SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2024, Vol. 67, Iss. 3, 132202:1–132202:19
https://doi.org/10.1007/s11432-023-3872-9

Constrained reinforcement learning with statewise projection: a control barrier function approach

Xinze JIN, Kuo LI & Qingshan JIA*

Department of Automation, Tsinghua University, Beijing 100084*, China*

Received 13 February 2023/Revised 25 May 2023/Accepted 28 June 2023/Published online 19 February 2024

Abstract Safety is a critical issue for reinforcement learning (RL), as it may be risky for some actual applications if the learning process involves unsafe exploration. Instead of formulating constraints as expectation-based in constrained RL, considering statewise safety in constrained RL is more meaningful. This work aims to address the issue of safe projection in RL by introducing a control barrier function that inherently learns a safe policy through a set certificate. We seek to analyze some theoretical properties of safe projection in the learning process, including convergence and performance bound, and extend the discussion into ensembles and guided controllers. Moreover, we approach analytical solutions for deterministic and stochastic system dynamics. Experimental results in different tasks show that the proposed method achieves better effects in terms of both performance and safety.

Keywords reinforcement learning, safe projection, control barrier function

1 Introduction

Reinforcement learning (RL) [1] has been put into practice successfully in various applications, including video games [2] and robotic control [3]. The key mechanism behind it is to iterate the decision-making capability of an agent by interacting with the environment through trial and error. During the learning process, an agent needs to explore from an unknown state space to learn the available policy [4]. However, such exploration may involve unsafe interactions. Since the agent cannot realize such dangerous actions before experiencing them, deploying algorithms in safety-critical cases with considerable costs of failure is impractical. Because of this challenge, safe RL [5] has attracted widespread attention in academia, with its contents ranging from theoretical to empirical. Among them, constrained policy optimization [6] provides a specific view for formulating constraints in a discounted expected formula akin to the description of rewards. However, it usually takes expectation-based constraints instead of statewise constraints into account, which may cause the system design to lack intended instruments.

Policy learning with statewise safety concerns is challenging as there are usually conflicting criteria between the performance objective and the constraints [7], which entail a trade-off between both components that should be considered. A potential remedy thereof is formulation using Lagrangian-based methods [8], which weigh constraints as soft items when optimizing the objective function. However, this incurs extensive hyperparameter tuning and may inevitably violate constraints within different states. Thus, an additional safety layer [9] is used to map the output within a safe set using conditional gradient descent. However, the parameters derived from Karush-Kuhn-Tucker conditions are not always differentiable for implementing backpropagation. Moreover, guaranteeing safety requires carefully performing policy updates from the ground up to satisfy constraints step by step while maintaining existing properties as much as possible.

There has been an increasing number of studies in the context of nonlinear safe control. For instance, Sidrane et al. [10] abstracted nonlinear functions with a set of optimally tight piecewise linear bounds. Meanwhile, Wabersich and Zeilinger [11] introduced a predictive safety filter to transfer a constrained dynamical system into an unconstrained safe system. Gurriet et al. [12] decoupled performance from

* Corresponding author (email: jiaqs@tsinghua.edu.cn)

enforcing hard safety constraints. Ghaemi and Vecchio [13] considered the control problem for uncertain systems with imperfect information. By updating the safety policy regularly through these means, safety can be ensured. Although the above methods may work well with a small number of state variables, they may not be effective when dealing with large numbers of state variables [14,15] or may lead to inaccurate approximations over time. In other words, they involve solving a finite-horizon optimal control problem in a receding-horizon manner to approximate an optimal control policy that would work indefinitely, which means verifying the properties of the control policy in isolation does not address the entire closed-loop system.

Furthermore, existing control techniques often necessitate explicit analytical models to develop control laws, and these guarantee certificates are vulnerable to uncertainties in the underlying model. In this regard, Fisac et al. [16] provided a classic Hamilton-Jacobi framework to guarantee constraint satisfaction. Besides, Liu et al. [17] designed a controller based on backstepping techniques, and Zhu et al. [18] transformed safety analysis into a class of reachability issues. However, building a safe set relies on specific knowledge of a system. To this end, learning-based flow provides an alternative for exploring unknown dynamics. However, current learning-based efforts have difficulties in persistently satisfying safety specifications during learning episodes. Prior studies mainly learned from logged demonstration data [19] or leveraged the pretrained baseline policy as guidance [20]. Consequently, ensuring compliance with input constraints and preventing potential unsafe states when designing a feedback controller are crucial. However, obtaining a policy that meets safe action constraints could be challenging.

From the perspective of learning-based control, an intuitive way to address the constrained optimization problem is using a solvable controller, namely, safe projection. After the nominal controller generates an executable policy, safe projection is operated as a guard to provide minimally invasive control that is close to the original policy. Numerous approaches have been proposed in the past years to combine learning-based methods with classic control techniques, such as model predictive control [21] and temporal logic [22]. If there exists a feasible policy that can persistently promise forward invariance within the state space to satisfy the safe constraints, it can help reduce redundant computations in the entire iterations. In this respect, the control barrier function (CBF) [23] can serve as a powerful tool, either pressing on constraints based on the iterated control law [24] or learning a safe policy through the set certificate [25]. Current efforts use it to design a modified controller for stability [26] or to enhance the efficiency of the learning process with Gaussian assumption [27]. However, certain issues remain, especially for the properties of projection and analytical solutions that could be embedded with general algorithms.

The introduction of learning-enabled components into safety-critical control tasks raises issues on learning effect, constraint satisfaction, and extensibility. In this work, we propose a framework to address safe projection in RL using the CBF to improve the convergence of performance and safety measurements. We demonstrate it with a theoretical analysis of the properties of safe projection, which encourages exploration without knowing explicit system dynamics.

The main contributions of this study are highlighted as follows:

• In terms of the learning effect of safe projection, we analyzed the convergence of the projection-based learning process of the value function and extended it into ensembled networks. A proof based on contraction mapping was given with the discussion of overestimation issues.

• In terms of the robustness of constraint satisfaction, we derived a lower bound on performance improvement of safe learning with policy optimization and extended it into a guided deployment. The algorithm provided an optimally tight formulation based on the perturbation theory with the discussion of policy calibration versus mere compensation.

• In terms of the extensibility of the proposed framework, we adapted a projection mechanism with different RL algorithms in various settings and provided tractable solutions for formulations in deterministic and stochastic dynamics. Experimental results on multiple tasks showed benefits in both learning effects and safety measurements.

The rest of this paper is structured as follows. We introduced some background information on the barrier certificate and RL in Section 2. Then, we formulated the designed controller with safe projection under deterministic dynamics and stochastic dynamics in Section 3 and derived the theoretical results in convergence analysis and performance bound in Section 4. We conducted several numerical experiments in Section 5 and finally highlighted the work and future direction briefly in Section 6.

2 Preliminaries

2.1 Barrier certificate

The main idea behind the CBF is to guarantee the forward invariance of a safe set rather than a non-periodic constraint. For a dynamic system in the control-affine form $\dot{x} = f(x) + g(x)u$, where the state $x \in \mathbb{R}^n$ and control $u \in U \subset \mathbb{R}^m$, f and g are locally Lipschitz continuous.

Consider the superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, through which the safe set $\mathcal{C}$ is defined as

$$\mathcal{C} : \{x \in \mathbb{R}^n \mid h(x) \geqslant 0\}. \tag{1}$$

The time derivate of $h(x)$ along the state trajectories could be written in Lie derivative:

$$\frac{\mathrm{d}h(x)}{\mathrm{d}t} = \frac{\partial h(x)}{\partial x}\dot{x} = \frac{\partial h(x)}{\partial x}(f(x) + g(x)u) = L_f h(x) + L_g h(x)u. \tag{2}$$

The function h can be regarded as CBF, if there exists an extended class-$\mathcal{K}$ function κ such that

$$\sup_{u \in U}[L_f h(x) + L_g h(x)u] \geqslant -\kappa(h(x)). \tag{3}$$

The admissible control space can then conclude into

$$S(u) = \{u \in U \mid L_f h(x) + L_g h(x)u + \kappa(h(x)) \geqslant 0\}, \tag{4}$$

which could guarantee the forward invariant property without the need for explicit computation.

2.2 Reinforcement learning

RL is used to learn the optimal control policy that minimizes the supposed cost function without requiring complete knowledge of the system dynamics.

The basic idea is derived from the Markov decision process (MDP), which takes a tuple $(\mathcal{S}, \mathcal{A}, R, P, d_0)$ into consideration. Here, $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function, and $d_0 : \mathcal{S} \to [0, 1]$ is the initial state distribution.

Given the agent's current state s, the policy $\pi(a|s) : \mathcal{S} \times \mathcal{A}$ selects an action for the agent to take. The objective is to find the optimal policy π^* maximizing a performance measure $J(\pi)$, which is typically taken as an infinite horizon cumulative discounted reward.

For calculating $J(\pi) := \mathbb{E}_{s_0,a_0,\ldots}[\sum_{t=0}^{\infty} \gamma^t r(s_t)]$, where $s_0 \sim d_0, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)$, several value estimations are critical. The state-action value function Q^π is defined as $Q^\pi(s_t, a_t) = \mathbb{E}_{s_{t+1},a_{t+1},\ldots}[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})]$, the value function V^π is defined as $V^\pi(s_t) = \mathbb{E}_{a_t,s_{t+1},\ldots}[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})]$, and the advantage function A^π is defined as $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$ accordingly. Besides, $d^\pi(s) = (1 - \gamma)\sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)$ denotes the discounted future state distribution, which could express the difference in performance between two policies.

For continuous control, the parametrized policy π_θ is usually updated by taking the gradient of the expected return $\nabla_\theta J(\pi_\theta)$ using the policy gradient algorithm $\nabla_\theta J(\pi_\theta) = \mathbb{E}_s[\nabla_a Q^{\pi_\theta}(s, a)|_{a=\pi_\theta} \nabla_\theta \pi_\theta]$. The most commonly used gradient estimator has the form $\hat{\mathbb{E}}_t[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t]$, where $\hat{A}_t$ is the estimated advantage and $\hat{E}_t$ is the empirical expectation over timesteps.

3 Problem formulation

In this section, we consider a constrained optimization problem on statewise safety. The objective is to find the optimal policy π^* maximizing the performance measure $J(\pi) = \mathbb{E}_{s_0,a_0,\ldots}[\sum_{t=0}^{\infty} \gamma^t r(s_t)]$ as well, where $s_0 \sim d_0, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)$ follow the system dynamics, and the constraints are depicted by a barrier certificate derived from (4) for safety measurement. Then, we break the problem down under the settings of deterministic dynamics (the policy could be depicted as a specific action) and stochastic dynamics (the policy is depicted within a distribution depending on both states and actions), where the variables for optimization and constraint are kind of different in the formulation. For deterministic dynamics, we can solve quadratic programming (QP) directly under a control-affine basis. Meanwhile, as issues are more complex in stochastic dynamics, we need a two-step approach to decouple the projection to acquire the desired guarantees.

3.1 Deterministic dynamics

The time evolution of the system with control-affine deterministic dynamics is given by

$$s_{t+1} = f(s_t) + g(s_t)a_t, \tag{5}$$

where $s_t \in S$, $a_t \in A$, and f and g compose a nominal model of the dynamics.

