Safe Exploration of Reinforcement Learning with Data-Driven Control Barrier Function

Chenlin Zhang[1], Shaochen Wang [1], Shaofeng Meng[1], and Zhen Kan [1,*]

2022 China Automation Congress (CAC) | 978-1-6654-6533-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/CAC57257.2022.10055848

***Abstract*—Reinforcement learning relies on exploration and exploitation to find optimal policies. However, unconstrained exploration might lead to unsafe actions that jeopardize the system safety. To address this issue, this work presents a RL-based framework that integrates model-based CBF to ensure safe exploration during learning. Rather than synthesizing CBF by hand for complex dynamic systems, we exploit data-driven methods to learn CBFs from collected demonstrations of safe and desirable behavior. Unlike prior works that restrict on off-line collected expert demonstrations to train CBF, the CBF in this work is learned not only from preliminary expert demonstrations, but also from the on-line generated data at runtime, resulting in improved adaptation to complex environments. Numerical simulations and physical experiments using Crazyflie quadrotors are carried out to demonstrate the effectiveness of the developed safe RL framework. The experiment video is available at https://youtu.be/uscl-BQsLRo.**

***Index Terms*—Control barrier function, reinforcement learning, shield, data-driven.**

I. Introduction

Reinforcement learning (RL) is a sequential decision making process in which agents learn effective policies by interacting with the environment [1]–[4]. To search for optimal policies, a key enabling technology in RL is exploration, which allows the learning agent to explore unvisited states and try new actions to find the most rewarding policy. However, unconstrained exploration might lead to unsafe actions that jeopardize the system safety. For example, when training UAVs for autonomous navigation missions, the UAVs may collide with obstacles or other UAVs during exploration. Therefore, safe exploration plays a vital role in RL to ensure system security.

When considering safe exploration in RL, earlier works considered the worst-case criterion and replaced the maximization of the cumulative rewards with the maximization of the minimum of possible rewards (i.e., choosing the best of the bad states) [5]. However, such approaches do not guarantee the optimality of the strategy. In [6], a conservative update strategy was developed to maximize the reward function with penalty constraints, which only ensures local satisfaction. Recently, shield-based approaches were developed to ensure system security [7]–[9], which monitors the learned actions and replaces them with safe ones if the chosen action violates the safety constraints. While conceptually appealing, they generally suffer from real-time implementation issues.

[1]Department of Automation, University of Science and Technology of China, Hefei, Anhui, China. (zhangchenlin@mail.ustc.edu.cn)

*Corresponding Author (zkan@ustc.edu.cn)

An alternative promising approach to ensure system security is based on control barrier functions (CBF) [10]–[13]. However, designing a qualified CBF is challenging and often requires great effort to hand-design certificates for a specific system. To address this issue, data-driven methods that model CBF as a neural network have shown significant progress. For instance, the CBF is parameterized by the support vector machine and the state space is characterized as either safe or unsafe based on the collected sensor data via supervised learning [14]. However, such an approach cannot guarantee in advance the existence of control actions such that the learned safe set is forward invariant. In [15], a deep neural network was trained via imitation learning to replicate a CBF-based controller. In [16], a learning-enabled perception-feedback hybrid controller was developed and in [17] safe control policies and barrier certificates were jointly learned to avoid collisions with static obstacles and other agents while reaching their goals. While most aforementioned works are empirically validated, no formal guarantees of safety were established. One exception is the work of [18], in which an optimization based approach was developed to learn CBF from expert trajectories with provable safety guarantees.

This work presents a RL-based framework that integrates model-based CBF to ensure safe exploration during learning. Since synthesizing CBF by hand for complex dynamic systems is very challenging, we exploit data-driven methods to learn CBF from collected demonstrations of safe and desirable behavior. Compared with the literature, the contributions are summarized as follows. First, we incorporate CBF with RL algorithms to enable safe exploration in a data-driven fashion, which leverages the power of RL algorithms in learning high-performance controllers without requiring the complete knowledge about the system environment while utilizing CBF as safety certificates to guide the learning process by constraining the set of explorable polices. Second, rather than synthesizing CBF by hand as in many works, we consider online learning of CBF, which is synthesized with RL to guarantee safety and demonstrates great policy exploration efficiency. In addition, unlike [18] that restricts on off-line collected expert demonstrations to train CBF, the CBF in this work is learned not only from preliminary expert demonstrations, but also from the on-line generated data during runtime, resulting in improved adaptation to complex environments. We also develop a look-ahead and proactive approach, namely predictive shielding control, to account for the uncertainty of NNs to further ensure system safety. Moreover, numerical simulations and physical experiments using Crazyflie quadrotors are carried out and

demonstrate that our method significantly outperforms other leading methods.

