Your Value Function is a Control Barrier Function

Daniel C.H. Tan [1] **Fernando Acero** [1] **Robert McCarthy** [1] **Dimitrios Kanoulas** [1] **Zhibin Li** [1]

Abstract

Guaranteeing safe behaviour of reinforcement learning (RL) policies poses significant challenges for safety-critical applications, despite RL's generality and scalability. To address this, we propose a new approach to apply verification methods from control theory to learned value functions. By analyzing task structures for safety preservation, we formalize original theorems that establish links between value functions and control barrier functions. Further, we propose novel metrics for verifying value functions in safe control tasks and practical implementation details to improve learning. Our work presents a novel method for certificate learning, which unlocks a diversity of verification techniques from control theory for RL policies, and marks a significant step towards a formal framework for the general, scalable, and verifiable design of RL-based control systems.

1. Introduction

Deep reinforcement learning (RL) (Sutton & Barto, 2018) is a powerful and scalable tool for solving control problems, such as Atari games (Mnih et al., 2013), robotic control (Lillicrap et al., 2015), and protein folding (Jumper et al., 2021). However, because of their black-box nature, it is difficult to determine the behaviour of neural networks. In extreme cases, out-of-distribution or adversarially constructed inputs (Goodfellow et al., 2014) can catastrophically degrade network performance. In the control context, this can lead to highly unsafe behaviour; it is thus risky to deploy such controllers in safety-critical applications, such as autonomous vehicles or human-robot interaction, as well as future applications for general-purpose robots.

The problem of safe control has been extensively studied in safe reinforcement learning, through the lens of constrained Markov Decision Processes (Altman, 1999). Such methods implicitly assume that there are known constraints which are sufficient to guarantee safety. In contrast, our work assumes no prior knowledge of safe dynamics and aims to learn a constraint (in the form of a barrier function) to guarantee safety. This enables our approach to handle applications where safety cannot be easily expressed analytically, such as avoiding dynamic obstacles from raw pixel input (Dawson et al., 2022b).

On the other hand, there exists rich literature in control theory on proving properties of dynamical systems using *certificate functions*. The most well-known are Lyapunov functions, which prove the stability of dynamical systems around a fixed point (Isidori, 1985). Traditionally, it is difficult to design certificate functions for complex systems of interest. We discuss recent learning-based methods in Section 5. Other prior work combines classical and RL-based control methods by learning high-level policies over programmatic low-level controllers (Margolis & Agrawal, 2022), which could be designed to respect safety constraints.

However, designing effective and safe low-level controllers is still difficult and time-consuming. In both cases, the difficulty of manual design limits scalability to arbitrary tasks. Drawing inspiration from control theory, we aim to design a learning-based control method that benefits from the verifiability of certificate functions without sacrificing the generality and flexibility of reinforcement learning.

Our contributions are twofold. Firstly, we propose a **reinforcement learning method** for synthesizing control barrier certificates. Under mild assumptions on task structure, we prove a strong connection between barrier functions and value functions. We implement and ablate principled design choices for learning good barrier functions. Secondly, we propose and empirically validate novel **metrics** to evaluate the quality of learned barrier functions. We demonstrate that these metrics capture important structure not reflected in standard RL metrics. Control barrier certificates verified by these metrics successfully allow safety-constrained exploration of a large fraction of the safe state space, as shown in Figure 1.

Concretely, our method involves considering a safety-preserving task, where the reward function is given by $r = 0$

[1]Department of Computer Science, University College London, London, United Kingdom. Correspondence to: Daniel C.H. Tan <daniel.tan.22@ucl.ac.uk>.

2^{nd} *Workshop on Formal Verification of Machine Learning*, Honolulu, Hawaii, USA. Colocated with ICML 2023.

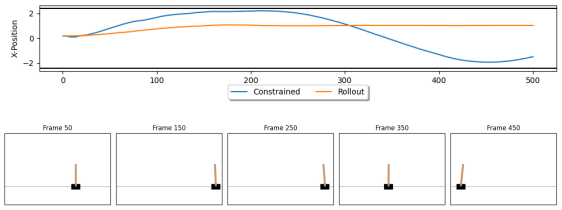

Figure 1. Top: Interpreting a value function as a control barrier function and performing safety-constrained exploration allows reaching both extremes of the CartPole safe state space (blue), compared to simply rolling out the optimal policy (orange). Bottom: The safety-constrained trajectory is visualized.