RL is used to learn the optimal action a^* that maximizes the expected return without requiring complete knowledge of the system dynamics. A control policy could be learned using the Bellman equation, although it may lead to unsafe executions. Meanwhile, the introduction of CBF could help synthesize an action $a^\perp$ subject to safety constraints. A direct way is to construct a QP problem merging the minimally-invasive objective and safety constraints, which could be efficiently solved at each time step. The constrained optimization problem with respect to the control variable a is formulated as

$$a^\perp = \underset{a}{\operatorname{argmin}} \ \|a - a^*\|^2 \quad \text{s.t.} \quad L_f h(s) + L_g h(s) a \geqslant 0, \tag{6}$$

where the constraint means that the CBF controller serves as a projection operator to filter the selected action. Intuitively, the part of CBF compensates for the part of RL to ensure safety, which could ensure that the nominal action is within the safe region every time step.

3.2 Stochastic dynamics

Leveraging the idea in Yang et al. [28], we decoupled the designed controller and implemented CBF-based constraints as a safe projection in a two-step approach to update the policy.

The first step updates the policy to optimize the expected return using policy optimization within a trust region. The goal is to maximize the reward advantage function $A^\pi(s,a)$ under a Kullback-Leibler (KL) divergence constraint, which updates the intermediate policy $\pi^{k+\frac{1}{2}}$ within a δ-neighborhood of π^k as

$$\pi^{k+\frac{1}{2}} = \arg\max_{\pi} \mathbb{E}_{s\sim d^{\pi^k}, a\sim\pi}\left[A^{\pi^k}(s,a)\right] \quad \text{s.t.} \quad \mathbb{E}_{s\sim d^{\pi^k}}\left[D_{\mathrm{KL}}\left(\pi(\cdot,s), \pi^k(\cdot,s)\right)\right] \leqslant \delta. \tag{7}$$

This reward improvement step could regulate the distance measurement between the intermediate policy $\pi^{k+\frac{1}{2}}$ and the original policy π^k and guarantee reward improvement at the same time.

The second step performs a minimally-invasive policy update using CBF conditions as safe constraints. The goal is to project the intermediate policy $\pi^{k+\frac{1}{2}}$ onto the safe set by minimizing the distance around $\pi^{k+\frac{1}{2}}$ using a KL divergence projection as

$$\pi^{k+1} = \arg\min_{\pi} D_{\mathrm{KL}}\left(\pi, \pi^{k+\frac{1}{2}}\right) \quad \text{s.t.} \quad \mathbb{E}_{s\sim d^{\pi^k}, a\sim\pi}\left[L_f h(s) + L_g h(s) a\right] \geqslant 0. \tag{8}$$

This constraint-satisfying projection step can ensure that the executed policy π^{k+1} is close to $\pi^{k+\frac{1}{2}}$ within the safe set. Notably, here we used KL divergence instead of the L2 norm in projection. We further show that using such a projection can provide provable guarantees in performance bound.

In this way, it grants the agent the capability of improving the reward and remains constraint-satisfying, in contrast to previous methods that rely on line search to gradually ensure constraint satisfaction.

4 Main results

4.1 Convergence analysis

In this part, we discuss the convergence of the CBF-based RL framework under the setting of Q-learning [29]. The main theoretical result is provided in Theorem 1 and Corollary 1. Lemma 1 is used to derive the contraction mapping property, whereas Lemma 2 serves as preliminary conditions to check in the infinite stochastic process.

Lemma 1. The Bellman optimality operator defined as $\mathcal{T}(v) = \max_{a\in A}\left[R^a + \gamma P^a v\right]$ is a contraction in uniform norm $\|\mathcal{T}(u) - \mathcal{T}(v)\|_\infty \leqslant \gamma \|u - v\|_\infty$, which means that $\mathcal{T}$ converges to a unique fixed point at a linear convergence rate of γ.

Proof. See Appendix A for details.

Lemma 2. Consider a stochastic process $(\alpha_t, \Delta_t, F_t), t \geqslant 0$ taking values in $\mathbb{R}$ and defined as

$$\Delta_{t+1}(x_t) = (1 - \alpha_t(x_t))\,\Delta_t(x_t) + \alpha_t(x_t)\,F_t(x_t). \tag{9}$$

Then, Δ_t converges to 0 with probability 1 (w.p.1) under the following conditions:
(1) The set of possible states X is finite;
(2) $\alpha_t(x_t) \in [0,1], \sum_t \alpha_t(x_t) = \infty, \sum_t (\alpha_t(x_t))^2 < \infty$ w.p.1;
(3) $\|\mathbb{E}[F_t(x_t)\,|P_t]\|_\infty \leqslant \eta\,\|\Delta_t\|_\infty + c_t$, where $\eta \in [0,1)$ and c_t converges to 0 w.p.1;
(4) $\mathrm{Var}[F_t(x_t)\,|P_t] \leqslant K(1 + \eta\,\|\Delta_t\|_\infty)^2$, where $K > 0$ is some constant.

Proof. See Singh et al. [30] for details.

Here, we consider a finite MDP setting. For the designed controller in Subsection 3.1, define $a^* = \mathrm{argmax}_{a\in\mathcal{A}}\,Q(s',a)$ and $a^\perp = \mathrm{argmax}_{a\in\mathcal{A}_s}\,Q(s',a)$, where $\mathcal{A}$ means the normal action set and $\mathcal{A}_s$ means the safe set depicted by the safety constraints in (43). Set $y_t = r_t + \gamma Q_t(s_{t+1}, a^\perp)$ and update the tabular value estimate Q_t with respect to the target y_t and the learning rate $\alpha_t(s,a)$ as

$$Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha_t\,(y_t - Q_t(s_t,a_t)). \tag{10}$$

After obtaining a^* based on the current value function, we project it into a safe action $a^\perp$ following the technique in (43). In CBF-based Q-learning, such projection may influence the update of the Q factor as the initial policy in the next time step is $a^\perp$ rather than a^*, if feasible. Next, we try to analyze the convergence of CBF-based Q-learning.

Theorem 1. CBF-based Q-learning defined by updating (10) converges to the optimal value function w.p.1 under the following conditions:
(1) The MDP is finite, $|S \times A| < \infty$;
(2) $\alpha_t \in [0,1], \sum_t \alpha_t = \infty, \sum_t \alpha_t^2 < \infty$ w.p.1;
(3) Each state-action pair is visited infinitely often;
(4) The reward function is bounded;
(5) The discount factor $\gamma \in [0,1)$.

Proof. Applying $X = S \times A$, the first and second conditions of Lemma 2 correspond to the first and second conditions of Theorem 1, respectively. Below, we show that the third and fourth conditions of Lemma 2 hold, respectively.

First, we take $\Delta_t = Q_t(s_t,a_t) - Q^*(s_t,a_t)$ and abbreviate $Q^*(s_t,a_t)$ as Q^* and $\alpha_t(s_t,a_t)$ as α_t in the following paragraphs. Then, Δ_{t+1} is updated with

$$\begin{aligned}\Delta_{t+1}(s_t,a_t) &= Q_t(s_t,a_t) + \alpha_t\,(y_t - Q_t(s_t,a_t)) - Q^* \\ &= (1-\alpha_t)\,(Q_t(s_t,a_t) - Q^*) + \alpha_t\,(y_t - Q^*) \\ &= (1-\alpha_t)\,\Delta_t(s_t,a_t) + \alpha_t F_t(s_t,a_t).\end{aligned} \tag{11}$$

Note that $F_t(s_t,a_t) := y_t - Q^*$, which means

$$\begin{aligned}F_t(s_t,a_t) &= r_t + \gamma Q_t\left(s_{t+1}, a^\perp\right) - Q^* \\ &= r_t + \gamma Q_t(s_{t+1},a^*) - Q^* + \gamma\left(Q_t(s_{t+1},a^\perp) - Q_t(s_{t+1},a^*)\right) \\ &= F_t^Q(s_t,a_t) + c_t.\end{aligned} \tag{12}$$

In the above formula, $F_t^Q(s_t,a_t) := r_t + \gamma Q_t(s_{t+1},a^*) - Q^*$ denotes the value of F_t under Vanilla Q-learning and $c_t := \gamma\left(Q_t(s_{t+1},a^\perp) - Q_t(s_{t+1},a^*)\right)$ denotes the tail end under CBF-based Q-learning.

We must verify $\|\mathbb{E}[F_t^Q(s_t,a_t)\,|P_t]\|_\infty \leqslant \gamma\|\Delta_t\|_\infty$ and that c_t converges to 0, respectively, to satisfy the condition of the expectation in Lemma 2.

Applying Proposition 1, we first have

$$\begin{aligned}\left\|\mathbb{E}\left[F_t^Q(s_t,a_t)\,|P_t\right]\right\|_\infty &= \|\mathbb{E}[r_t + \gamma Q_t(s_{t+1},a^*) - Q^*|P_t]\|_\infty \\ &= \|(\mathcal{T}Q_t)(s_t,a_t) - (\mathcal{T}Q^*)(s_t,a_t)\|_\infty \\ &\leqslant \gamma\,\|Q_t(s_t,a_t) - Q^*\|_\infty \\ &= \gamma\,\|\Delta_t\|_\infty.\end{aligned} \tag{13}$$

Then, for the item of c_t, as the updating process in (10) is only related to actions in the safe set, we assign the value of $Q(s, a^*)$ in $A \setminus A_s$ as $Q(s, a^\perp)$, which means $Q_t(s_{t+1}, a^\perp) = Q_t(s_{t+1}, a^*)$ in the whole domain and induces that $c_t = 0$ naturally.

We consider the item $\mathrm{var}[F_t^Q(s_t, a_t)|P_t]$ and $\mathrm{var}[c_t]$ in turns to satisfy the condition of variance in Lemma 2.

Using $\mathbb{E}[F_t^Q(s_t, a_t)|P_t]$ extracted from (13), we first get

$$\begin{aligned}
\mathrm{var}\left[F_t^Q(s_t, a_t)|P_t\right] &= \mathbb{E}\left[\left(F_t^Q(s_t, a_t) - \mathbb{E}\left[F_t^Q(s_t, a_t)|P_t\right]\right)^2\right] \\
&= \mathbb{E}\left[\left(r_t + \gamma Q_t(s_{t+1}, a^*) - Q^* - (\mathcal{T}Q_t)(s_t, a_t) + Q^*\right)^2\right] \\
&= \mathbb{E}\left[\left(r_t + \gamma Q_t(s_{t+1}, a^*) - (\mathcal{T}Q_t)(s_t, a_t)\right)^2\right] \\
&= \mathrm{var}\left[r_t + \gamma Q_t(s_{t+1}, a^*)|P_t\right].
\end{aligned} \tag{14}$$

As Q_t consisted of the expected future reward and the reward in different time horizons could be regarded as independent identically distribution (i.i.d.), then $\mathrm{var}[r_t + \gamma Q_t(s_{t+1}, a^*)|P_t]$ could be divided into $\mathrm{var}[r_t]$ and $\mathrm{var}[\gamma Q_t(s_{t+1}, a^*)|P_t]$.