II. Preliminaries

A. Control Barrier Function

Consider an affine dynamical system

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \quad x(0) \in \mathbb{R}^n \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are known locally Lipschitz continuous nonlinear functions, and $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ represent the system state and system input, respectively. The system in (1) is considered as safe if the states are restricted within the set

$$\mathcal{C} \coloneqq \{x \in \mathbb{R}^n \mid h(x) \geq 0\}, \tag{2}$$

where $h\colon \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable function. That is, the set $\mathcal{C}$ satisfies safety specifications and can be made forward invariant with appropriate control input. The boundary and the interior of the set are denoted by $\mathrm{bd}(\mathcal{C})$, $\mathrm{int}(\mathcal{C})$, respectively.

Definition 1. Let $\mathcal{D}$ be an open set such that $\mathcal{D} \supset \mathcal{C}$. The function $h(x)$ is a valid CBF on $\mathcal{D}$ if there exists a locally Lipschitz continuous extended class $\mathcal{K}$ function $\alpha\colon \mathbb{R} \to \mathbb{R}$ such that for all $x \in \mathcal{D}$

$$\sup_{u \in \mathcal{U}} \{L_f h(x) + L_g h(x) u(x) + \alpha(h(x))\} \geq 0, \tag{3}$$

where $L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$ and $L_g h(x) = \frac{\partial h(x)}{\partial x} g(x)$, and $\mathcal{U} \in \mathbb{R}^m$ defines the constrains on the control input u. Therefore, a set of CBF consistent inputs induced by a valid CBF is as

$$\mathcal{V}(x) \coloneqq \{u \in \mathbb{R}^m \mid L_f h(x) + L_g h(x) u(x) + \alpha(h(x)) \geq 0\}. \tag{4}$$

The following lemma from [11] indicates that the system safety can be guaranteed with appropriate control input $u(x) \in \mathcal{V}(x)$ for all $x \in \mathcal{D}$.

Lemma 2. *Assume that $h(x)$ is a valid control barrier function on $\mathcal{D}$ and $u\colon \mathcal{D} \to \mathcal{U}$ with $u(x) \in \mathcal{V}(x)$ is locally Lipschitz continuous. Then, it holds that $x(0) \in \mathcal{C}$ implies $x(t) \in \mathcal{C}$ for all $t \geq 0$.*

B. Reinforcement Learning

As a sequential decision-making process, RL is often modeled as a Markov decision process (MDP) $\mathcal{M} = (S, A, \mathcal{T}, R, \gamma)$ with state space S and action space A. A policy π maps a state to an action and generates the next state according to the state-action transition function $\mathcal{T} : S \times A \mapsto S$. The reward $r : S \times A \mapsto \mathbb{R}$ is then received, where $r \in \mathbb{R}$. By defining the expected cumulative reward as $Q_\pi = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$ with the discount factor $\gamma \in [0, 1]$, the goal of the agent is to learn an optimal policy $\pi^* = \underset{\pi}{argmax} Q_\pi$.

III. Problem Formulation

To ensure safe exploration, this work adopts CBF to constrain the agent actions. To learn a valid CBF from data in the form of (3), we first distinguish the geometric safe zone $\mathcal{Z}$ and the safe set $\mathcal{C}$, where $\mathcal{Z}$ is specified directly from the system state space and $\mathcal{C}$ is defined as in (2).

Suppose there are expert trajectories consisting of N discretized state-action pairs $\mathcal{P}_{exp} \coloneqq \{(x_i, u_i)_{i=1}^N\}$ with $x_i \in \mathrm{int}(\mathcal{Z})$. For $\epsilon > 0$, we define the sets

$$\mathcal{D}' \coloneqq \bigcup_{i=1}^{N} \mathcal{B}_{\epsilon,p}(x_i) \quad \text{and} \quad \mathcal{D} \coloneqq \mathcal{D}' \setminus \mathrm{bd}(\mathcal{D}'), \tag{5}$$

where $\mathcal{B}_{\epsilon,p}(x_i) \coloneqq \{x \in \mathbb{R}^n \mid\mid\mid x - x_i \mid\mid_p \leq \epsilon\}$ denotes the closed p-norm ball around $x_i \in \mathbb{R}^n$ with radius ϵ. Note that $\mathcal{D}$ is obtained from the expert trajectory and the conditions on ϵ will be specified later to ensure the validity of the learned CBF. Define the set

$$\mathcal{L} \coloneqq \{\mathrm{bd}(\mathcal{D}) \oplus \mathcal{B}_{\sigma,p}(0)\} \setminus \mathcal{D}$$

