameters in **Time scale 1**.

3. Because $\hat{\nabla}_\theta\mathcal{J}_\pi(\theta)|_{\theta=\theta_k}$ is a sample of $\nabla_\theta\mathcal{L}(\theta,\xi)|_{\theta=\theta_k}$, we can conclude that $\mathbb{E}[\delta\theta_{k+1} \mid \mathcal{F}_{\theta,k}] = 0$, where $\mathcal{F}_{\theta,k} = \sigma(\theta_m, \delta\theta_m, m \leq k)$ is the filtration generated by different independent trajectories (Chow et al., 2017).

Based on the three facts, the θ-update given by (20) is a stochastic approximation of the continuous system $\theta(t)$ (Borkar, 2009), described by an ODE:

$$\dot{\theta} = \Upsilon_\theta[-\nabla_\theta\mathcal{L}(\theta,\lambda)], \tag{21}$$

where

$$\Upsilon_\theta[F(\theta)] \triangleq \lim_{\eta\to0^+}\frac{\Gamma_\Theta(\theta+\eta F(\theta)) - \Gamma_\Theta(\theta)}{\eta}$$

is the left directional derivative of the function $\Gamma_\Theta(\theta)$ in the direction of $F(\theta)$. The purpose of the directional derivative is to guarantee the update $\Upsilon_\theta[-\nabla_\theta\mathcal{L}(\theta,\lambda)]$ will point in the descent direction along the boundary of Θ when the θ-update hits the boundary. Invoking the Step 2 in (Chow et al., 2017, Appendix A.2), one can know that $d\mathcal{L}(\theta,\xi)/dt = -\nabla_\theta\mathcal{L}(\theta,\lambda)^T \cdot \Upsilon_\theta[-\nabla_\theta\mathcal{L}(\theta,\lambda)] \leq 0$ and the value will be non-zero if $\|\Upsilon_\theta[-\nabla_\theta\mathcal{L}(\theta,\lambda)]\| \neq 0$.

Let us consider the continuous system. For a given ξ, define a Lyapunov function

$$L_\xi(\theta) = \mathcal{L}(\theta,\xi) - \mathcal{L}(\theta^*,\xi)$$

where θ^* is a local minimum point. Therefore, there exists a scalar r such that $\forall\theta \in B_r(\theta^*) = \{\theta \mid \|\theta-\theta^*\| \leq r\}$, $L_\xi(\theta) \geq 0$. Moreover, according to (Bertsekas, 2016, Proposition 1.1.1), we obtain $\Upsilon_\theta[-\nabla_\theta\mathcal{L}(\theta,\lambda)]|_{\theta=\theta^*} = 0$, which means θ^* is a stationary point. Due to the non-positive property of $d\mathcal{L}(\theta,\xi)/dt$ and refer to the (Khalil & Grizzle, 2002, Chapter 4), aforementioned contents show that for any given initial condition $\theta \in B_r(\theta^*)$, the continuous trajectory of $\theta(t)$ of (21) converges to θ^*, i.e. $\mathcal{L}(\theta^*,\xi) \leq \mathcal{L}(\theta(t),\xi) \leq \mathcal{L}(\theta(0),\xi)$ for $\forall t \geq 0$.

Finally, because of the following properties:

1. From (Chow et al., 2017, Proposition 17), $\nabla_\theta\mathcal{L}(\theta,\xi)$ is a Lipschitz function in θ;

2. The step size schedules follow Assumption 5.4;

3. $\delta\theta_{k+1}$ is a square Martingale difference sequence and $\delta\theta\epsilon$ is a vanishing error;

4. $\theta_k \in \Theta$, which implies that $\sup_k \|\theta_k\| < \infty$ almost surely,

one can invoke (Borkar, 2009, Theorem 2, Chapter 6) (multi-time scale stochastic approximation theory) to know that the sequence $\{\theta_k\}, \theta_k \in \Theta$ converges almost surely to the solution of (21), which further converges almost surely to the locally minimum point θ^*.