The first one is related to the reward function and bound with K_r, while the second is derived as follows:

$$\begin{aligned}
\mathrm{var}\left[\gamma Q_t(s_{t+1}, a^*)|P_t\right] &= \gamma^2 \mathrm{var}\left[Q_t(s_{t+1}, a^*)|P_t\right] \\
&= \gamma^2 \mathrm{var}\left[Q_t(s_{t+1}, a^*) - Q^*|P_t\right] \\
&\leqslant \gamma^2 \mathbb{E}\left[\left(Q_t(s_{t+1}, a^*) - Q^*\right)^2\right] \\
&\leqslant \gamma^2 \|\Delta t\|_\infty^2.
\end{aligned} \tag{15}$$

Then, for the second item c_t, we have

$$\begin{aligned}
\mathrm{var}[c_t] &= \gamma^2 \mathrm{var}\left[Q_t(s_{t+1}, a^*) - Q_t\left(s_{t+1}, a^\perp\right)\right] \\
&\leqslant \gamma^2 \mathrm{var}\left[Q_t(s_{t+1}, a^*) - Q^*\right] + \gamma^2 \mathrm{var}\left[Q_t\left(s_{t+1}, a^\perp\right) - Q^*\right] \\
&\leqslant \gamma^2 \mathbb{E}\left[\left(Q_t(s_{t+1}, a^*) - Q^*\right)^2\right] + \gamma^2 \mathbb{E}\left[\left(Q_t\left(s_{t+1}, a^\perp\right) - Q^*\right)^2\right].
\end{aligned} \tag{16}$$

Here, we divide the variance of c_t into two expectations. Borrowed from the result of (13) directly, the first one is no more than $\gamma^2 \|\Delta t\|_\infty^2$. Besides, the nuance in the second one is only the choice of action, which may not influence the derivation in Proposition 1 by substituting the corresponding domain. Intuitively, we could reach the same bound of $\gamma^2 \|\Delta t\|_\infty^2$.

On the basis of (14)–(16), we derive that the variance of $F_t^Q + c_t$ is no more than $K_r + 3\gamma^2 \|\Delta t\|_\infty^2$. By adopting $K = \max\{K_r, 3\}$, we can verify the bound of $\mathrm{var}[F_t(s_t, a_t)|P_t]$.

Above all, CBF-based Q-learning defined by updating (10) converges to the optimal value function with all conditions satisfied.

Notably, in a function approximation setting, inaccurate values are easily induced when overestimation or disturbance occurs. Thus, an analogous update using disentangled value estimates from different networks may ease such symptoms. Here, we provide an extension of CBF-based Q-learning in an ensemble setting and show the applicability of the convergence results.

For the CBF-based Ensembled Q-learning with Q^0 to execute and $\{Q^i\}_{i=1}^N$ to amend value estimations, we define $a^* = \mathrm{argmax}_{a \in A} Q^0(s', a)$ and $a^\perp = \mathrm{argmax}_{a \in A_s} Q^0(s', a)$, where A means the normal set and A_s means the safe set. We set $y_t = r_t + \gamma \min_i Q_t^i(s_{t+1}, a^\perp)$ and then update the tabular value estimate $\{Q_t^i\}_{i=0}^N$ with respect to the target y_t and learning rate $\alpha_t(s, a)$ as

$$Q_{t+1}^i(s_t, a_t) = Q_t^i(s_t, a_t) + \alpha_t\left(y_t - Q_t^i(s_t, a_t)\right). \tag{17}$$

Corollary 1. CBF-based Ensembled Q-learning defined by updating (17) converges to the optimal value function w.p.1 under the following conditions:

(1) The MDP is finite, $|S \times A| < \infty$;

(2) $\alpha_t(s_t, a_t) \in [0, 1], \sum_t \alpha_t(s_t, a_t) = \infty, \sum_t (\alpha_t(s_t, a_t))^2 < \infty$ w.p.1;

(3) Each state-action pair is visited infinitely often;

(4) The reward function is bounded;
(5) The discount factor $\gamma \in [0,1)$.

Proof. Following the proof in Theorem 1, we show that the third and fourth conditions of Lemma 2 hold.

First, we take $\Delta_t = Q_t^0(s_t,a_t) - Q^*$ and note $\alpha_t(s_t,a_t)$ as α_t. Then, Δ_{t+1} is updated with

$$\begin{aligned}\Delta_{t+1}(s_t,a_t) &= Q_t^0(s_t,a_t) + \alpha_t\left(y_t - Q_t^0(s_t,a_t)\right) - Q^* \\ &= (1-\alpha_t)\left(Q_t^0(s_t,a_t) - Q^*\right) + \alpha_t(y_t - Q^*) \\ &= (1-\alpha_t)\Delta_t(s_t,a_t) + \alpha_t F_t(s_t,a_t).\end{aligned} \tag{18}$$

Marking $y_t = r_t + \gamma \min_i Q_t^i(s_{t+1},a^\perp) := r_t + \gamma Q_t^-(s_{t+1},a^\perp)$, then we have

$$\begin{aligned}F_t(s_t,a_t) &= r_t + \gamma Q_t^-\left(s_{t+1},a^\perp\right) - Q^* \\ &= r_t + \gamma Q_t^0(s_{t+1},a^*) - Q^* + \gamma\left(Q_t^-(s_{t+1},a^\perp) - Q_t^-(s_{t+1},a^*)\right) \\ &\quad + \gamma\left(Q_t^-(s_{t+1},a^*) - Q_t^0(s_{t+1},a^*)\right) \\ &= F_t^Q(s_t,a_t) + c_t + c_t^-.\end{aligned} \tag{19}$$

In the above formula, $F_t^Q(s_t,a_t) := r_t + \gamma Q_t^0(s_{t+1},a^*) - Q^*$ denotes the value of F_t under Vanilla Q-learning and $c_t + c_t^- := \gamma(Q_t^-(s_{t+1},a^\perp) - Q_t^0(s_{t+1},a^*))$ denotes the tail end under CBF-based Ensembled Q-learning. The analysis of F_t^Q and c_t is the same as that in Theorem 1. Here, we discuss only the property of c_t^-.

Let $\Delta_t^-(s_t,a_t) = Q_t^-(s_t,a_t) - Q_t^0(s_t,a_t)$. Then, c_t^- converges to 0 if Δ_t^- converges to 0.

$$\begin{aligned}\Delta_{t+1}^-(s_t,a_t) &= \left(Q_t^- + \alpha_t(y - Q_t^-)\right) - \left(Q_t^0 + \alpha_t(y - Q_t^0)\right) \\ &= \left(Q_t^- - Q_t^0\right) + \alpha_t\left(Q_t^0 - Q_t^-\right) \\ &= (1-\alpha_t)\Delta_t^-(s_t,a_t).\end{aligned} \tag{20}$$

Because $\alpha_t \in [0,1]$ and $\sum_t \alpha_t = \infty$, then α_t is not always 0, which easily concludes that Δ_t^- will converge to 0, implying that c_t^- converges to 0.

Then considering the variance of c_t^-, we have

$$\begin{aligned}\operatorname{var}\left[c_t^-\right] &\leqslant \gamma^2 \operatorname{var}\left[Q_t^-(s_{t+1},a^*) - Q_t^0(s_{t+1},a^*)\right] \\ &\leqslant \gamma^2 \operatorname{var}\left[Q_t^-(s_{t+1},a^*) - Q^*\right] + \gamma^2 \operatorname{var}\left[Q_t^0(s_{t+1},a^*) - Q^*\right] \\ &\leqslant \gamma^2 \mathbb{E}\left[\left(Q_t^-(s_{t+1},a^*) - Q^*\right)^2\right] + \gamma^2 \mathbb{E}\left[\left(Q_t^0(s_{t+1},a^*) - Q^*\right)^2\right] \\ &\leqslant 2\gamma^2 \|\Delta t\|_\infty^2.\end{aligned} \tag{21}$$

Notably, the last line is borrowed from the result of (13) directly. On the basis of the results in (14)–(16) and (21), we derive that the variance of $F_t^Q + c_t + c_t^-$ is no more than $K_r + 5\gamma^2\|\Delta t\|_\infty^2$. By adopting $K = \max\{K_r, 5\}$, we can satisfy the bound of the variance condition in Lemma 2.

Above all, CBF-based Ensembled Q-learning defined by updating (17) converges to the optimal value function with all conditions satisfied.

Besides the extension in multiple networks to alleviate inaccurate estimations, the critical network in the deterministic policy gradient is also updated as akin to Q-learning, which means the proof could extend to further prospects.

4.2 Performance bound

In this part, we discuss the performance bound of the CBF-based RL framework under the setting of policy optimization [31]. The main theoretical results are provided in Theorem 2 and Corollary 2. Meanwhile, Lemma 3 is used to derive the inequality needed for the perturbation theory, and Lemma 4 is used to present the property of the distance measurement.

On the basis of the preliminary definition mentioned in Subsection 2.2, we could obtain the expected reward of a different policy $\tilde{\pi}$ in terms of the advantage over π as

$$J(\tilde{\pi}) = J(\pi) + \sum_s d_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A^\pi(s,a). \tag{22}$$

However, the complex dependency of $d_{\tilde{\pi}}(s)$ on π makes it difficult to optimize directly. Instead, the local approximation to $J(\tilde{\pi})$ is introduced as

$$L(\tilde{\pi}) = J(\pi) + \sum_s d_\pi(s) \sum_a \tilde{\pi}(a|s) A^\pi(s,a), \tag{23}$$

where the visitation frequency d_π is used instead of $d_{\tilde{\pi}}$, ignoring changes in state visitation density due to changes in the policy.

Lemma 3. Let α denote the maximum total variation divergence between the stochastic policies π and $\tilde{\pi}$, defined as $D_{\mathrm{TV}}^{\max}(\pi,\tilde{\pi}) = \max_s \frac{1}{2}\sum_i |\pi(\cdot|s)_i - \tilde{\pi}(\cdot|s)_i|$. Then, we can conclude a bound in policy update toward the approximation of the true expected reward objective J as

$$J(\tilde{\pi}) \geqslant L(\tilde{\pi}) - \alpha^2 \frac{4\gamma\epsilon}{(1-\gamma)^2}, \tag{24}$$

where $\epsilon = \max_{s,a} |A^\pi(s,a)|$ and $\gamma \in [0,1)$ are the discount factors.

Proof. See Appendix B for details.