where $\sigma > 0$ and $\oplus$ is the Minkowski sum[1]. Geometrically, $\mathcal{L}$ represents a layer with width σ around $\mathcal{D}$. By enforcing the value of the learned CBF is negative on $\mathcal{L}$, the zero-level set $\{x \in \mathbb{R}^n \mid h(x) = 0\}$ is guaranteed to be contained within $\mathcal{D}$. The motivation to enforce that $\mathcal{C} \subset \mathcal{D}$ is based on the analysis in [11]. To facilitate the learning of CBF from given data, we redefine $\mathcal{C}$ as

$$\mathcal{C} \coloneqq \{x \in \mathcal{L} \cup \mathcal{D} \mid h(x) \geq 0\}. \tag{6}$$

Therefore, the CBF can be learned by only sample data from the domain $\mathcal{L} \cup \mathcal{D}$. To ensure the correctness of the learned CBF, as discussed in [18], these sets have to satisfy $\mathcal{C} \subset \mathcal{D} \subseteq \mathcal{Z}$, which are visualized in Fig. 1. Although the theoretically provable CBF can be learned in [18], it relies on off-line collected expert demonstrations. New challenges arise when integrating with RL, since the CBF can only be gradually learned at runtime.

Objective: Denote by $\mathcal{P}_{dem} = \left\{(x_i, u_i)_{i=1}^M\right\}$ the preliminary off-line collected (probably a small amount of) expert demonstrations. Since the CBF learned from $\mathcal{P}_{dem}$ only cannot ensure full system security, the goal of this work is to exploit data-driven methods to further improve the learning of CBF using the on-line generated data at runtime to ensure safe exploration in RL.

IV. Data-driven CBF-Based Safe Learning Framework

This section presents the developed data-driven CBF-based safe RL. The approach overview is shown in Fig. 2.

[1]For two sets $\mathcal{D}_1$ and $\mathcal{D}_2$, the Minkowski sum is defined as $\mathcal{D}_1 \oplus \mathcal{D}_2 \coloneqq \{x_1 + x_2 \in \mathbb{R}^n \mid x_1 \in \mathcal{D}_1, x_2 \in \mathcal{D}_2\}$.

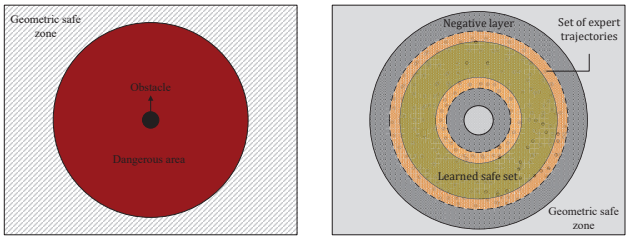


Fig. 1: The left diagram represents the geometric safe zone $\mathcal{Z}$, where the black dot represents the obstacle and the red disk area is the dangerous area. The area outside the dangerous area is the safe zone. The right diagram shows the relationship between $\mathcal{Z}$ and the learned safe set $\mathcal{C}$ (green ring), where the grey rings indicate the set $\mathcal{L}$ of width σ, and the orange ring indicates the set $\mathcal{D}$.

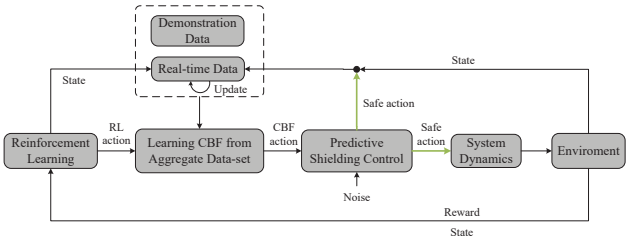


Fig. 2: The framework of the data-driven CBF based safe RL.

A. Reward Design for DRL

To elaborate the CBF-based safe RL framework, we consider a navigation task as a running example throughout this work, where the agents (e.g., quadrotors) are required to navigate to their own destinations while staying close (i.e., maintaining network connectivity for reliable inter-agent information exchange) and avoiding collision during flight. Such a problem is challenging as the constraint of network connectivity requires the agents do not move far away from others while the constraint of collision avoidance requires them do not stay too close. Let (x_i, y_i) and Θ_i be the position and orientation of agent i. The control actions are the angular and translational velocities, which are denoted by ω_i and v_i, respectively. With ω_i and v_i, we can obtain the agent's direction of motion and travel distance, i.e., its position and orientation at the next moment.