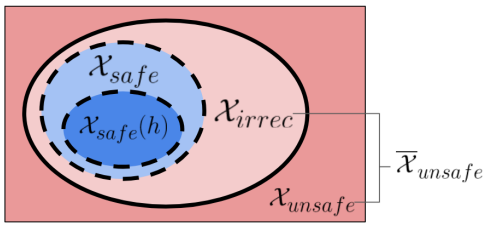

Figure 2. Partitioning of $\mathcal{X}$. $\mathcal{X}_{\text{unsafe}}$ is chosen by the engineer; $\mathcal{X}_{\text{safe}}, \mathcal{X}_{\text{irrec}}$ are then uniquely defined by $(f, \mathcal{X}_{\text{unsafe}})$. We define $\overline{\mathcal{X}}_{\text{unsafe}} = \mathcal{X}^c_{\text{safe}} = \mathcal{X}_{\text{unsafe}} \cup \mathcal{X}_{\text{irrec}}$. For most environments of interest, we expect that $\mathcal{X}_{\text{irrec}}$ is small relative to $\mathcal{X}_{\text{safe}}$.

in safety-violating states and $r = 1$ otherwise. We show that the value function V satisfies properties to be a control barrier function and we derive a threshold for predicting safety. We then learn V by using standard RL techniques, propose new metrics to verify learned V as a control barrier function, and finally demonstrate that our metrics capture the safety-preserving capacity of V. B y connecting value functions to certificate functions, our work presents a novel perspective on learning certificate functions, which offers a new approach for applying the wealth of verification strategies in control theory to reinforcement learning.

2. Preliminaries

In this work, we consider Markov Decision Processes (MDPs) and reinforcement learning (RL), with states $x \in \mathcal{X}$ and actions $u \in \mathcal{U}$. Further exposition is provided in Appendix A.

2.1. Indefinitely Safe Control

We consider augmenting an MDP with a set of *safety violations* $\mathcal{X}_{\text{unsafe}}$. This divides the state space into three subsets $\mathcal{X}_{\text{unsafe}}, \mathcal{X}_{\text{safe}}, \mathcal{X}_{\text{irrec}}$, illustrated in Figure 2. $\mathcal{X}_{\text{safe}}$ consists of *indefinitely safe* states; i.e. there exists a controller $\pi : \mathcal{X} \to \mathcal{U}$ such that $\mathcal{X}_{\text{safe}}$ is *forward-invariant* under f_π. $\mathcal{X}_{\text{irrec}}$ consists of *irrecoverable* states. For example, a car travelling at high velocity on a low-friction surface may inevitably collide with an iminent obstacle despite applying maximum braking effort. We define $\overline{\mathcal{X}}_{\text{unsafe}} = \mathcal{X}_{\text{unsafe}} \cup \mathcal{X}_{\text{irrec}}$.

Finite irrecoverability. In general, due to $\mathcal{X}_{\text{irrec}}$, safe control requires perfect knowledge of dynamics for arbitrarily long horizons, which can be intractable; hence we assume stronger conditions on $\mathcal{X}_{\text{irrec}}$. We say x is k-irrecoverable if it is guaranteed to enter $\mathcal{X}_{\text{unsafe}}$ within $k \in$ timesteps regardless of control. For $x \in \mathcal{X}_{\text{irrec}}$, let k_x be the minimum integer such that x is k-irrecoverable. We will assume $\{k_x : x \in \mathcal{X}\}$ is upper bounded by a constant $H < \infty$. Finite irrecoverability has been studied in previous work (Thomas et al., 2022), and is expected to be satisfied for reasonably well-actuated dynamics f and well-behaved choices of $\mathcal{X}_{\text{unsafe}}$.