Time scale 3 (Convergence of ξ-update) The parameter ξ of multiplier is on the slowest time scale so we can assume that during the ξ-update, the policy has converged to the local minimum point, i.e. $\|\theta_k - \theta^*(\xi_k)\| = 0$ and the safety value has converged to a fixed quantity such that $\|Q_h(s,a;\phi_k) - Q_h^{\pi_{\theta_k}}(s,a)\| = 0$. With the continuity of $\nabla_\xi\mathcal{L}(\theta,\xi)$, we have $\|\nabla_\xi\mathcal{L}(\theta,\xi)|_{\theta=\theta_k,\xi=\xi_k} - \nabla_\xi\mathcal{L}(\theta,\xi)|_{\theta=\theta^*(\xi_k),\xi=\xi_k}\| = 0$ almost surely. Thus, the ξ-update can be expressed as:

$$\begin{aligned}\xi_{k+1} &= \Gamma_\Xi\left(\xi_k + \beta_3(k)Q_h(s_t,\pi_{\theta_k}(s_t);\phi_k)\nabla_\xi\lambda(s_t)|_{\xi=\xi_k}\right) \\ &= \Gamma_\Xi\left(\xi_k + \beta_3(k)(\nabla_\xi\mathcal{L}(\theta,\xi)|_{\theta=\theta^*(\xi_k),\xi=\xi_k} + \delta\xi_{k+1})\right)\end{aligned} \tag{22}$$

where

$$\begin{aligned}\delta\xi_{k+1} &= -\nabla_\xi \mathcal{L}(\theta,\xi)|_{\theta=\theta^*(\xi_k),\xi=\xi_k} + Q_h(s_t,\pi_{\theta_k}(s_t);\phi_k)\nabla_\xi\lambda(s_t)|_{\xi=\xi_k} \\ &= -\mathbb{E}_{s\sim\mathcal{D}}[Q_h^{\pi_{\theta^*}}(s,\pi_{\theta^*}(s))\nabla_\xi\lambda(s;\xi)|_{\xi=\xi_k}] + Q_h(s_t,\pi_{\theta_k}(s_t);\phi_k)\nabla_\xi\lambda(s_t)|_{\xi=\xi_k} \\ &= -\mathbb{E}_{s\sim\mathcal{D}}[Q_h^{\pi_{\theta^*}}(s,\pi_{\theta^*}(s))\nabla_\xi\lambda(s;\xi)|_{\xi=\xi_k}] + \\ &\qquad (Q_h(s_t,\pi_{\theta_k}(s_t);\phi_k) - Q_h^{\pi_{\theta_k}}(s_t,\pi_\theta(s_t)) + Q_h^{\pi_{\theta_k}}(s_t,\pi_\theta(s_t)))\nabla_\xi\lambda(s_t)|_{\xi=\xi_k}.\end{aligned}$$

Similar with θ-update, we need to prove the followings:

1. $\delta\xi_{k+1}$ is square integrable because

   $$\mathbb{E}[\|\delta\xi_{k+1}\|^2 \mid \mathcal{F}_{\xi,k}] \le 2 \times \max_{s\in\mathcal{S}} |h(s)|^2 \times K_6(1+\|\xi_k\|_\infty^2) < \infty$$

   where K_6 is a sufficiently large number.

2. Since $\|Q_h(s_t,\pi_\theta(s_t);\phi_k) - Q_h^{\pi_{\theta_k}}(s_t,\pi_\theta(s_t)\|_\infty \to 0$ and $Q_h^{\pi_{\theta_k}}(s_t,\pi_\theta(s_t))\nabla_\xi\lambda(s_t)|_{\xi=\xi_k}$ is sample of $\nabla_\xi\mathcal{L}(\theta,\xi)|_{\theta=\theta^*(\xi_k),\xi=\xi_k}$, one can conclude that $\mathbb{E}[\delta\xi_{k+1} \mid \mathcal{F}_{\xi,k}] = 0$ almost surely, where $\mathcal{F}_{\xi,k} = \sigma(\xi_m, \delta\xi_m, m \le k)$ is the filtration of ξ generated by different independent trajectories.

Therefore, the ξ-update is a stochastic approximation of the continuous system

$$\dot{\xi} = \Upsilon_\Xi[\nabla_\xi\mathcal{L}(\theta,\xi)|_{\theta=\theta^*(\xi)}] \tag{23}$$

with a Martingale difference error $\delta\xi_k$, where Υ_Ξ is the left directional derivative similarly defined in **Time scale 2**. Analogous to **Time scale 2** and in (Chow et al., 2017, Appendix B.2), $d\mathcal{L}(\theta^*(\xi),\xi)/dt = \nabla_\xi\mathcal{L}(\theta,\xi)|_{\theta=\theta^*(\xi)}{}^T \cdot \Upsilon_\Xi[\nabla_\xi\mathcal{L}(\theta,\xi)|_{\theta=\theta^*(\xi)}] \ge 0$, which is non-zero if $\|\Upsilon_\Xi[\nabla_\xi\mathcal{L}(\theta,\xi)|_{\theta=\theta^*(\xi)}]\| \ne 0$.