Due to the consideration of continuous state and action spaces, the actor-critic framework is employed in this work, in which the critic network is parameterized by θ and the actor network is parameterized by ϕ. The target networks are parameterized by θ' and ϕ' accordingly. The control action a is selected according to a policy network π_ϕ. After applying a, the reward r and new state s' are obtained. The transition tuple (s, a, r, s') is stored in the replay buffer.

The deep Q learning is treated as a regression problem to minimize the objective of

$$\mathbb{E}_{s,a\sim\mathcal{D}}\left[\left(y - Q_\theta\left(s,a\right)\right)^2\right],$$

where y is the target Q network. Since using y can lead to the overestimation of the Q value which can negatively affect the policy, Twin Delayed Deep Deterministic Policy Gradient algorithm (TD3) from [19] is adapted in this work. Specifically, two sets of critic networks (Twin) are used to represent Q_1 and Q_2 values, which are parameterized by θ_1 and θ_2, respectively. The target networks are parameterized by θ_1' and θ_2', respectively, with the relatively smaller one serving as the target Q network, i.e.,

$$y = r + \gamma \min_{i=1,2} Q_{\theta_i'}(s', \tilde{a}).$$

To enable adequate exploration and smoothing regularization, clipped Gaussian noise is added to the action as

$$\tilde{a} = \pi_{\phi'}(s') + \xi, \qquad \xi \sim clip\left(\mathcal{N}\left(0, \tilde{\sigma}\right), -c, c\right).$$

Randomly sampling mini-batch of T transitions from the replay buffer, the critics are updated by following

$$\theta_i = argmin_{\theta_i} \frac{1}{T} \sum \left(y - Q_{\theta_i}\left(s,a\right)\right)^2, i = 1, 2.$$

The policy is optimized by following the gradient

$$\nabla_a Q_{\theta_1}\left(s,a\right)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi\left(s\right).$$

It is well known that the design of the reward function is particularly important in RL. Next, we present the reward design in RL for the considered navigation task. The goal of the reward design is to encourage the agents to reach their destinations. The constraint of network connectivity and collision avoidance will be handled by the learned CBF in the sequel. The reward function is designed as

$$r = \begin{cases} -c_n & \exists i,\ d_i \geq B \\ c_p / \sum_{i=1}^{N} d_i & \forall i,\ d_i < B \text{ and } \exists i,\ d_i > \delta \\ r_{goal} - c_t \cdot t_{step} & \forall i,\ d_i \leq \delta \end{cases}$$

where d_i denotes the distance of agent i to its destination, $i \in \{1, ..., N\}$, and B represents the workspace boundary. If agent i moves beyond the workspace boundary B, the penalty $-c_n < 0$ will be received. The term $c_p / \sum_{i=1}^{N} d_i$ is designed to encourage the agents to move toward its destination, where $c_p > 0$ is a positive constant and $\delta > 0$ represents the proximity of the destination. Once the agent is sufficient close to its destination (i.e., $d_i \leq \delta$), a large positive reward $r_{goal} > 0$ will be received. The term $c_t \cdot t_{step}$ is included to encourage fast approach to the destination, where t_{step} is the number of steps to complete the navigation task and $c_t > 0$ is a tunable coefficient. With the designed reward function r, TD3 algorithm is used to solve this problem.

B. Data-set Aggregation for Training CBF

Unlike [18] that uses off-line collected $\mathcal{P}_{exp}$ to train the CBF, this work develops an on-line training strategy that exploits the data generated at runtime to continuously update the learned CBF, which in turn further refines the action

selection for safe exploration. This section presents how $\mathcal{P}_{exp}$ can be constructed online to facilitate the learning of CBF.

Let X_{RL} be a data-set that stores on-line collected state-action pairs at runtime. To avoid an oversize data-set, X_{RL} only stores the most recent data for a given period. Among the collected data in X_{RL}, we create a data-set $\mathcal{P}_{RL}$ by only selecting the data that demonstrates how to move away from unsafe set. The state-action pairs near workspace boundary B or safe pairs in X_{RL}, i.e., far away from the destinations or obstacles, are of little interest, as they are not quite helpful in improving the learning of CBF. Hence, $\mathcal{P}_{RL}$ is continuously updated based on the on-line collected X_{RL}. By initializing $\mathcal{P}_{exp} = \mathcal{P}_{dem}$, the expert data-set can then be continuously updated as $\mathcal{P}_{exp} = \mathcal{P}_{dem} \cup \mathcal{P}_{RL}$ by periodically enriching $\mathcal{P}_{dem}$ with the new data $\mathcal{P}_{RL}$. Using the data in $\mathcal{P}_{exp}$ the CBF will be gradually learned.