For the arbitrary distributions π and $\tilde{\pi}$, the TV divergence and KL divergence are related by $D_{\mathrm{TV}}(\pi,\tilde{\pi}) \leqslant \sqrt{D_{\mathrm{KL}}(\pi,\tilde{\pi})/2}$. With (23) and (24), the bound can be further viewed to describe the worst-case surrogate as

$$\begin{aligned} J(\pi') - J(\pi) &\geqslant \frac{1}{1-\gamma} \mathop{\mathrm{E}}_{\substack{s\sim d^\pi \\ a\sim\pi'}} \left[A^\pi(s,a) - \frac{4\gamma\epsilon}{1-\gamma} D_{TV}(\pi,\pi') \right] \\ &\geqslant \frac{1}{1-\gamma} \mathop{\mathrm{E}}_{\substack{s\sim d^\pi \\ a\sim\pi'}} \left[A^\pi(s,a) - \frac{2\gamma\epsilon}{1-\gamma} \sqrt{2 D_{KL}(\pi,\pi')} \right]. \end{aligned} \tag{25}$$

Lemma 4. If the current policy π^k satisfies the constraints in a closed and convex set, the KL divergence for the reward improvement step is $\mathbb{E}_{s\sim d^k}[D_{\mathrm{KL}}(\pi^{k+\frac{1}{2}}(\cdot|s), \pi^k(\cdot|s))] \leqslant \delta$. Then in the projection step, we could bound the performance within the same step size as

$$\mathbb{E}_{s\sim d^{\pi^k}} \left[D_{\mathrm{KL}} \left(\pi^{k+1}(\cdot|s), \pi^k(\cdot|s) \right) \right] \leqslant \delta. \tag{26}$$

Proof. According to the Bregman divergence projection inequality, with π^k being in the constraint set, and π^{k+1} being the projection of $\pi^{k+\frac{1}{2}}$ onto the constraint set, we have

$$\begin{aligned} \mathbb{E}_{s\sim d^{\pi^k}} \left[D_{\mathrm{KL}} \left(\pi^k(\cdot|s), \pi^{k+\frac{1}{2}}(\cdot|s) \right) \right] &\geqslant \mathbb{E}_{s\sim d^{\pi^k}} \left[D_{\mathrm{KL}} \left(\pi^k(\cdot|s), \pi^{k+1}(\cdot|s) \right) \right] \\ &\quad + \mathbb{E}_{s\sim d^{\pi^k}} \left[D_{\mathrm{KL}} \left(\pi^{k+1}(\cdot|s), \pi^{k+\frac{1}{2}}(\cdot|s) \right) \right]. \end{aligned} \tag{27}$$

As KL divergence is always greater than zero, then we have

$$\delta \geqslant \mathbb{E}_{s\sim d^{\pi^k}} \left[D_{\mathrm{KL}} \left(\pi^k(\cdot|s), \pi^{k+1}(\cdot|s) \right) \right]. \tag{28}$$

KL divergence is known to be asymptotically symmetric with the policy updating in a local neighborhood. Thus, Eq. (26) is followed.

Theorem 2. CBF-based policy optimization as defined by (7) and (8) has a lower bound on reward improvement in each step under the projection if the current policy satisfies the constraint, which is

$$J\left(\pi^{k+1}\right) - J\left(\pi^k\right) \geqslant -\frac{2\sqrt{2\delta}\gamma\epsilon}{(1-\gamma)^2}, \tag{29}$$

where $\epsilon = \max_{s,a} |A^\pi(s,a)|$, $\gamma \in [0,1)$ is the discount factor, and δ is the threshold of KL divergence.

Proof. As Eq. (8) only updates the policy of (7) with a safe projection, this means it may follow the reward improvement in the corresponding domain. On the basis of (25) derived from Lemma 3, we have the performance bound toward each policy update as

$$J\left(\pi^{k+1}\right) - J\left(\pi^k\right) \geqslant \frac{1}{1-\gamma} \mathop{\mathrm{E}}_{\substack{s\sim d^{\pi^k} \\ a\sim\pi^{k+1}}} \left[A^{\pi^k}(s,a) - \frac{2\gamma\epsilon}{1-\gamma} \sqrt{2 D_{\mathrm{KL}}\left(\pi^{k+1}, \pi^k\right)} \right]. \tag{30}$$

Furthermore, $A^{\pi^k}(s,a) \geqslant 0$ as there is at least one state-action pair with a positive advantage value and nonzero state visitation probability to improve the policy performance till convergence. Then, we could borrow the result in Lemma 4 and put forward (30) to

$$\begin{aligned} J\left(\pi^{k+1}\right) - J\left(\pi^k\right) &\geqslant \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi^k} \\ a\sim\pi^{k+1}}} \left[-\frac{2\gamma\epsilon}{1-\gamma} \sqrt{2D_{\mathrm{KL}}\left(\pi^{k+\frac{1}{2}}, \pi^k\right)} \right] \\ &\geqslant -\frac{2\sqrt{2\delta}\gamma\epsilon}{(1-\gamma)^2}. \end{aligned} \tag{31}$$

In learning-based safety-critical control, the RL controller provides a feedforward control, and the CBF controller compensates with the minimum control necessary to render the safe set forward invariant. Intuitively, the flowchart is derived as $u_k(s) = u_{\theta_k}^{\mathrm{RL}}(s) + u_k^{\mathrm{CBF}}(s, u_{\theta_k}^{\mathrm{RL}})$.

Furthermore, the deployed controller could learn around a safe region rather than $u_{\theta_k}^{\mathrm{RL}}$ to avoid operating in an irrelevant area of state space to achieve efficient learning. As depicted in Cheng et al. [27], for every policy iteration, the overall controller incorporates all previous CBF controllers as

$$\begin{aligned} u_k(s) &= u_k^{\mathrm{safe}}(s) + u_k^{\mathrm{CBF}}\left(s, u_{\theta_k}^{\mathrm{RL}} + \sum_{j=0}^{k-1} u_j^{\mathrm{CBF}}\right), \\ u_k^{\mathrm{safe}}(s) &= u_{\theta_k}^{\mathrm{RL}}(s) + \sum_{j=0}^{k-1} u_j^{\mathrm{CBF}}\left(s, \left\{u_{\theta_i}^{\mathrm{RL}}\right\}_{i=0}^{j-1}\right). \end{aligned} \tag{32}$$

The controller defined in this fashion leads to policy updates around the previously deployed controller, which enhances the efficiency of the learning process by encouraging the policy to operate in desired areas of the state space.

Corollary 2. CBF-based guided policy optimization achieves the performance guarantee $J(\pi_k^{\mathrm{safe}}) \geqslant J(\pi_{k-1}) - \frac{2\sqrt{2\delta}\gamma\epsilon}{(1-\gamma)^2}$ following the control law in (32), where $\lambda = \max_s |\mathbb{E}_{a\sim\pi_k^{\mathrm{safe}}}[A^{\pi_{k-1}}(s,a)]|$, $\gamma \in [0,1)$ is the discount factor, and δ is the threshold of KL divergence.

Proof. For the part of the RL controller, we only consider the reward improvement step. On the basis of the result in (25) and the property of the advantage function, we could easily derive the following inequality:

$$J\left(\pi_{\theta_k}^{\mathrm{RL}}\right) - J\left(\pi_{\theta_{k-1}}^{\mathrm{RL}}\right) \geqslant -\frac{2\sqrt{2\delta}\gamma\epsilon}{(1-\gamma)^2}. \tag{33}$$

Note that the implemented CBF controllers in (32) are all deterministic. Rewriting $\sum_{j=0}^{k-1} u_j^{\mathrm{CBF}} + u_k^{\mathrm{CBF}}$ as u_k^{CBFs}, we could then formulate (32) as

$$\begin{aligned} &\begin{cases} u_{k-1}(s) = u_{\theta_{k-1}}^{\mathrm{RL}}(s) + u_{k-1}^{\mathrm{CBFs}}(s), \\ u_k^{\mathrm{safe}}(s) = u_{\theta_k}^{\mathrm{RL}}(s) + u_{k-1}^{\mathrm{CBFs}}(s), \end{cases} \\ \Longrightarrow &\begin{cases} \pi_{k-1}(a|s) = \pi_{\theta_{k-1}}^{\mathrm{RL}}\left(a - u_{k-1}^{\mathrm{CBFs}}(s)|s\right), \\ \pi_k^{\mathrm{safe}}(a|s) = \pi_{\theta_k}^{\mathrm{RL}}\left(a - u_{k-1}^{\mathrm{CBFs}}(s)|s\right). \end{cases} \end{aligned} \tag{34}$$

Substituting the expression into (33), we have the bound

$$J\left(\pi_k^{\mathrm{safe}}\right) \geqslant J\left(\pi_{k-1}\right) - \frac{2\lambda\gamma}{(1-\gamma)^2}\delta_\pi. \tag{35}$$

5 Experiments

For the deterministic dynamics setting in Subsection 3.1, it is not such hard to filter the safe action by solving $a \geqslant -L_g h(s)^{-1} L_f h(s)$. Thus, we used a pretrained neural network with sampling approximation to translate the constraints depicted by the zero sublevel set into a binary value to accelerate

Table 1 Comparative indices for Q-learning[a)]

Index	Case	Vanilla Q-learning	Penalized Q-learning	Projected Q-learning
NER	(a)	$\mathbf{0.940} \pm 0.010$	0.541 ± 0.138	0.938 ± 0.006
	(b)	$\mathbf{0.959} \pm 0.006$	0.458 ± 0.074	$\mathbf{0.959} \pm 0.006$
	(c)	0.942 ± 0.014	0.437 ± 0.038	$\mathbf{0.964} \pm 0.006$
SPI	(a)	62.4	73.0	**100.0**
	(b)	59.2	70.6	**100.0**
	(c)	50.8	57.4	**100.0**

a) Numbers in bold represent the best value achieved by each method under the corresponding index.

the calculation. In this way, we could simply project the action set into a safe area during the learning process.

For the stochastic dynamics setting in Subsection 3.2, directly solving the CBF-based RL is impractical because of the computational cost when introducing a large neural network with lots of parameters. However, using an analytical approach (see Appendix C for details), we could derive the policy as

$$\theta^{k+1} = \theta^k + \sqrt{\frac{2\delta}{{\mu_1}^{\mathrm{T}} H^{-1} \mu_1}} H^{-1}\mu_1 + \max\left(0, \frac{-\sqrt{\frac{2\delta}{{\mu_1}^{\mathrm{T}} H^{-1}\mu_1}}{\mu_2}^{\mathrm{T}} H^{-1}\mu_1}{{\mu_2}^{\mathrm{T}} H^{-1}\mu_2}\right) H^{-1}\mu_2, \tag{36}$$

where θ is the parameter of the policy.

In this section, we showcase the experimental results of the proposed framework in different settings as the environment progressively becomes more complicated. First, we tested a series of Q-learning algorithms in an environment with discrete state space and discrete action space. The system dynamics of the toy MDP following a probability matrix, and the safety constraints were described as some forbidden state-action pairs. Second, we implemented homologous DQN algorithms in an environment with continuous state space and discrete action space. With the adoption of OpenAI Gym, those cases had inherent physical engines and different dimensions in system dynamics. Safety constraints were depicted as restrictions in state space and action space, rendering a little revision in the movement of every single step. Third, we benchmarked the policy optimization methods in an environment with continuous state space and continuous action space. The system dynamics and safety constraints were also based on the OpenAI Gym, but the dimensions of state and action were extended, making it more complex to handle. Finally, we used a case study of motion planning to illustrate the potential of our approach in more general tasks instead of predesigned games.

We tested the learning effect and safety measurement of the proposed method in the aforementioned situations, as elaborated further in the following segments. Notably, safety was specified as constraint satisfaction, with different definitions based on patterns in specific environments. Besides the normalized episodic return (NER) index, we counted a metric named step per intervention (SPI) to measure the interventions of different methods in the learning process.

5.1 Discrete state space and discrete action space

In this part, we implemented CBF-based Q-learning in some toy MDPs with different scales to test its performance in discrete state space and discrete action space. We took the Vanilla algorithm without any constraints (Vanilla Q-learning) to complete the task first and then used soft constraint as a penalty item added into the reward during the learning process (Penalized Q-learning) as a benchmark. Finally, we compared them with the proposed Projected Q-learning.

The performance was evaluated for every fixed episode and adapted to fit the task at different levels. The steps to be executed in each episode were set as 100, which means episodes would not be finished early unless the agent violated the constraints. We conducted five random trials and normalized the episodic return to plot the learning curve. Table 1 shows that Projected Q-learning dominated in NER (no more than 1) and SPI (no more than 100), while Penalized Q-learning was a little helpful in terms of safety but was weak in terms of performance as there was ambiguity when introducing penalties into the learning process. Moreover, Figure 1 shows that Projected Q-learning reached the same convergence as that of Vanilla Q-learning, even at a faster rate, which corresponded to the theoretical result in Theorem 1.