For a local maximum point ξ^*, define a Lyapunov function

$$L(\xi) = \mathcal{L}(\theta^*(\xi),\xi^*) - \mathcal{L}(\theta^*(\xi),\xi).$$

There exists a scalar r' such that for $\forall\xi \in B_{r'}(\xi^*) = \{\xi \in \Xi \mid \|\xi-\xi^*\| \le r'\}$, $L(\xi) \ge 0$. Moreover, $dL(\xi(t))/dt = -d\mathcal{L}(\theta^*(\xi),\xi)/dt \le 0$ and the equal sign only holds when $\Upsilon_\Xi[\nabla_\xi\mathcal{L}(\theta,\xi)|_{\theta=\theta^*(\xi)}] = 0$. This means ξ^* is a stationary point. One can invoke (Khalil & Grizzle, 2002, Chapter 4) and conclude that given any initial condition $\xi \in B_{r'}(\xi^*)$, the trajectory of (23) convergences to ξ^*, which is a locally maximum point.

Now with (1) $\{\xi_k\}$ is a stochastic approximation to $\xi(t)$ with a Martingale difference error; (2) the step size schedules in Assumption 5.4; (3) the convex and compact property in projection and (4) $\nabla_\xi\mathcal{L}(\theta^*(\xi),\xi)$ is a Lipschitz function in ξ. we can apply the multi-time scale stochastic approximation theory again and show that $\{\xi_k\}$ converges to a local maximum point ξ^* almost surely, i.e. $\mathcal{L}(\theta^*(\xi^*),\xi^*) \ge \mathcal{L}(\theta^*(\xi),\xi)$.

Local Saddle Point. From **Time scale 2** and **3** we know that $\mathcal{L}(\theta^*(\xi^*),\xi^*) \ge \mathcal{L}(\theta^*(\xi),\xi)$ and $\mathcal{L}(\theta^*,\xi) \le \mathcal{L}(\theta,\xi)$. Thus, $\mathcal{L}(\theta^*,\xi) \le \mathcal{L}(\theta^*,\xi^*) \le \mathcal{L}(\theta,\xi^*)$, which means (θ^*,ξ^*) is a local saddle point of $\mathcal{L}(\theta,\xi)$. With the saddle point theorem (Bertsekas, 2016, Proposition 5.1.6), we finally come to the conclusion that $\pi(\cdot;\theta^*)$ is a locally optimal policy to the RCRL problem (RCRL).

C. Implementation Details of Algorithms

C.1. The Gap between Assumptions and Practical Implementations

Finite MDP. The boundness of $\mathcal{S}$, A, and reward function can be guaranteed in common RL tasks. However, it is in most of the cases that $\mathcal{S}$ and $\mathcal{A}$ are continuous such that they are infinite. One can discretize the space to get a finite one at the cost of inaccuracy but we will keep the space continuous.

Parameterized approximation. A popular choice of function approximators is deep neural networks (NN) that is differentiable w.r.t. its parameters. However, the general conclusion about the continuity and Lipschitz constant of a NN is still an open problem (Kim et al., 2021). We still adopt NN in our experiments and leverage clipped gradient update (Zhang et al., 2020) as the projection operator to keep the parameters of NNs in compact sets as mentioned in Section 4.2. Moreover, a Lagrange multiplier network introduced by (Ma et al., 2021b) is used for statewise constraint-satisfaction.

Step sizes. Actually we cannot make any schedule of learning rates to make their sum goes to infinity due to a limited number of steps but the sum of the square of learning rates are finite. Furthermore, we utilize $\beta_1(k) > \beta_2(k) > \beta_3(k), \forall k$ to approximate the relationships among the learning rates.

Exploration issue for deterministic policies. Deterministic policies may lack exploration due to overestimation error (Lillicrap et al., 2016; Fujimoto et al., 2018) but this can be mitigated by off-policy updates with a replay buffer, where the learning and the exploration is treated independently. Hence, we construct a stochastic policy giving means and variances of a multivariate Gaussian distribution but only take the means during evaluation.