It is worth pointing out that, when using $\mathcal{P}_{exp}$ to train the CBF, the data has to be carefully selected. By defining a data-set $X_{safe} = \{x_i : (x_i, u_i) \in \mathcal{P}_{exp}\}$, the idea behind is to select data that demonstrates how to move away from the unsafe set. To this end, we select data such that the vector field $f(x_i, u_i)$ is relatively parallel to the inward pointing normal $\nabla h(x_i)$, (i.e., the vector field $f(x_i, u_i)$ is transverse to the level sets of $h(x_i)$). Such a data selection can result in a large inner-product term in the constraint of

$$\langle \nabla h(x_i), f(x_i) + g(x_i)u_i \rangle \geq -\alpha(h(x_i)) + \gamma_{exp}, \quad x_i \in X_{safe},$$

which will facilitate the learning of the CBF $h(x_i)$. In the next section, a series of conditions, as well as the positive constant γ_{exp}, will be derived to constrain the CBF to ensure that the learned CBF is locally valid on $\mathcal{D}$.

C. Learning CBF

Inspired by [18], this section presents an optimization based approach to learn the control barrier functions $h(x)$ from periodically updated $\mathcal{P}_{exp}$. To this end, let $X_{\mathcal{L}} = \{x_i\}_{i=1}^{M}$ be a data-set sampled from the set $\mathcal{L}$. Since the points in $\mathcal{L}$ are not generated by the expert, no inputs are associated with the samples in $X_{\mathcal{L}}$. Given a twice continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, its local Lipschitz bound is

$$L_h(x) \coloneqq \sup_{x_1, x_2 \in \mathcal{B}_{\epsilon,p}(x)} \frac{| h(x_1) - h(x_2) |}{\| x_1 - x_2 \|_p},$$

which can be efficiently estimated by post-hoc sampling.

The CBF $h(x)$ can be learned from data by solving the following optimization problem:

$$\begin{aligned}
&\min \| h \| \\
&\text{s.t.} \quad h(x_i) \geq \gamma_{safe}, \quad \forall x_i \in \bar{X}_{safe}(L_h), \\
&\qquad h(x_i) \leq -\gamma_{unsafe}, \\
&\qquad Lip(h(x_i), \bar{\epsilon}) \leq L_h, \quad \forall x_i \in X_{\mathcal{L}}, \\
&\qquad q(x_i, u_i) \coloneqq L_f h(x_i) + L_g h(x_i) u_i + \alpha(h(x_i)) \geq \gamma_{exp}, \\
&\qquad Lip(q(x_i, u_i), \epsilon) \leq L_q, \quad \forall (x_i, u_i) \in \mathcal{P}_{exp}
\end{aligned} \tag{7}$$

where the constraints in (7) are sufficient conditions to ensure that $h(x)$ is a valid local CBF on $\mathcal{D}$. The hyperparameters γ_{safe}, γ_{unsafe}, γ_{exp}, L_h and L_q are positive constants determined in the sequel using the data from the data-sets X_{safe} and $X_{\mathcal{L}}$.

Intuitively, the learned CBF should satisfy

$$h(x_i) \geq \gamma_{safe} \quad \forall x_i \in X_{safe}, \tag{8}$$

where $\gamma_{safe} > 0$ (i.e., the set $\mathcal{C}$ defined by (2) has non-empty interior), and satisfies

$$h(x_i) \leq -\gamma_{unsafe} \quad \forall x_i \in X_{\mathcal{L}} \tag{9}$$

with $\gamma_{unsafe} > 0$, where $X_{\mathcal{L}}$ is an $\bar{\epsilon}$-net of $\mathcal{L}$ with $\bar{\epsilon} < \gamma_{unsafe}/L_h(x_i)$ for all $x_i \in X_{\mathcal{L}}$. The constraints in (8) and (9) together ensure that $h(x) > 0$ for all $x \in \mathcal{C}$ and $h(x) < 0$ for all $x \in \mathcal{L}$, which indicates that $\mathcal{C} \subset \mathcal{D} \subseteq \mathcal{Z}$. However, as discussed in [18], if the Lipschitz bound L_h is low, it is difficult for $h(x)$ to vary from γ_{safe} to γ_{unsafe} at a very short distance $\bar{\epsilon}$. Therefore, by defining $L_h \coloneqq sup_{x_i \in X_{\mathcal{L}}} L_h(x_i)$, the constraint (8) is relaxed to

$$h(x_i) \geq \gamma_{safe} \quad \forall x_i \in \bar{X}_{safe}, \tag{10}$$

where

$$\bar{X}_{safe} = \left\{ x_i \in X_{safe} | \inf_{x \in X_{\mathcal{L}}} \|x - x_i\|_p \geq \frac{\gamma_{safe} + \gamma_{unsafe}}{L_h} \right\}$$

which gives rise to the first two constraints in (7).