2.2. Control Barrier Functions

Control barrier functions (CBFs) are a useful tool for solving safe control problems. A CBF $h : \mathcal{X} \to \mathbb{R}$ can be thought of as a classifier that classifies safe and unsafe states according to its level set $h(x) = 0$. The set $\{x : h(x) \geq 0\}$ defines a safe set $\mathcal{X}_{\text{safe}}(h)$. Loosely speaking, larger values of $h(x)$ correspond to 'safer' states. Formally, given $(M, \mathcal{X}_{\text{unsafe}})$, and $\alpha \in (0, 1]$, we say that $h : \mathcal{X} \to \mathbb{R}$ is a (discrete-time) CBF *against* $\mathcal{X}_{\text{unsafe}}$ if it satisfies:

$$\begin{aligned} &(i) \quad \forall x \in \mathcal{X}_{\text{unsafe}}, \quad h(x) < 0 \\ &(ii) \quad \forall x : h(x) \geq 0, \quad \sup_u \{h(f(x,u))\} \geq (1-\alpha)h(x) \end{aligned} \tag{1}$$

We note the following properties, proved in the appendix:

Lemma 2.1. *By condition (1)(i), $\mathcal{X}_{\text{safe}}(h) \cap \mathcal{X}_{\text{unsafe}} = \emptyset$. By condition (1)(ii), there exists a policy π such that $\mathcal{X}_{\text{safe}}$ is forward-invariant under f_π; if $x \in \mathcal{X}_{\text{safe}}$, then $f(x, \pi(x)) \in \mathcal{X}_{\text{safe}}$.*

A CBF h is useful for safe control because it eliminates the need to reason about dynamics over long horizons. Instead, we only need to check a one-step bound in condition (1)(ii) to guarantee safety indefinitely. One edge case occurs when $\mathcal{X}_{\text{safe}}(h) = \emptyset$; we call such CBFs *trivial*. Subsequently we assume that we can always find nontrivial CBFs against $\mathcal{X}_{\text{unsafe}}$ (if not, this indicates that $\mathcal{X}_{\text{unsafe}}$ is 'too large' and we should reconsider the choice of $\mathcal{X}_{\text{unsafe}}$).

Transforms of CBFs. Lastly, we note that certain classes of transformations preserve the control barrier function property, formalized as follows:

Lemma 2.2. *Let $h : \mathcal{X} \to \mathbb{R}$ be a CBF. Let $w : \mathbb{R} \to \mathbb{R}$ such that $\text{Im}(h) \subseteq \text{Dom}(w)$. Suppose there exists $C \in \mathbb{R}$*

such that for all $x, y \in \mathrm{Dom}(w)$, we have:

$$\begin{aligned} &(i) \quad w(x) \geq Cx \\ &(ii) \quad w(x) - w(y) \geq C(x-y) \\ &(iii) \quad \{x : w(x) \geq 0\} = \{x : x \geq 0\}. \end{aligned} \tag{2}$$

Then $\tilde{h} = w \circ h$ is also a CBF. We will say that such w are CBF-preserving transforms.

The proof is straightforward and given in the appendix.

3. Learning of Control Barrier Functions

3.1. Safety Preserving Task Structure

In the safety-preserving task framework, we assume a reward structure of: $r(x, u, x') = 0$ when $x' \in \mathcal{X}_{\text{unsafe}}$; otherwise, $r(x, u, x') = 1$. We also assume *early termination*, where the episode terminates immediately when $x' \in \mathcal{X}_{\text{unsafe}}$. Within this task structure, we analyze the optimal value function V^* under the partition illustrated in Figure 2, which consists of:

- $x \in \mathcal{X}_{\text{unsafe}}$. Since the episode terminates immediately, we trivially have $V(x) = 0$.
- $x \in \mathcal{X}_{\text{safe}}$. In this case, we know there exists a policy which preserves safety indefinitely, hence we have $V(x) = \sum_{j=0}^{\infty} \gamma^j(1) = \frac{1}{1-\gamma}$.
- $x \in \mathcal{X}_{\text{irrec}}$. Let x be k-irrecoverable. Then $V^*(x) = \sum_{j=0}^{k-1} \gamma^j = \frac{1-\gamma^k}{1-\gamma}$.

We make two remarks from this analysis. Firstly, V^* is *bounded* ; we have $V^*(x) \in [0, \frac{1}{1-\gamma}]$. Secondly, the range of V^* is *partitioned* by $\mathcal{X}_{\text{safe}}, \overline{\mathcal{X}}_{\text{unsafe}}$:

$$\sup_{\overline{\mathcal{X}}_{\text{unsafe}}} \{V(x)\} = \frac{1-\gamma^H}{1-\gamma} < \frac{1}{1-\gamma} = \inf_{\mathcal{X}_{\text{safe}}} \{V(x)\} \tag{3}$$

These two observations motivate us to propose CBFs of the form $h = V^* - R$, formalized below.