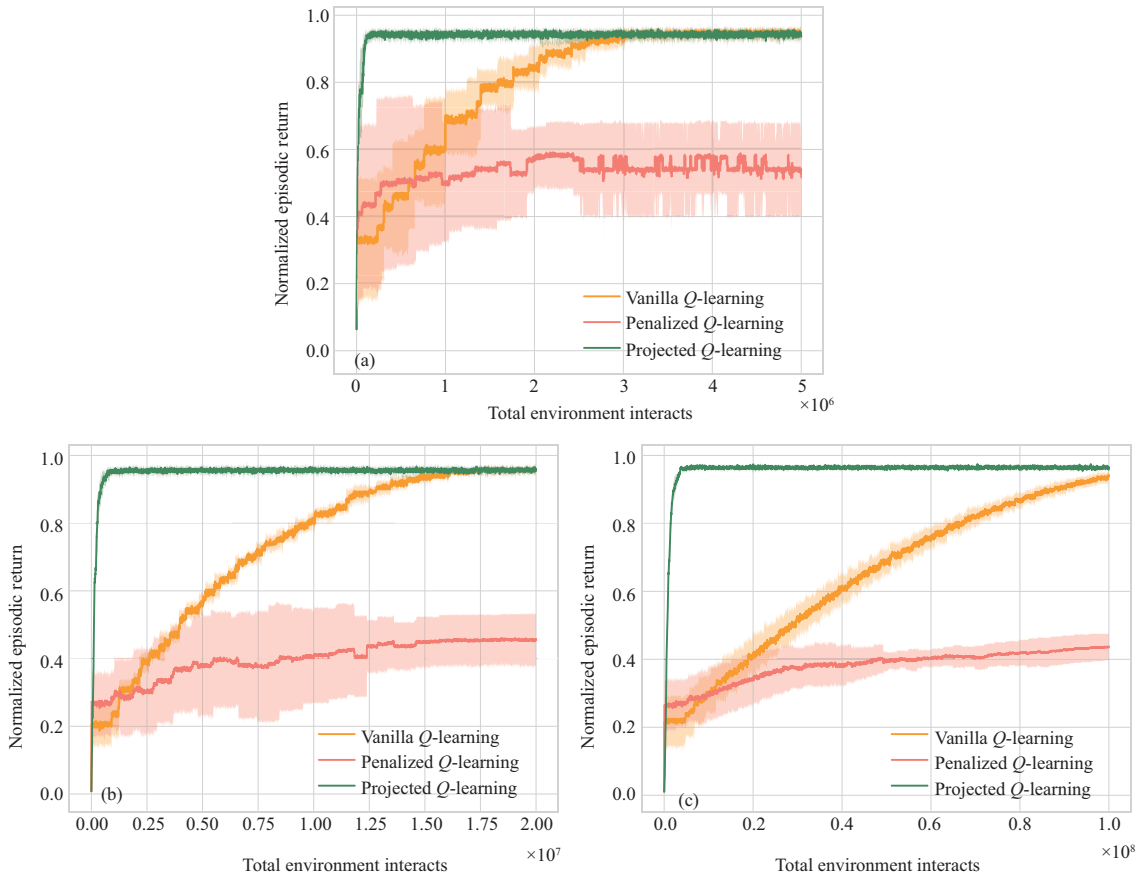

Figure 1 (Color online) Algorithm performance in discrete state space and discrete action space. (a) MDP-1 ($|S| = 10, |A| = 5$); (b) MDP-2 ($|S| = 20, |A| = 10$); (c) MDP-3 ($|S| = 50, |A| = 20$).

5.2 Continuous state space and discrete action space

In this part, we implemented CBF-based DQN in a classic control environment in OpenAI Gym to test its performance in continuous state space and discrete action space. We took the Vanilla algorithm without any constraints (Vanilla-DQN) to complete the task first. Because penalized constraint was shown to be less effective, we did not include it in this part. Instead, we tested the proposed Projected-DQN directly and extended it with additional networks (here, we adopted $N = 5$) to check the power of Ensembled-DQN when connecting with safe projection issues, which corresponded to the theoretical result in Corollary 1.

Although MountainCar, CartPole, and Acrobot are not specially designed for safety tasks, they are general and mature enough to test RL algorithms. We formulated barrier certificates based on their dynamics and set the safe threshold for each accordingly to specify the constraints. Akin to the settings in Subsection 5.1, we conducted five random trials and calculated the statistics of NER and SPI. Table 2 shows that the Projected-DQN and Ensembled-DQN showed advantages in NER and SPI compared with Vanilla-DQN, but the actual effect might be slightly different in specific tasks. The results show the effectiveness of projection in enhancing safety without sacrificing too much in terms of performance and could explain the power of introducing complex ensembles. Moreover, Figure 2 shows that both Projected-DQN and Ensembled-DQN presented learning processes similar to that of Vanilla-DQN, even under the existence of a neural network.

5.3 Continuous state space and continuous action space

In this part, we implemented CBF-based policy optimization in the Mujoco environment in OpenAI Gym to its performance in continuous state space and continuous action space. Notably, we incorporated proximal policy optimization with a penalty (Penalized-PO) as a benchmark, which approximately solves a KL-constrained update like trust—region policy optimization but penalizes the KL divergence in the

Table 2 Comparative indices for DQN[a)]

Index	Case	Vanilla-DQN	Projected-DQN	Ensembled-DQN
NER	(a)	**0.953** $\pm$ 0.013	0.927 $\pm$ 0.034	**0.953** $\pm$ 0.011
	(b)	0.994 $\pm$ 0.007	0.986 $\pm$ 0.018	**0.996** $\pm$ 0.005
	(c)	0.824 $\pm$ 0.034	0.742 $\pm$ 0.025	**0.851** $\pm$ 0.022
SPI	(a)	53.4	**91.8**	89.6
	(b)	56.8	90.4	**95.2**
	(c)	55.0	87.2	**93.0**

a) Numbers in bold represent the best value achieved by each method under the corresponding index.

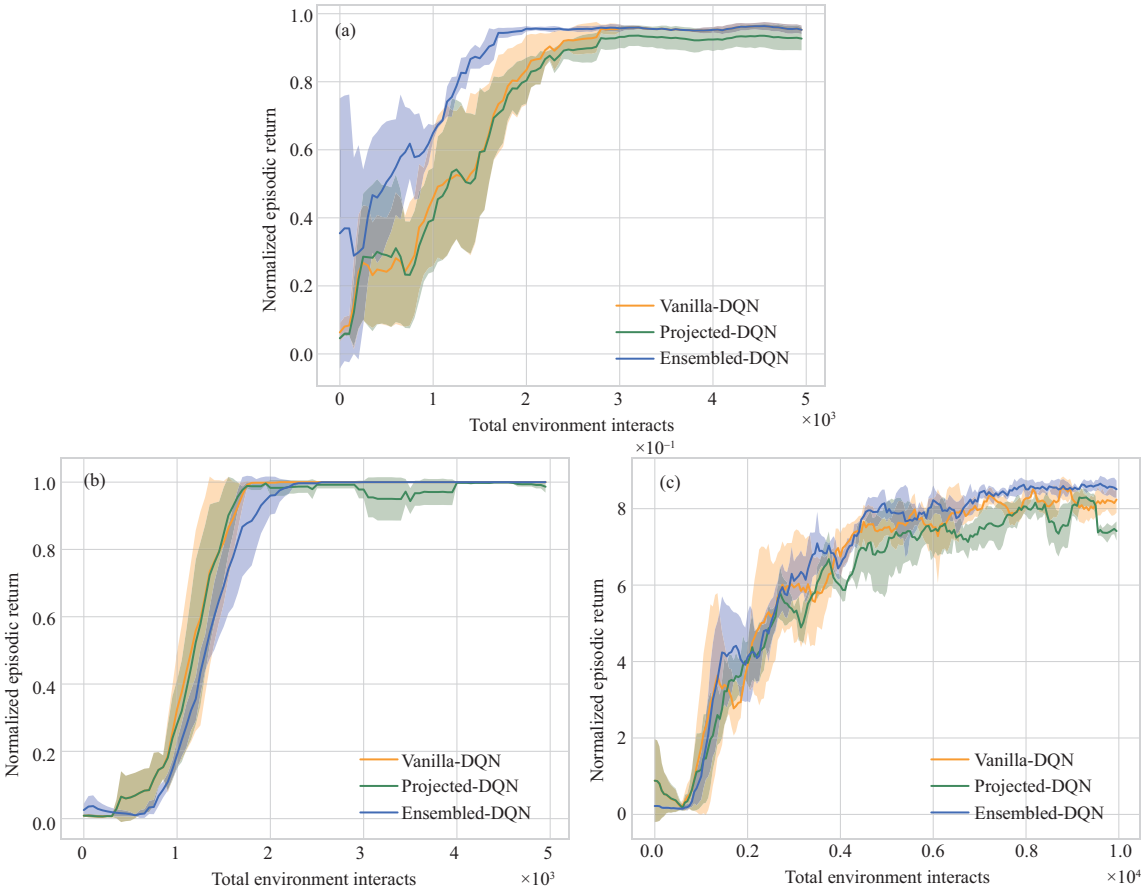

Figure 2 (Color online) Algorithm performance in continuous state space and discrete action space. (a) MountainCar ($d(S) = 2, |A| = 3$); (b) CartPole ($d(S) = 4, |A| = 2$); (c) Acrobot ($d(S) = 6, |A| = 3$).

objective function instead of making it a hard constraint. Moreover, we tested the proposed Projected-PO using an analytical solution and extended it into Guided-PO, as depicted in Corollary 2, to verify the effectiveness of the performance and safety improvements.

Here, we adopted the cases of Hopper, HalfCheetah, and Ant in the Mujoco environment and conducted five random trials for each. To describe safety constraints, we generated a series of features embedded with barrier certificates, which were based on system dynamics and could reflect the limitations in particular states and actions. As the dimensions of the state space and action space grew, they took more effort to get an ideal resolution with safety satisfaction. Table 3 shows that the Projected-PO and Guided-PO could improve the indices much better than Penalized-PO with soft constraints as the latter does not cover the minimally invasive step of projection. Besides, we could also find in Figure 3 that both Projected-PO and Guided-PO performed a smoother learning process than Penalized-PO when related to safety issues, while the guided one was even more effective to some extent.

Table 3 Comparative indices for policy optimization[a)]

Index	Case	Penalized-PO	Projected-PO	Guided-PO
NER	(a)	0.597 ± 0.090	0.442 ± 0.132	$\mathbf{0.664} \pm 0.215$
	(b)	0.697 ± 0.266	0.793 ± 0.180	$\mathbf{0.919} \pm 0.046$
	(c)	0.801 ± 0.027	$\mathbf{0.888} \pm 0.054$	0.866 ± 0.044
SPI	(a)	70.2	89.4	**93.8**
	(b)	63.8	**91.2**	90.4
	(c)	69.4	87.0	**92.6**

a) Numbers in bold represent the best value achieved by each method under the corresponding index.