C.2. Off-policy Parts

Implementation details about off-policy RL algorithms compared in safe-control-gym are covered in this section. For fair comparison, all methods are implemented under the same code base, see (Guan et al., 2021). The only differences among them is the constrained function and some hyperparameters, which will be explained in detail in the following content.

C.2.1. Algorithms

RAC implementation is similar to common off-policy Lagrangian-based CRL methods but with a different constrained function, i.e. the safety value function. As shown in Algorithm 1, at each update step gradients of the critic, the safety value function, the actor and the multiplier are computed through samples collected from the environment. The actor is updated on an intermediate frequency and the multiplier at the slowest frequency, correspond to Assumption 5.4.

SAC-Lagrangian is a SAC-based implementation of RCPO (Tessler et al., 2019). The constraint imposed on the RL problem is $\mathcal{J}_c^\pi = \mathbb{E}_{s\sim\mathcal{D},a\sim\pi}[Q_c^\pi(s,a)] \leq \eta$, where $Q_c^\pi(s,a) = \sum_t \gamma^t c(s_t|s_0 = s, a_0 = a, \pi)$ and η is the constraint threshold. Bsecause the constraint is about expectation rather than statewise, the multiplier λ is a scalar here, but updated with dual ascent similarly with (10).

SAC-Reward Shaping is a SAC-based implementation of fixed penalty optimization (FPO) mentioned in (Achiam et al., 2017; Tessler et al., 2019). It only adds an additional term in the reward function to punish constrain violation, without any constrained optimization approaches. The reward function during training is modified into $r'(s,a) = r(s,a) - \rho h(s)$, where $\rho > 0$ is a fixed penalty coefficient. Then the networks are updated through standard RL and here SAC. Choosing an appropriate ρ is engineering-intuitive and sometimes the tuning process will be time-consuming.

SAC-CBF is inspired by CBF for safe control in control community (Dawson et al., 2021; Ma et al., 2021a; Choi et al., 2021). The core idea is to make potential unsafe behaviors smooth out exponentially as the agent approaches the safe boundary. The constrained function is called barrier function $B(s,a) \triangleq \dot{h}(s) + \mu h(s) \leq 0$ where $\mu \in (0,1)$ is a hyperparameter.

SAC-SI leverages a human-designed safe index (SI) as the energy function. The control policy needs to keep the system energy low ($\varphi \leq 0$) and dissipate the energy when the system is at high energy ($\varphi > 0$) (Ma et al., 2022). Hence, the constraint is $\Delta(s,a) \triangleq \varphi(s') - \max\{\varphi(x) - \eta_D, 0\} \leq 0$, where η_D is a slack variable controlling the decent rate of SI. A commonly used SI is in the form of $\varphi(s) = \sigma - (-h(s))^n + k\dot{h}(s)$ (Zhao et al., 2021), which is chosen in this paper.

C.2.2. Hyperparameters

Table 1 shows the hyperparameters of algorithms evaluated in safe-control-gym.

C.3. On-policy Parts

Implementation details about on-policy RL algorithms including the on-policy version of RCRL benchmarked in Safety-Gym are covered in this section. For fair comparison, all methods are implemented under the same code base, see (Achiam & Amodei, 2019).

C.3.1. Algorithms

RCO. The advantages function for reward value and safety value are denoted as A^π and A_h^π. Denote policy parameterization as π_θ, the loss function of RCO when policy parameters, $\theta = \theta_k$ is

$$\mathcal{J}(\theta,\xi) = \mathbb{E}_{s,a\sim\pi_{\theta_k}}\left\{\overline{A^{\pi_{\theta_k}}}(s,a) + \lambda_\xi(s)\overline{A_h^{\pi_{\theta_k}}}(s,a)\right\} \tag{24}$$