To be a valid CBF, we still need to enforce that the derivative constraint in (3) holds. It suffices to show that there exists a control input $u \in \{u_i : (x_i, u_i) \in \mathcal{P}_{exp}\}$ from expert demonstrations such that (3) holds. To that end, for a given u_i, we define the function

$$q(x) \coloneqq L_f h(x) + L_g h(x) u_i + \alpha(h(x)),$$

which is Lipschitz continuous with $L_q(x)$ denoting its Lipschitz constant. Let $\gamma_{exp} > 0$ and X_{safe} be an ϵ-net of $\mathcal{D}$ with $\epsilon \leq \gamma_{exp}/L_q(x_i)$ for all $x_i \in X_{safe}$. Then,

$$q(x_i) \geq \gamma_{exp}, \quad \forall x_i \in X_{safe}, \tag{11}$$

ensures that that $q(x) \geq 0$ for all $x \in \mathcal{D}$ (i.e., the derivative constraint in (3) holds). Let $Lip(\cdot, \epsilon)$ be a function that returns an upper bound on the Lipschitz constant of its argument in an ϵ-neighborhood. The constraints

$$Lip(h(x_i), \bar{\epsilon}) \leq L_h, \quad \forall x_i \in X_{\mathcal{L}} \tag{12}$$

and

$$Lip(q(x_i, u_i), \epsilon) \leq L_q, \quad \forall (x_i, u_i) \in \mathcal{P}_{exp} \tag{13}$$

are included to further ensure the satisfaction of Lipschitz bound.

When h is a DNN, an unconstrained relaxation of problem 7 is proposed, and the unconstrained relaxation results in the optimization problem $min \ \| \theta \|^2 + \lambda_s T_s + \lambda_u T_u + \lambda_e T_e$, where

$$T_s = \sum_{x_i \in \bar{X}_{safe}} max\{\gamma_{safe} - h_\theta(x_i), 0\}$$

$$T_u = \sum_{x_i \in X_{\mathcal{L}}} max\{\gamma_{unsafe} + h_\theta(x_i), 0\}$$

$$T_e = \sum_{p_i \in \mathcal{P}_{exp}} max\{\gamma_{exp} - (\langle \nabla h_\theta(x_i), f(x_i) + g(x_i)u_i \rangle + \alpha(h_\theta(x_i))), 0\}.$$

The positive parameters λ_s, λ_u, λ_e are used to weigh the relative importance of each term, and p_i is a state-action pair (x_i, u_i).

D. Predictive Shielding Control

Due to the limited $\mathcal{P}_{dem}$, the CBF is gradually learned and continuously evolves using $\mathcal{P}_{exp}$ generated at runtime. It is inevitable that the learned CBF, especially at the early leaning stage, cannot ensure full security of the system. To address this issue, the predictive shielding control (PSC) is developed and included as an add-on modular to enable safety evaluation during training. That is, as shown in Fig. 2, PSC acts as a shield to the output (i.e., actions) of the learned CBF to keep the system safe from violating safety constraints. Specifically, let s be the current state and $\pi_{cbf}(s)$ be the action generated by the learned CBF. With the known dynamics in (1), we can predict the subsequent H states assuming that π_{cbf} is applied. If no safety constraint will be violated, the actions generated π_{cbf} will be implemented. Otherwise, random Gaussian noise will be added to $\pi_{cbf}(s)$ until no constraint violation can occur in H steps. It is empirically demonstrated in the simulation and experiment that the inclusion of PSC can not only ensure system security, but also speed up the learning rate, since unsafe actions are filtered out.

V. Simulations and Experiment

In this section, numerical simulation and physical experiment using two Crazyflie quadrotors are provided to demonstrate the effectiveness of the developed safe RL framework.

A. Simulation 1: Collision Avoidance

We first demonstrate the capability of learned CBF in avoiding inter-agent collisions via simulations. As shown in Fig. 3(a), two quadrotors are tasked to navigate to their own destinations while avoiding collision during flight. Without constraining their motions, the agents might collide as indicated by the dotted line in Fig. 3(a). The joint state of the two quadrotors is $x = [x_1\ y_1\ \Theta_1\ x_2\ y_2\ \Theta_2]^\top \in \mathbb{R}^6$. and the system inputs are the translational velocity v and the angular velocity ω, i.e., the actions $a = [v_1\ \omega_1\ v_2\ \omega_2]^\top \in \mathbb{R}^4$, which are constrained with $0 < v < 0.2$ and $-\pi \leq \omega \leq \pi$, respectively. The learning agent is trained for 10,000 episodes using the TD3 algorithm, with a maximum of 300 training steps per episode.