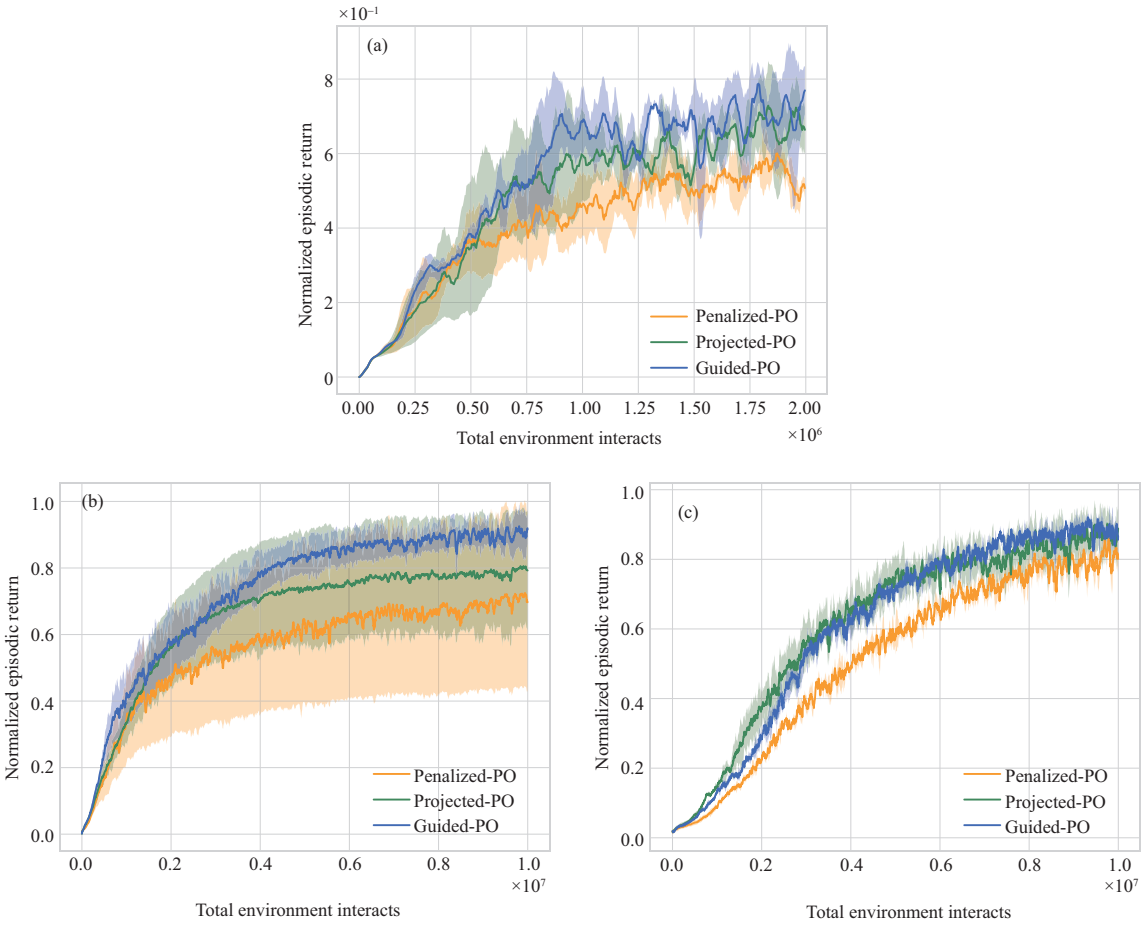

Figure 3 (Color online) Algorithm performance in continuous state space and continuous action space. (a) Hopper ($d(S) = 11, d(A) = 3$); (b) HalfCheetah ($d(S) = 17, d(A) = 6$); (c) Ant ($d(S) = 27, d(A) = 8$).

5.4 Case study

Besides the settings regarding the different statuses of state space and action space, we also illustrate a case representing a step toward the reliable deployment of the proposed method in more general cases.

In this part, we considered an unsignalized intersection case used in [32], as depicted in Figure 4. The blue vehicle is an ego agent, and the red vehicles are conflicts with unknown intentions. The objective of the ego vehicle is to pass the intersection successfully without colliding with the surrounding vehicles from other directions.

Referring to the formulation in [32], we derived the system dynamics and barrier certificate as follows. When the index $i = 0$, it refers to the ego vehicle, while $i \in 1, \ldots, N$ refers to the surrounding vehicles.

The state space is defined as $s = (s^{(0)}, s^{(1)}, \ldots, s^{(N)})$, encompassing the observable components, where $s^{(i)} = (x^{(i)}, y^{(i)}, \rho^{(i)}, \phi^{(i)})$, while the intended goal position and preferred speed are considered hidden components. Within the state space, $x^{(i)}$ and $y^{(i)}$ represent the position along different axes, respectively, whereas $\rho^{(i)}$ and $\phi^{(i)}$ represent the velocity magnitude and heading, respectively.

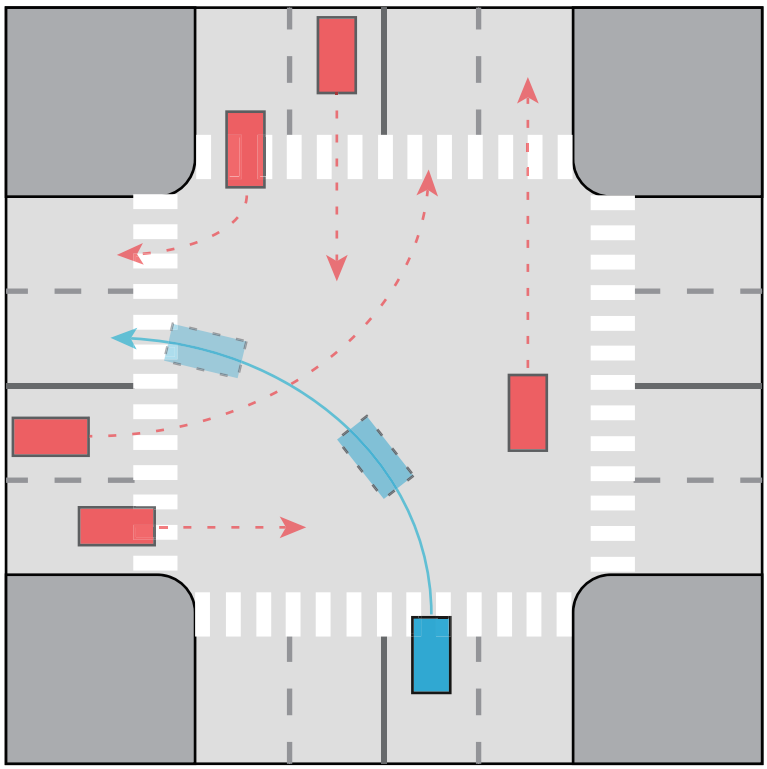

Figure 4 (Color online) Schematic diagram of motion planning task.

The action space is defined as $a = (a^{(0)})$, where $a^{(i)} = (\tau^{(i)}, \varphi^{(i)})$. Here, $\tau^{(i)} \in [-5, 5]$ denotes the acceleration in speed, while $\varphi^{(i)} \in [-\pi/6, \pi/6]$ represents the jerk in angle.

The nominal kinematics could be formulated as a control-affine scheme,

$$\begin{bmatrix} \dot{p}^{(i)} \\ \dot{v}^{(i)} \end{bmatrix} = \begin{bmatrix} 0 & I_{2\times 2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p^{(i)} \\ v^{(i)} \end{bmatrix} + \begin{bmatrix} 0 \\ I_{2\times 2} \end{bmatrix} u^{(i)}, \tag{37}$$

where $p^{(i)} = [x^{(i)}, y^{(i)}]^{\mathrm{T}}$, $v^{(i)} = [\rho^{(i)} \cos\phi^{(i)}, \rho^{(i)} \sin\phi^{(i)}]^{\mathrm{T}}$, and $u^{(i)} = [\tau^{(i)} \cos\varphi^{(i)}, \tau^{(i)} \sin\varphi^{(i)}]^{\mathrm{T}}$ stand for the position, velocity, and acceleration of agent i, respectively. Furthermore, the collective decision spaces of all agents are defined as $(p, v) \in \mathbb{R}^{4(N+1)}$ and $u \in \mathbb{R}^{2(N+1)}$.

The relative position and relative velocity between the ego agent and other agents (potential obstacles) are defined as $\Delta pi = p^{(i)} - p^{(0)}$ and $\Delta vi = v^{(i)} - v^{(0)}$. Then, we decompose these into normal and tangent components. As the tangent component only causes rotation around each other, we only need to consider the normal component $\bar{v} = |\dot{\Delta p}|$ to regulate the potential risk of collisions.

Suppose that take T to reach $\Delta\bar{v}(t_0 + T) = 0$ from the current time instance t_0. In this case, the following constraint must be satisfied to maintain a greater distance from the safe distance D_s:

$$\|\Delta p_i\| + \int_{t_0}^{t_0+T} \Delta\bar{v}\,(t_0 + t)\,\mathrm{d}t \geqslant D_s. \tag{38}$$

Assuming that $|u^{(0)}|\infty \leqslant \tau\mathrm{max}$, the constraint can be simplified using kinematic laws when the maximum braking acceleration is applied to the ego agent. Consequently, we can deduce the barrier certificate as

$$h_i(p, v) = \sqrt{2\tau_{\max}\left(\|\Delta p_i\| - D_s\right)} + \frac{\Delta p_i^{\mathrm{T}}}{\|\Delta p_i\|}\Delta v_i. \tag{39}$$

Considering $\alpha(h_i(p, v)) = \omega h_i(p, v)$, as defined in (4), we can derive the following inequalities that

Table 4 Comparative indices for motion planning task[a)]

Index	Penalized-PO	Projected-PO	Guided-PO
Pass times (s)	4.80	**4.32**	4.35
Successful rate (%)	75.4	96.9	**99.2**
Average acceleration (m/s^2)	1.28	**1.73**	1.65
Average jerk (rad)	0.16	0.14	**0.13**

a) Numbers in bold represent the best value achieved by each method under the corresponding index.

include the control variable:

$$\begin{aligned} -\Delta p_i^{\mathrm{T}} \Delta u_i \leqslant \omega h_i(p,v) \left\|\Delta p_i\right\| - \frac{(\Delta p_i^{\mathrm{T}} \Delta v_i)^2}{\left\|\Delta p_i\right\|^2} + \left\|\Delta v_i\right\|^2 \\ + \frac{\tau_{\max} \Delta p_i^{\mathrm{T}} \Delta v_i}{\sqrt{2]\tau_{\max}\left(\left\|\Delta p_i\right\| - D_s\right)}}, \quad i = 1, 2, \ldots, N. \end{aligned} \tag{40}$$

As $\Delta ui = u^{(i)} - u^{(0)}$, we only consider the input of the ego agent. When the state p, v is given, the inequality can be represented as a linear constraint on $u^{(0)}$ as $Ai \cdot u^{(0)} \leqslant b_i$, where $i = 1, 2, \ldots, N$, and A_i and b_i are defined as follows:

$$\begin{cases} A_i = \Delta p_i^{\mathrm{T}}, \\ b_i = \Delta p_i^{\mathrm{T}} u^{(i)} + \omega h_i(p,v) \left\|\Delta p_i\right\| - \dfrac{(\Delta p_i^{\mathrm{T}} \Delta v_i)^2}{\left\|\Delta p_i\right\|^2} + \left\|\Delta v_i\right\|^2 + \dfrac{\tau_{\max} \Delta p_i^{\mathrm{T}} \Delta v_i}{\sqrt{2\tau_{\max}\left(\left\|\Delta p_i\right\| - D_s\right)}}. \end{cases} \tag{41}$$

In this way, we can derive the constraints of the whole system as

$$\begin{aligned} S(u^{(0)}) = \{u^{(0)} \in \mathbb{R}^2 \mid A_i u^{(0)} \leqslant b_i,\ i = 1, 2, \ldots, N, \\ \bar{A}_j u^{(0)} \leqslant \bar{b}_j,\ j = 1, 2, \ldots, M\}. \end{aligned} \tag{42}$$

We can then formulate it as a constrained optimization problem and implement the safe projection as follows:

$$\begin{aligned} u^* = \underset{u \in \mathbb{R}^2}{\operatorname{argmin}}\ J(u) = \left\|u - \hat{u}\right\|^2 \\ \text{s.t.} \quad u \in S(u^{(0)}), \\ \left\|u\right\|_\infty \leqslant \tau_{\max}. \end{aligned} \tag{43}$$

We implemented a series of policy optimization algorithms (i.e., Penalized-PO, Projected-PO, and Guided-PO, the same as the ones in Subsection 5.3) in this case and trained 10 different instances of each algorithm with unknown patterns. The evaluation rollout was performed every 1000 time steps. We studied 100 cases toward different initial states and different types of noise to calculate the mean performance and then collected the track record of pass times, successful rate, average acceleration, and average jerk in Table 4.