Table 1. Off-policy Algorithms Hyperparameters in safe-control-gym

Parameter	Value
Shared	
Optimizer	Adam ($\beta_1 = 0.99, \beta_2 = 0.999$)
Approximation function	Multi-layer Perceptron
Number of hidden layers	2
Number of neurons in a hidden layer	256
Nonlinearity of hidden layer	ELU
Nonlinearity of output layer of multiplier	Softplus
Critic/Constrained function learning rate	Linear annealing 1e-4 → 1e-6
Actor learning rate	Linear annealing 2e-5 → 1e-6
Temperature coefficient α learning rate	Linear annealing 8e-5 → 8e-6
Reward discount factor (γ)	0.99
Policy update interval (m_π)	4
Multiplier ascent interval (m_λ)	12
Target smoothing coefficient (τ)	0.005
Max episode length (N)	360
Expected Entropy ($\bar{\mathcal{H}}$)	-2
Replay buffer size	50,000
Replay batch size	512
RAC	
Multiplier learning rate	Linear annealing 6e-7 → 1e-7
SAC-Lagrangian	
Multiplier learning rate	3e-4
SAC-SI	
Multiplier learning rate	Linear annealing 1e-6 → 1e-7
σ, n, k	0.1, 2, 1
η_D	0.1
SAC-CBF	
Multiplier learning rate	Linear annealing 1e-6 → 1e-7
μ	0.1
SAC-Reward Shaping	
Critic learning rate	Linear annealing 3e-5 → 3e-6
Actor learning rate	Linear annealing 8e-5 → 8e-6
Policy update interval (m_π)	1
ρ	0.5

where

$$\overline{A^{\pi_{\theta_k}}}(s,a) = \min\left(\frac{\pi_\theta(a \mid s)}{\pi_{\theta_k}(a \mid s)} A^{\pi_{\theta_k}}(s,a), g\left(\epsilon, A^{\pi_{\theta_k}}(s,a)\right)\right),\ g(\epsilon, A) = \begin{cases}(1+\epsilon)A & A \geq 0\\(1-\epsilon)A & A < 0\end{cases}$$

$\overline{A_h^{\pi_\theta}}(s,a)$ has a similar computation.

PPO-CBF, PPO-SI. Constraint functions of these baselines are the same as the off-policy version, only the base algorithm is replaced with PPO (Schulman et al., 2017). Compared with Algorithm 2, only the computation of cost-to-go is replaced with the energy-function-based versions.

C.3.2. Hyperparameters

See Table 2.

Algorithm 2 Reachable Constrained Optimization (RCO)

```
Require: Initial policy parameters $\theta_0$, value and cost value function parameters $\omega_0, \phi_0$, multiplier network parameters $\xi_0$
for $k = 0, 1, 2, \dots$ do
    Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ with policy $\pi_{\theta_k}$, where $\tau_i$ is a $T$-step episode.
    Compute reward-to-go $\hat{R}_t \doteq \sum_{i=t}^{T} \gamma^i r_i$ and cost-to-go $\hat{H}_t \doteq \max_t h_t$.
    Compute advantage functions $A^{\pi_{\theta_k}}, A_h^{\pi_{\theta_k}}$, according to the value function $V_{\omega_k}$ and safety value function $V_{h_{\phi_k}}$. Compute the multiplier $\lambda_\xi$.
    Fit value function, safety value function by regression on mean-square error.
    Update the policy parameters $\theta$ by minimizing (24).
    Update the multiplier parameters $\xi$ by maximizing (24).
end for
```

Table 2. Detailed hyperparameters of on-policy algorithm and baselines.

Algorithm	Value
Shared	
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)
Approximation function	Multi-layer Perceptron
Number of hidden layers	2
Number of hidden units per layer	64
Nonlinearity of hidden layer	ELU
Nonlinearity of output layer (other than multiplier net)	linear
Critic learning rate	Linear annealing $3\text{e}{-4} \to 0$
Reward discount factor (γ)	0.99
Cost discount factor (γ_c)	0.99
GAE parameters	0.95
Batch size	8000
Max episode length (N)	1000
Actor learning rate	Linear annealing $3\text{e}{-4} \to 0$
Clip ratio	0.2
KL margin	1.2
RCO, PPO-CBF, PPO-SI	
Nonlinearity of output layer, multiplier net	softplus
Multiplier learning rate	Linear annealing $1\text{e}{-4} \to 0$
PPO-Lagrangian	
Init λ	$0.268(softplus(0))$
PPO-SI	
σ, n, k	0.1, 2, 1
PPO-CBF	
μ	0.1

D. Details about Experiments

D.1. Quadrotor Trajectory Tracking in safe-control-gym

Details about the quadrotor trajectory tracking task and training will be covered in this section. The task for the quadrotor is to track a counter-clockwise circle trajectory as accurately as possible while keeping its altitude z between $[0.5, 1.5]$, meaning the lower and upper bound of a tunnel. Note that only the next waypoint is accessible to the quadrotor at each time step, so no planning or predictive control in advance exists in this task.