Fig. 3(b) shows the evolution of the learned CBF, where the red scatters represent the initial CBF learned from $\mathcal{P}_{dem}$, and the blue scatters represent the finally learned CBF. $p_{x,r}$ and $p_{y,r}$ represent the relative distances of two quadrotors in the x and y directions, respectively. The value of function $h(x)$ decreases as the distance between the quadrotors decreases. The performance and reward evolution of using conventional RL, RL with CBF, and RL with CBF and PSC, are shown in Fig. 3(c) and (d), respectively. Clearly, the learned CBF

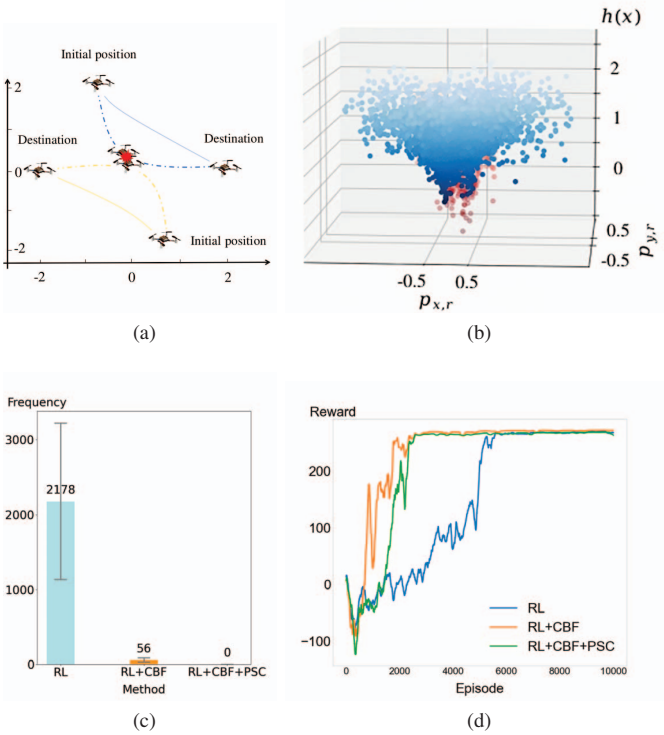
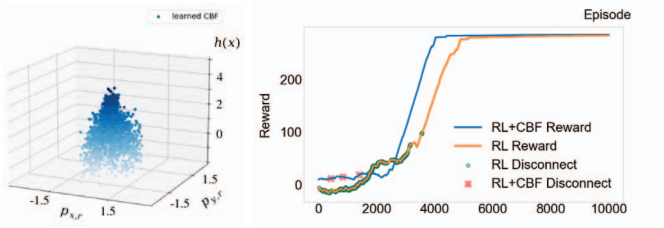


Fig. 3: (a) The navigation task, where the solid and dotted lines represent possible trajectories with and without consideration of collision avoidance, respectively. (b) The learned CBF for collision avoidance. (c) The collision frequency for conventional RL, RL with CBF, and RL with CBF and PSC, respectively. (d) The comparison of collected reward.

is effective in avoiding collisions. It is also observed that, when using CBF, the learning rate is significantly improved. A plausible explanation is that the exploration space is reduced since unsafe actions are filtered out.

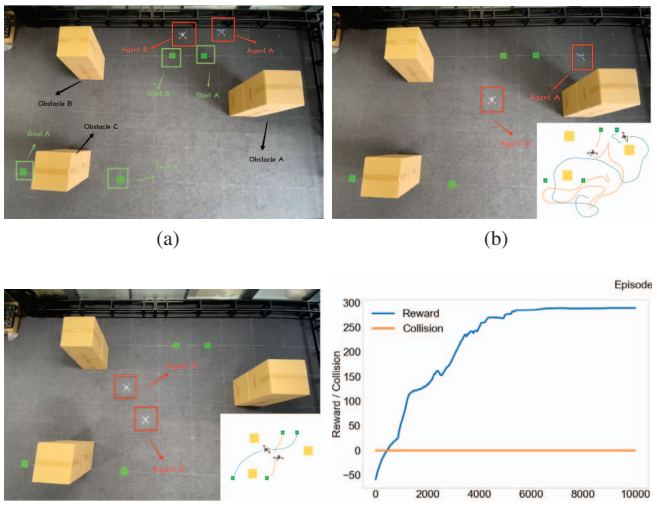
B. Simulation 2: Maintenance of Inter-Agent Connectivity

To further test the capability of the learned CBF in handling other types of constraint, we constrain the agent motion such that they have to stay close during navigation. Specifically, the two quadrotors are required to stay within a distance of 1.5 during flight. The finally learned CBF for maintaining the agents connected is shown in Fig. 4(a). By applying the learned CBF, the reward evolution and the number of disconnections (i.e., the times that inter-agent distance is greater than 1.5) are shown in Fig. 4(b). The solid lines represent the evolution of reward using conventional RL and RL with CBF, respectively. The dots and crosses indicate disconnections that occurred during training. Apparently, with the evolution of the learned CBF, the disconnections no longer occur after 2000 episodes, and far less often than when agents are allowed to move freely.