Theorem 3.1. *Let M be an MDP and suppose (a) early termination is employed with termination condition $c(x) = 1$, (b) r has safety-preserving reward structure, and (c) there exists an upper bound H on irrecoverability. Then for any $R \in (\frac{1-\gamma^H}{1-\gamma}, \frac{1}{1-\gamma}]$, we have that $h = V^* - R$ is a control barrier function against $\mathcal{X}_{\text{unsafe}}$.*

In practice, we do not have access to V^*; we only have access to learned functions $V \approx V^*$. Nonetheless, so long as V is 'not too far' from V^*, we can use $h = V(x) - R$ as a barrier function.

Theorem 3.2. *Let M be an MDP and let the assumptions (a) - (c) of Theorem 3.1 hold. Additionally, assume that V satisfies (d) ϵ-optimality; $\sup_{x \in X} |V(x) - V^*(x)| < \epsilon$, (e) $\epsilon < \frac{\gamma^H}{2(1-\gamma)}$. Then for $\alpha \in [\frac{2\epsilon}{\frac{1}{1-\gamma} + \epsilon - R}, 1]$ and any $R \in (\frac{1-\gamma^H}{1-\gamma} + \epsilon, \frac{1}{1-\gamma} - \epsilon]$, we have that $h = V - R$ is a control barrier function against $\mathcal{X}_{\text{unsafe}}$.*

We find that the bound on ϵ is very permissive. To illustrate how loose the bound is, let $H = 10$ (Thomas et al., 2022) and $\gamma = 0.99$. Then $\epsilon \leq \frac{1-\gamma^H}{2(1-\gamma)} \approx 47$ suffices, inducing a corresponding $R = \frac{1}{1-\gamma} - \epsilon \approx 53$ and $\alpha = 0.96$. For smaller ϵ, a wider range of values of R will be valid. In our experiments we find that $R = \frac{1}{2(1-\gamma)} = 50$ and $\alpha = 0.1$ work well empirically. Note that in our approach, we do not need to explicitly set H; rather, it is defined implicitly by R.

3.2. Reinforcement Learning Framework

We train a Deep Q-Network (Mnih et al., 2013) for 2×10^6 timesteps on the CartPole environment in OpenAI Gym (Brockman et al., 2016). A detailed description is provided in Appendix C. The network parametrizes a Q-function; the corresponding value function is $V(x) = \sup_{u \in \mathcal{U}} Q(x, u)$. The network is trained via standard temporal-difference learning (Sutton & Barto, 2018) to minimize the TD error: $\mathcal{L}_{TD} = \mathbb{E}_{(x,u,x') \sim f_\pi} \|r(x, u) + \gamma V(x') - V(x)\|^2$. Our baseline uses the implementation in CleanRL (Huang et al., 2022). Training results are visualized in Appendix B.

Implementation details. In theory, training a sufficiently expressive V for sufficiently long on the TD objective results in V converging uniformly to V^*. In practice, we find that training a vanilla DQN is insufficient; certain additional implementation details are required to obtain high-quality barrier functions. Below, we describe and motivate these design choices.

Bounded value. Recall from Section 3.1 that V^* is bounded; this motivates us to consider a parametrization of the form $V(x) = g(\sigma(\phi(x)))$ where ϕ is a neural network, $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function and $g(x) = \frac{x}{1-\gamma}$ is a linear mapping that allows V to have the correct range. We hypothesize that this aids learning by stabilizing the learning signal on the network weights, by essentially converting the two-sided regression loss into a one-sided classification loss. We denote architectures with bounded value by `SIGMOID`, and those without by `MLP`.

Supervision of V. Recall that we analytically know $V^*(x) = 0$ for $x \in \mathcal{X}_{\text{unsafe}}$. This motivates us to introduce a supervised loss $\mathcal{L}_{\text{unsafe}} = \mathbb{E}_{x \sim \mathcal{X}_{\text{unsafe}}} \|V(x)\|$. Since we can specify $\mathcal{X}_{\text{unsafe}}$, this loss can be approximated by sampling $\mathcal{X}_{\text{unsafe}}$ (e.g. by rejection sampling). We hypothesize that the supervised loss provides a valuable auxiliary learning signal that complements the standard TD objec-

tive. Because it is undesirable to enter unsafe states, we expect such states to be sparsely sampled. Furthermore, due to early termination, it may be outright impossible to reach most of $\mathcal{X}_{\text{unsafe}}$ (beyond a thin boundary). Hence, the supervised loss over $\mathcal{X}_{\text{unsafe}}$ provides a learning signal exactly in the regions where the TD objective does not, and vice versa. We indicate models trained with supervision by {SIGMOID,MLP}-SUP.