Elements of the RL setting. The state space $\mathcal{S} \subseteq \mathbb{R}^{12}$ consists of the current state of the quadrotor $\mathbf{x} = [x, \dot{x}, z, \dot{z}, \theta, \dot{\theta}]^T$ and the information of the next waypoint $\mathbf{x}^{\text{ref}}$, thus $s_t = [\mathbf{x}_t; \mathbf{x}_t^{\text{ref}}]$. The action is the thrusts given by the two motors on both sides $[T_1, T_2]$, whose value will be normalized to $[0,1] \times [0,1]$. The system dynamics and information about the whole trajectory are inaccessible to the agent. The circle center is at $(0,1)$ and its radius is 1. The circle is discretized into 360 points so at each time step the reward function is the weighted sum of the difference between $(\mathbf{x}, a)$ and the reference $(\mathbf{x}^{\mathrm{ref}}, a^{\mathrm{ref}})$, specifically, $r(s_t, a_t) = -(\mathbf{x}_t - \mathbf{x}_t^{\mathrm{ref}})^T Q(\mathbf{x}_t - \mathbf{x}_t^{\mathrm{ref}}) - (a_t - a_t^{\mathrm{ref}})^T R(a_t - a_t^{\mathrm{ref}})$ where $Q = diag(10, 1, 10, 1, 0.2, 0.2), R = diag(1e-4, 1e-4)$. The constraint is $0.5 \leq z \leq 1.5$.

Initialization. For better exploration and generality of the learned feasible set and policy, we initialize the quadrotor uniformly in a rectangle in the xz-plane with uniformly distributed vertical and horizontal speed, pitch angle and pitch angle rate, specific ranges in Table 3. The nearest discrete waypoint on the trajectory to the initial location of the quadrotor is assigned as the start waypoint. In other words, the start waypoints change as the initial location changes. Quadrotor initialized at the center will be assigned a start waypoint randomly.

Table 3. The Initialization Range of Each Variable

Variable	Range
x	$[-1.5, +1.5]$
$\dot{x}$	$[-1.0, +1.0]$
z	$[0.25, +1.75]$
$\dot{z}$	$[-1.5, +1.5]$
θ	$[-0.2, +0.2]$
$\dot{\theta}$	$[-0.1, +0.1]$

Training. At each time step, the quadrotor outputs the two torques based on its state, including the waypoint next to the one in the last time step in the counter-clockwise direction. Then it receives the state transition, the reward and constraint function or cost signal. The (s, a, r, s', h, c) will be sent to the replay buffer. Simultaneously, the learner gets batches of samples from the replay buffer and compute gradients to update the function approximators. The maximum length T of an episode equals to the number of the discrete waypoints, i.e. 360. The episode will be ended and reset when the maximum length is reached or the quadrotor flies out of the bounded region $\{s \mid |x| \leq 2, |z| \leq 3\}$.

Evaluation. The policy is evaluated for four runs at one time. It is initialized statically at $(1,1), (-1,1), (0, 0.53), (0, 1.47)$ respectively in a run where safety can be guaranteed by hovering so the four initial states are feasible. Then the average return $\sum_{t=0}^{T-1} r(s_t, a_t)$ and constraint violation rate $\frac{\sum_{t=0}^{T-1} c(s_t)}{T}$ are taken as the performance and constraint-satisfaction metrics, respectively.

Feasible sets slices. The approximated constrained function in each algorithm ($Q_h^\pi(s,a), B^\pi(s,a), \Delta(s,a)$ in RAC, SAC-CBF, SAC-SI, respectively) is a function $f : \mathbb{R}^{12} \mapsto \mathbb{R}$. Hence, we need to project the high-dimensional state to a lower one to visualize the constrained function. Because the imposed constraints is about the z-coordinate, we choose to project each state onto the xz-plane and observe the changing trend with varying $\dot{z}$. Coordinates in x- and z-axis are uniformly sampled from the set $\{(x,z) \mid |x| < 1.5, 0.5 < z < 1.5\}$ while $\dot{z}$ is chosen among $\{-1, 0, 1\}$ and $\dot{x}$, θ and $\dot{\theta}$ are all set to zero. The tracking waypoint of each sample is the nearest one on the circle trajectory to the (x,z) sample. Then we generate state s for a given (x,z) tuple according to the aforementioned rules and get action a from the trained policy. The constrained value can be calculated with $f(s,a)$.