The projection-based method outweighed the penalized one in all indices in terms of performance and safety. Furthermore, the safe projection with guided exploration performed a bit better in safety measurement with little sacrifice in speed performance, which was not a major concern in this case. Above all, through this case study, we showed the potential of the proposed method in leveraging effects in more general situations.

6 Conclusion

In this work, we proposed a safe projection method embedded with CBF to address the statewise constraints in constrained RL. We analyzed the convergence in a deterministic dynamic setting and the performance improvement in a stochastic dynamic setting. Theoretical results toward ensembles and guided resolution were extended. Experiments were conducted in different scenarios to present the effectiveness of the projected framework combined with different RL algorithms. The proposed method achieved superior results in terms of learning processes and performance and safety indices. In future work, some feasibility issues will be considered, and a more comprehensive study toward scalability is worth studying.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2022YFA1004600), National Natural Science Foundation of China (Grant Nos. 62125304, 62073182, 62192751), and Tsinghua University Initiative Scientific Research Program.

References

1 Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge: MIT Press, 2018
2 Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518: 529–533
3 Chen F Y, Liu M, Everett M, et al. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In: Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 2017. 285–292
4 Zhu J, Wei Y T, Kang Y, et al. Adaptive deep reinforcement learning for non-stationary environments. Sci China Inf Sci, 2022, 65: 202204
5 García J, Fernández F. A comprehensive survey on safe reinforcement learning. J Machine Learning Res, 2015, 16: 1437–1480
6 Achiam J, Held D, Tamar A, et al. Constrained policy optimization. In: Proceedings of the International Conference on Machine Learning, Sydney, 2017. 22–31
7 Cui R H, Xie X J. Adaptive state-feedback stabilization of state-constrained stochastic high-order nonlinear systems. Sci China Inf Sci, 2021, 64: 200203
8 Stooke A, Achiam J, Abbeel P. Responsive safety in reinforcement learning by pid lagrangian methods. In: Proceedings of the International Conference on Machine Learning, Virtual, 2020. 9133–9143
9 Amos B, Kolter Z. OptNet: differentiable optimization as a layer in neural networks. In: Proceedings of the International Conference on Machine Learning, Sydney, 2017. 136–145
10 Sidrane C, Maleki A, Irfan A, et al. OVERT: an algorithm for safety verification of neural network control policies for nonlinear systems. J Machine Learning Res, 2022, 23: 5090–5134
11 Wabersich K P, Zeilinger M N. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. Automatica, 2021, 129: 109597–109609
12 Gurriet T, Mote M, Singletary A, et al. A scalable safety critical control framework for nonlinear systems. IEEE Access, 2020, 8: 187249–187275
13 Ghaemi R, Vecchio D D. Control for safety specifications of systems with imperfect information on a partial order. IEEE Trans Automat Contr, 2014, 59: 982–995
14 Jiang Z Y, Jia Q-S, Guan X H. On large action space in EV charging scheduling optimization. Sci China Inf Sci, 2022, 65: 122201
15 Li J N, Nie H, Chai T Y, et al. Reinforcement learning for optimal tracking of large-scale systems with multitime scales. Sci China Inf Sci, 2023, 66: 170201
16 Fisac J F, Akametalu A K, Zeilinger M N, et al. A general safety framework for learning-based control in uncertain robotic systems. IEEE Trans Automat Contr, 2019, 64: 2737–2752
17 Liu L, Liu Y-J, Chen A Q, et al. Integral barrier Lyapunov function-based adaptive control for switched nonlinear systems. Sci China Inf Sci, 2020, 63: 132203
18 Zhu Z R, Chai Y, Yang Z M, et al. Safety criteria based on barrier function under the framework of boundedness for some dynamic systems. Sci China Inf Sci, 2022, 65: 122203
19 Brown D, Goo W, Nagarajan P, et al. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In: Proceedings of the International Conference on Machine Learning, Vienna, 2019. 783–792
20 Balakrishna A, Thananjeyan B, Lee J, et al. On-policy robot imitation learning from a converging supervisor. In: Proceedings of the Conference on Robot Learning, London, 2020. 24–41
21 Zanon M, Gros S. Safe reinforcement learning using robust MPC. IEEE Trans Automat Contr, 2021, 66: 3638–3652
22 Sadigh D, Kim S E, Coogan S, et al. A learning based approach to control synthesis of Markov decision processes for linear temporal logic specifications. In: Proceedings of the IEEE Conference on Decision and Control, Los Angeles, 2014. 1091–1096
23 Ames A D, Xu X, Grizzle J W, et al. Control barrier function based quadratic programs for safety critical systems. IEEE Trans Automat Contr, 2016, 62: 3861–3876
24 Taylor A, Singletary A, Yue Y, et al. Learning for safety-critical control with control barrier functions. In: Proceedings of the Learning for Dynamics and Control, California, 2020. 708–717
25 Wang L, Ames A D, Egerstedt M. Safety barrier certificates for collisions-free multirobot systems. IEEE Trans Robot, 2017, 33: 661–674
26 Romdlony M Z, Jayawardhana B. Stabilization with guaranteed safety using control Lyapunov-barrier function. Automatica, 2016, 66: 39–47
27 Cheng R, Orosz G, Murray M R, et al. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence, Hawaii, 2019. 3387–3395
28 Yang T Y, Rosca J, Narasimhan K, et al. Accelerating safe reinforcement learning with constraint-mismatched baseline policies. In: Proceedings of the International Conference on Machine Learning, Virtual, 2021. 11795–11807
29 Hasselt H. Double Q-learning. In: Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, 2010. 2613–2621
30 Singh S, Jaakkola T, Littman M L, et al. Convergence results for single-step on-policy reinforcement-learning algorithms. Machine Learn, 2000, 38: 287–308
31 Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization. In: Proceedings of the International Conference on Machine Learning, Lille, 2015. 1889–1897
32 Jin X Z, Jia Q S, Zhang T, et al. Learning-based safety-critical motion planning with input-to-state barrier certificate. In: Proceedings of the International Conference on Automation Science and Engineering, Lyon, 2021. 1967–1972

Appendix A Proof of Lemma 1

Proof of Lemma 1. Derived from the Bellman equation, the optimal state-value function satisfies

$$V^*(s) = \max_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} \mathcal{P}_a(s,s') \left[r(s,a,s') + \gamma V^*(s')\right], \tag{A1}$$

and the optimal action-value function satisfies

$$\begin{aligned} Q^*(s,a) &= \sum_{s'\in\mathcal{S}} \mathcal{P}_a(s,s') \left[r(s,a,s') + \gamma V^*(s')\right] \\ &= \sum_{s'\in\mathcal{S}} \mathcal{P}_a(s,s') \left[r(s,a,s') + \gamma \max_{a'\in\mathcal{A}} Q^*(s',a')\right]. \end{aligned} \tag{A2}$$

Then the operator $\mathcal{T}$ over the generic function $q : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ with respect to (A2) can be defined as

$$(\mathcal{T}q)(s,a) = \sum_{s'\in\mathcal{S}} \mathcal{P}_a(s,s') \left[r(s,a,s') + \gamma \max_{a'\in\mathcal{A}} q(s',a')\right]. \tag{A3}$$

We can conclude that $\mathcal{T}$ is a contraction in uniform norm through the following derivations:

$$\begin{aligned} \left\|\mathcal{T}q_{t+1} - \mathcal{T}q_t\right\|_\infty =& \max_{s,a} \left| \sum_{y\in\mathcal{S}} \mathcal{P}_a(s,s') \left[\gamma \max_{a'\in\mathcal{A}} q_{t+1}(s',a') - \gamma \max_{a'\in\mathcal{A}} q_t(s',a')\right]\right| \\ =& \max_{s,a} \gamma \left| \sum_{s'\in\mathcal{S}} \mathcal{P}_a(s,s') \left[\max_{a'\in\mathcal{A}} q_{t+1}(s',a') - \max_{a'\in\mathcal{A}} q_t(s',a')\right]\right| \\ \leqslant& \max_{s,a} \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}_a(s,s') \left|\max_{a'\in\mathcal{A}} q_{t+1}(s',a') - \max_{a'\in\mathcal{A}} q_t(s',a')\right| \\ \leqslant& \max_{s,a} \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}_a(s,s') \max_{\tilde{s},a'} \left|q_{t+1}(\tilde{s},a') - q_t(\tilde{s},a')\right| \\ =& \max_{s,a} \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}_a(x,y) \left\|q_{t+1} - q_t\right\|_\infty \\ =& \gamma \left\|q_{t+1} - q_t\right\|_\infty. \end{aligned} \tag{A4}$$

Since $\mathcal{T}q^* = q^*$, then we have $\left\|\mathcal{T}q_t - q^*\right\|_\infty = \left\|\mathcal{T}q_t - \mathcal{T}q^*\right\|_\infty \leqslant \gamma \left\|q_t - q^*\right\|_\infty$. Hence, as t grows, q_t will eventually converge to q^* at a linear rate of γ.

Appendix B Proof of Lemma 3

Proof of Lemma 3. Let α denote the maximum total variation divergence between stochastic policies π and $\tilde{\pi}$, defined as $D_{\mathrm{TV}}^{\max}(\pi,\tilde{\pi}) = \max_s \frac{1}{2}\sum_i |\pi(\cdot|s)_i - \tilde{\pi}(\cdot|s)_i|$. Then we can conclude a bound in policy update towards the approximation of the true expected reward objective J as

$$J(\tilde{\pi}) \geqslant L(\tilde{\pi}) - \alpha^2 \frac{4\gamma\epsilon}{(1-\gamma)^2}, \tag{B1}$$

where $\epsilon = \max_{s,a} |A^\pi(s,a)|$ and $\gamma\in[0,1)$ is the discount factor.