Exploration. We implement stronger exploration by modifying the initial state distribution to be more diverse. Because the TD objective only acts on states experienced during rollout, improved exploration provides a learning signal to V over a larger region of $\mathcal{X}$. Exploration through diverse reset initialization is enabled by default; to evaluate its impact, we perform an experiment using the original state distribution, denoted by NOEXP.

Recalling Lemma 2.2, we define barrier functions of the form $h = w(V(x) - R)$ with $R = \frac{1}{2(1-\gamma)}$. In the case where unbounded value functions are used, we let w be identity; i.e. $h(x) = V(x) - R$. In the case where bounded value functions are used, we define $h(x) = \phi(x) = w(V(x) - R)$. The corresponding transform is $w(x) = \tilde{\sigma}^{-1} \circ g^{-1}$, with $\tilde{\sigma} = \sigma(x) - 0.5$. We assert that w is CBF-preserving; a proof is given in the Appendix.

Remark 3.3. $w = \tilde{\sigma}^{-1} \circ g^{-1}$ is CBF-preserving.

Overall, we found that enabling all design choices together results in the best performance; furthermore, bounded value parametrization and supervision have negative effects alone but when combined synergize to produce an overall positive effect.

4. Verification of Learned CBFs

After obtaining candidate barrier functions through the learning process, it is crucial to verify whether they meet the conditions in (1). We investigate a total of 5 experimental settings, ablating each design choice, summarized in Table 1, and perform 5 seeded runs of each setting. We visualize the learned barrier functions for each setting in Figure 3. Overall, the SIGMOID-SUP model is best. Supervision is essential to ensuring $\mathcal{X}_{\text{safe}}(h) \cup \mathcal{X}_{\text{unsafe}} = \emptyset$. Exploration results in a larger $\mathcal{X}_{\text{safe}}(h)$. We also remark that SIGMOID results in more even contours than MLP.

Despite clear differences in CBFs between model variants, we note that standard metrics used in RL such as episode return and TD error fail to capture this discrepancy, as evidenced in Appendix B and Figure 6. Therefore, we further propose metrics that evaluate the quality of learned barrier functions.

Validity. Given h, we aim to quantify the extent to which it is valid across the state space, satisfying the conditions in (1). Concretely, we will define a *validity* metric m_{valid} to measure the quality of the learned CBF. Hence we rewrite (1) as logical assertions p_i and define associated predicates $\rho_i : \mathcal{X} \to \{0, 1\}$ indicating whether p_i holds for $x \in \mathcal{X}$.

- $p_1(x) := x \in \mathcal{X}_{\text{unsafe}} \implies h(x) < 0$. We define the associated predicate $\rho_1(h) = 1 - \mathbb{1}\{x \in \mathcal{X}_{\text{unsafe}}, h(x) \geq 0\}$.
- $p_2(x, \alpha) := h(x) \geq 0 \implies \sup_u\{h(f(x, u))\} \geq (1 - \alpha)h(x)$. We define the associated predicate $\rho_2(h, \alpha) = 1 - \mathbb{1}\{h(x) \geq 0, \sup_u\{h(f(x, u))\} < (1 - \alpha)h(x)\}$.

We have defined ρ_1, ρ_2 such that $\mathbb{E}_{x \in \mathcal{X}}[\rho_1(h)(x)]$ (respectively $\rho_2(h, \alpha)$) measures the fraction of states where condition (1)(i) (respectively (1)(ii)) holds. Since we need both conditions to hold for h to be a barrier function, it makes sense to define the metric $m_{\text{valid}}(h) = \mathbb{E}_{x \in \mathcal{X}}[\rho_1(h)(x)\rho_2(h, \alpha)(x)]$. In all experiments, we use a value of $\alpha = 0.1$.