(a) (b)

Fig. 4: (a) The learned CBF for inter-agent connectivity. (b) The curve of reward and the occurred disconnections.



(a) (b) (c) (d)

Fig. 5: (a) The experimental environment, which contains three static obstacles. The quadrotors are tasked to navigate to their own goals, while avoiding collision and maintain inter-agent connectivity. (b) and (c) The learning performance at the 1649th and 10,000th episodes, respectively. (d) The blue and orange lines indicate the reward and agent collision curves respectively.

C. Experiment Results

Besides numerical simulation, physical experiment using two Crazyflie quadrotors is carried out for the navigation task while considering both collision avoidance and inter-agent connectivity. The environment setup is shown in Fig.5(a). The green points represent the initial and target locations, respectively. Fig.5(b) and (c) show the learning performance, where the bottom right plots show the trajectories of the quadrotors. Although the agents have not learned to complete the task in Fig.5(b), the safety constraints are always met during exploration. Fig.5(d) shows the curve of reward and strong safety guarantees, as no collision occur during the training. See the specific details of the experimental video.

VI. Conclusion

This work presents a RL-based framework that integrates CBF for safe exploration. A data-driven method is developed to learn the CBF from the on-line generated data at runtime, resulting in improved adaptation to complex environments. Since the current research relies on the model knowledge (i.e., system dynamics) to learn a valid CBF, future research will consider exploring advanced deep learning algorithms to learn CBFs for more complex tasks but with less model knowledge.

VII. Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 62173314.

References

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

[2] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2018, pp. 5571–5580.

[3] J. Schrittwieser, T. Hubert, A. Mandhane, M. Barekatain, I. Antonoglou, and D. Silver, "Online and offline reinforcement learning by planning with a learned model," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021.

[4] M. Cai, H. Peng, Z. Li, and Z. Kan, "Learning-based probabilistic ltl motion planning with environment and motion uncertainties," *IEEE Trans. Autom. Control*, vol. 66, no. 5, pp. 2386–2392, 2021.

[5] M. Heger, "Consideration of risk in reinforcement learning," in *Proc. Int. Conf. Mach. Learn.* Elsevier, 1994, pp. 105–111.

[6] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2015, pp. 1889–1897.

[7] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.

[8] S. Li and O. Bastani, "Robust model predictive shielding for safe reinforcement learning with stochastic dynamics," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2020, pp. 7166–7172.

[9] O. Bastani, S. Li, and A. Xu, "Safe reinforcement learning via statistical model predictive shielding," *Proc. Robot.: Sci. Syst.*, 2021.

[10] M. Z. Romdlony and B. Jayawardhana, "Uniting control lyapunov and control barrier functions," in *Proc. IEEE Conf. Decis. Control*. IEEE, 2014, pp. 2293–2298.

[11] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. on Autom. Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[12] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. IEEE Eur. Control Conf.* IEEE, 2019, pp. 3420–3431.

[13] H. Zhang, Z. Li, and A. Clark, "Model-based reinforcement learning with provable safety guarantees via control barrier functions," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2021, pp. 792–798.

[14] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela, "Synthesis of control barrier functions using a supervised machine learning approach," in *Proc.IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2020, pp. 7139–7145.

[15] S. Yaghoubi, G. Fainekos, and S. Sankaranarayanan, "Training neural network controllers using control barrier functions in the presence of disturbances," in *Proc. IEEE Intell. Transp. Syst. Conf.* IEEE, 2020, pp. 1–6.

[16] C. Dawson, B. Lowenkamp, D. Goff, and C. Fan, "Learning safe, generalizable perception-based hybrid control with certificates," *IEEE Robot.Autom. Lett.*, 2022.

[17] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning safe multi-agent control with decentralized neural barrier certificates," *arXiv preprint arXiv:2101.05436*, 2021.

[18] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *Proc. IEEE Conf. Decis. Control*. IEEE, 2020, pp. 3717–3724.

[19] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Int. Conf. Mach. Learn.* PMLR, 2018, pp. 1587–1596.