Proof. Let $G = 1 + \gamma P_\pi + (\gamma P_\pi)^2 + \cdots = (1-\gamma P_\pi)^{-1}$, and similarly $\tilde{G} = (1-\gamma P_{\tilde{\pi}})^{-1}$. Note that $J(\pi) = rGd_0$ and $J(\tilde{\pi}) = r\tilde{G}d_0$, where rd is a scalar meaning the expected reward under density d.

Let $\Delta = P_{\tilde{\pi}} - P_\pi$ and start with some standard perturbation theory manipulations,

$$\begin{aligned} G^{-1} - \tilde{G}^{-1} &= (1-\gamma P_\pi) - (1-\gamma P_{\tilde{\pi}}) = \gamma\Delta \\ \Longrightarrow \tilde{G} - G &= \gamma G\Delta\tilde{G}. \end{aligned} \tag{B2}$$

Then we have

$$\tilde{G} = G + \gamma G\Delta\tilde{G} = G + \gamma G\Delta G + \gamma^2 G\Delta G\Delta\tilde{G}. \tag{B3}$$

The goal is to bound

$$J(\tilde{\pi}) - J(\pi) = r(\tilde{G} - G)d_0 = \gamma rG\Delta Gd_0 + \gamma^2 rG\Delta G\Delta\tilde{G}d_0. \tag{B4}$$

It is observed that $rG = v$ denotes the infinite-horizon cost-to-go function, and $Gd_0 = d_\pi$ denotes the distribution of the current state. Thus, we can write the first term in (B4) as $\gamma rG\Delta Gd_0 = \gamma v\Delta d_\pi$. The following derivations in (B5) show that it equals the expected advantage $L(\tilde{\pi}) - L(\pi)$.

$$\begin{aligned} &L(\tilde{\pi}) - L(\pi) \\ &= \sum_s d_\pi(s) \sum_a (\tilde{\pi}(a|s) - \pi(a|s)) A^\pi(s,a) \\ &= \sum_s d_\pi(s) \sum_a (\tilde{\pi}(a|s) - \pi(a|s)) \cdot \left[r(s) + \sum_{s'} p\left(s'|s,a\right)\gamma v\left(s'\right) - v(s)\right] \\ &= \sum_s d_\pi(s) \sum_{s'} \sum_a (\tilde{\pi}(a|s) - \pi(a|s)) p\left(s'|s,a\right)\gamma v\left(s'\right) \\ &= \sum_s d_\pi(s) \sum_{s'} \left(p_{\tilde{\pi}}\left(s'|s\right) - p_\pi\left(s'|s\right)\right)\gamma v\left(s'\right) \\ &= \gamma v\Delta d_\pi. \end{aligned} \tag{B5}$$

For the second term $\gamma^2 rG\Delta G\Delta\tilde{G}d$ in (B4), we separate it into two parts as $\gamma rG\Delta = \gamma v\Delta$ and $G\Delta\tilde{G}d$.

For the first item, consider the component s of the dual vector

$$\begin{aligned}|(\gamma v\Delta)_s| &= \left|\sum_a(\tilde{\pi}(a|s)-\pi(a|s))A^{\pi}(s,a)\right| \\ &= \left|\sum_a(\tilde{\pi}(a|s)-\pi(a|s))Q^{\pi}(s,a)\right| \\ &\leqslant \sum_a|\tilde{\pi}(a|s)-\pi(a|s)|\cdot\max_a|A^{\pi}(s,a)| \\ &\leqslant 2\alpha\cdot\epsilon.\end{aligned} \tag{B6}$$

Then, for the second item, take the ℓ_1 operator norm $\|A\|_1=\sup_d\{\frac{\|Ad\|_1}{\|d\|_1}\}$ into account, where we have $\|G\|_1=\|\tilde{G}\|_1=1/(1-\gamma)$ and $\|\Delta\|_1=2\alpha$. That gives

$$\begin{aligned}\|G\Delta\tilde{G}d\|_1 &\leqslant \|G\|_1\|\Delta\|_1\|\tilde{G}\|_1\|d\|_1 \\ &= \frac{1}{1-\gamma}\cdot 2\alpha\cdot\frac{1}{1-\gamma}\cdot 1.\end{aligned} \tag{B7}$$

Combining (B6) and (B7), we have that

$$\begin{aligned}\gamma^2 rG\Delta G\Delta\tilde{G}d &\leqslant \gamma\|\gamma rG\Delta\|_{\infty}\|G\Delta\tilde{G}d\|_1 \\ &\leqslant \gamma\cdot 2\alpha\epsilon\cdot\frac{2\alpha}{(1-\gamma)^2} \\ &= \alpha^2\frac{4\gamma\epsilon}{(1-\gamma)^2}.\end{aligned} \tag{B8}$$

Substituting π into (23), we could observe that

$$\begin{aligned}L(\pi) &= J(\pi)+\sum_s d_{\pi}(s)\sum_a\pi(a|s)A^{\pi}(s,a) \\ &= J(\pi)+\sum_s d_{\pi}(s)\sum_a\pi(a|s)(Q^{\pi}(s,a)-V_{\pi}(s)) \\ &= J(\pi)+\sum_s d_{\pi}(s)\sum_a\pi(a|s)Q^{\pi}(s,a)-\sum_s d_{\pi}(s)V_{\pi}(s)\sum_a\pi(a|s) \\ &= J(\pi)+\sum_s d_{\pi}(s)\left(\sum_a\pi(a|s)Q^{\pi}(s,a)-V_{\pi}(s)\right) \\ &= J(\pi).\end{aligned} \tag{B9}$$

Since $J(\pi)=L(\pi)$, moving the item of (B5) into (B4), we could conclude that $|J(\tilde{\pi})-L(\tilde{\pi})|\leqslant\alpha^2\frac{4\gamma\epsilon}{(1-\gamma)^2}$ according to (B8); then Eq. (24) is followed.

Appendix C Analytical solution for stochastic dynamic

Within a small step size, we could approximate the reward function and constraints with a first order expansion, and measure the KL divergence with a second order expansion.

Define $\mu_1:=\nabla_{\theta}\mathbb{E}_{\substack{s\sim d^{\pi^k}\\a\sim\pi}}[A^{\pi^k}(s,a)]$ as the gradient of the reward advantage function, $\mu_2:=\nabla_{\theta}\mathbb{E}_{\substack{s\sim d^{\pi^k}\\a\sim\pi}}[L_fh(s)+L_gh(s)a]$ as the gradient of the control barrier function, and $H:=\frac{\partial^2\mathbb{E}_{s\sim d^{\pi^k}}[D_{\mathrm{KL}}(\pi,\pi^k)]}{\partial\theta_i\theta_j}$ as the Hessian matrix of the KL divergence, where θ is the parameter of the policy.

Through this way, we could reparametrize the formulation in Subsection 3.2. In the first step, we optimize the expected reward as

$$\begin{aligned}&\theta^{k+\frac{1}{2}}=\arg\max_{\theta}\ \mu_1^{\mathrm{T}}\left(\theta-\theta^k\right) \\ &\text{s.t. }\frac{1}{2}\left(\theta-\theta^k\right)^{\mathrm{T}}H\left(\theta-\theta^k\right)\leqslant\delta;\end{aligned} \tag{C1}$$

then in the second step, we project the policy within the safe set as

$$\begin{aligned}&\theta^{k+1}=\arg\min_{\theta}\ \frac{1}{2}\left(\theta-\theta^{k+\frac{1}{2}}\right)^{\mathrm{T}}H\left(\theta-\theta^{k+\frac{1}{2}}\right) \\ &\text{s.t. }\mu_2^{\mathrm{T}}\left(\theta-\theta^k\right)\geqslant 0,\end{aligned} \tag{C2}$$

where $\mu_1,\mu_2\in\mathbb{R}^n$, $\delta>0\in\mathbb{R}$, and $H\in\mathbb{R}^{n\times n}$ is symmetric positive semi-definite.

The first step is a second-order cone programming. Let θ_1^* and λ_1^* denote the solutions to the primal problem and dual problem in (C1). We could form the Lagrangian as

$$L(\theta_1,\lambda_1)=-\mu_1^{\mathrm{T}}\left(\theta_1-\theta^k\right)+\lambda_1\left(\left(\theta_1-\theta^k\right)^{\mathrm{T}}H\left(\theta_1-\theta^k\right)-\delta\right). \tag{C3}$$

The KKT conditions are derived as

$$\begin{cases} -\mu_1 + \lambda_1^* H(\theta_1^* - \theta^k) = 0, \\ \frac{1}{2}(\theta_1^* - \theta^k)^{\mathrm{T}} H(\theta_1^* - \theta^k) - \delta = 0, \end{cases} \\ \Longrightarrow \begin{cases} \theta_1^* = \theta^k + \frac{1}{\lambda_1^*} H^{-1}\mu_1, \\ \lambda_1^* = \sqrt{\frac{\mu_1^{\mathrm{T}} H^{-1}\mu_1}{2\delta}}. \end{cases} \tag{C4}$$

Hence, the solution of (C1) is

$$\theta^{k+\frac{1}{2}} = \theta_1^* = \theta^k + \sqrt{\frac{\mu_1^{\mathrm{T}} H^{-1}\mu_1}{2\delta}} H^{-1}\mu_1. \tag{C5}$$

The second step is a quadratic programming. Let θ_2^* and λ_2^* denote the solutions to the primal problem and dual problem in (C2). We could form the Lagrangian as

$$L(\theta_2, \lambda_2) = \frac{1}{2}\left(\theta - \theta^{k+\frac{1}{2}}\right)^{\mathrm{T}} H\left(\theta - \theta^{k+\frac{1}{2}}\right) - \lambda_2 \mu_2^{\mathrm{T}}\left(\theta - \theta^k\right). \tag{C6}$$

The KKT conditions are derived as

$$\begin{cases} H\left(\theta_2^* - \theta^{k+\frac{1}{2}}\right) - \lambda_2^*\mu_2 = 0, \\ \mu_2^{\mathrm{T}}\left(\theta_2^* - \theta^k\right) = 0, \\ \lambda_2^* \geqslant 0, \end{cases} \\ \Longrightarrow \begin{cases} \theta_2^* = \theta^{k+\frac{1}{2}} - \lambda_2^* H^{-1}\mu_2, \\ \lambda_2^* = \max\left(0, \frac{\mu_2^{\mathrm{T}}(\theta^{k+\frac{1}{2}} - \theta^k)}{\mu_2^{\mathrm{T}} H^{-1}\mu_2}\right). \end{cases} \tag{C7}$$

Hence, the solution of (C2) is

$$\begin{aligned} \theta^{k+1} = \theta_2^* &= \theta^{k+\frac{1}{2}} - \max\left(0, \frac{\mu_2^{\mathrm{T}}(\theta^{k+\frac{1}{2}} - \theta^k)}{\mu_2^{\mathrm{T}} H^{-1}\mu_2}\right) H^{-1}\mu_2 \\ &= \theta^k + \sqrt{\frac{2\delta}{{\mu_1}^{\mathrm{T}} H^{-1}\mu_1}} H^{-1}\mu_1 \\ &\quad + \max\left(0, \frac{-\sqrt{\frac{2\delta}{{\mu_1}^{\mathrm{T}} H^{-1}\mu_1}}{\mu_2}^{\mathrm{T}} H^{-1}\mu_1}{{\mu_2}^{\mathrm{T}} H^{-1}\mu_2}\right) H^{-1}\mu_2. \end{aligned} \tag{C8}$$