Coverage. Given h, we would also like to measure the size of its safe set. A trivial barrier function (where $\mathcal{X}_{\text{safe}}(h) = \emptyset$) is of no practical use even if it is valid everywhere. We measure this with the *coverage* metric $m_{\text{cov}}(h) = \mathbb{E}_{x \in \mathcal{X}}[\mathbb{1}\{h(x) \geq 0\}]$, computed by sampling. In practice, we sample from a bounded subset $\mathcal{X}'$ which is assumed to contain $\mathcal{X}_{\text{safe}}$.

Discussion. Throughout the training history of different architectures, we observe a trade-off between validity and coverage, demonstrated in Figure 4. Validity refers to the extent to which a barrier function satisfies the specified conditions, while coverage measures the proportion of the state space on which the barrier function is applicable. The goal is to find the best barrier functions that achieve a validity metric, m_{valid}, equal to 1, indicating complete satisfaction of the conditions. Simultaneously, we aim to maximize the coverage, measured by the metric m_{cov}, while still maintaining the high validity. We visualize training histories of $m_{\text{cov}}, m_{\text{valid}}$ in Figure 7 of Appendix B. The final results are also summarized in Table 1. Empirically, bounded parametrization and supervision both aid in improving validity, whereas exploration aids in improving coverage. Thus, our experimental design choices are vindicated by evaluation on barrier metrics. More importantly, we note that standard RL metrics such as episode reward and TD error did not accurately distinguish between the learned networks in this regard. This demonstrates that our proposed barrier metrics provide a valuable and *orthogonal* perspective for evaluating learned barrier functions.

4.1. Safety Constraints with Barrier Functions

One common use of control barrier functions is to constrain a nominal policy π_{nom} to respect safety constraints.

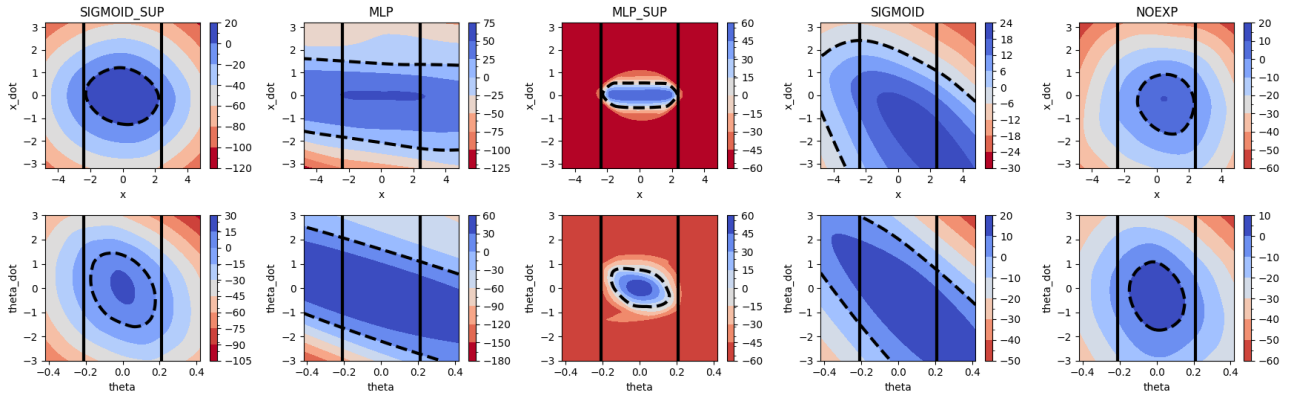

Figure 3. Phase diagram of learned control barrier functions for various experimental settings. In all cases, we use $R = 1/2(1-\gamma) = 50$. Top: $h(x, \dot{x}, 0, 0)$ for varying $x, \dot{x}$. Bottom: $h(0, 0, \theta, \dot{\theta})$ for varying $\theta, \dot{\theta}$. Solid lines demarcate $\mathcal{X}_{\text{unsafe}}$. Dashed lines indicate $\mathcal{X}_{\text{safe}}(h) = \{x : h(x) \geq 0\}$. Bounded parametrization results in more even contours. Without supervision, learned barrier function incorrectly classifies some unsafe states as safe.

EXPERIMENT	MLP	SIGMOID	MLP-SUP	SIGMOID-SUP	NOEXP
BOUNDED	NO	YES	NO	YES	YES
SUPERVISED	NO	NO	YES	YES	YES
EXPLORATION	YES	YES	YES	YES	NO
π^* RETURN	493 ± 14.8	$\mathbf{500} \pm 0$	465 ± 48.0	$\mathbf{500} \pm 0.0$	$\mathbf{500} \pm 0.0$
TD ERROR	2.43 ± 0.71	2.09 ± 0.27	0.958 ± 0.14	0.746 ± 0.057	$\mathbf{0.607} \pm 0.053$
$m_{valid}(h)$	0.476 ± 0.140	0.752 ± 0.130	0.603 ± 0.046	0.991 ± 0.002	$\mathbf{0.993} \pm 0.002$
$m_{cov}(h)$	$\mathbf{0.767} \pm 0.146$	0.477 ± 0.141	0.595 ± 0.048	0.106 ± 0.010	0.063 ± 0.013
π_h RETURN	9.36 ± 0.16	21.3 ± 14.4	21.3 ± 11.2	$\mathbf{163.5} \pm 54.7$	114.6 ± 85.1

Table 1. Description and final metrics for 5 seeds of 5 settings. On all metrics except coverage, enabling both bounded parametrization and supervision outperformed all ablations. The lower coverage can be explained by the trade-off between m_{valid} and m_{cov}.

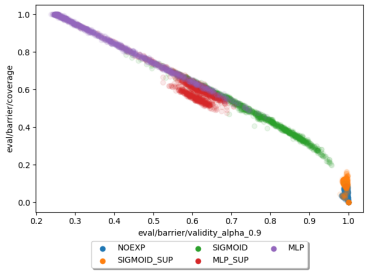

Figure 4. As validity increases, coverage tends to decrease. The best barrier functions have $m_{valid} = 1$, and m_{cov} as high as possible subject to that.

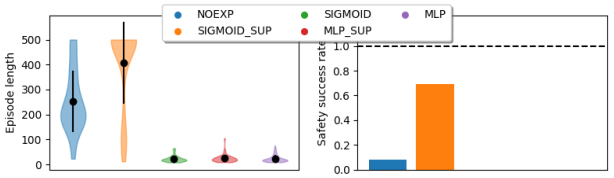

Figure 5. Safety-constrained exploration with learned CBFs. Left: Average safety-constrained episode length over 100 rollouts. Right: Safety success rate, defined as fraction of episodes with no safety violations. `SIGMOID-SUP` is the best safety constraint.

While this naively requires one-step lookahead to calculate $h(f(x,u))$, we note that the Q-function allows us to perform *implicit* one-step lookahead through the Bellman optimality condition $Q^*(x,u) = r(x,u) + \gamma V^*(f(x,u))$. Thus, we define the safety-constrained policy:

$$\pi_h(x) = \begin{cases} \pi_{nom}(x) & \text{if } Q(x, \pi_{nom}(x)) \geq R \\ \text{argmax}_u Q(x,u) & \text{if } Q(x, \pi_{nom}(x)) < R \end{cases} \tag{4}$$

We take π_{nom} to be the uniform random policy and roll out 100 episodes of π_h for varying h. For each architecture, we

evaluate (i) the safety-constrained episode length, and (ii) the safety success rate, defined as the fraction of episodes without safety violations. The results are summarized in Figure 5. On the whole, the architectures (blue, purple) with higher validity m_{valid} serve as better safety constraints, justifying the use of m_{valid} for model selection. However, we note that the best architecture failed to reach a safety success rate of 100%; we attribute this to the fact that m_{valid} is not a rigorous measure of validity, but only provides statistical evidence of validity through sampling.

5. Related work

Certificates and RL. Previous work on leveraging certificate functions in RL focuses on practical algorithms for learning safe control. Cheng et al. (2019) used CBFs during online learning to both guide exploration and guarantee safety with high probability. Westenbroek et al. (2021) proposed a method to learn safe, stabilizing policies for locomotion by minimizing control Lyapunov-Barrier functions using reinforcement learning. A similar work used CLFs in the reward formulation for sample-efficient learning on real-world CartPole and quadruped systems (Westenbroek et al., 2022). Concurrently with our work, Zhao et al. (2023) proposed an actor-critic method for learning discrete-time barrier functions, using an augmented Lagrangian method to constrain V. Compared to previous work, we are the first to formalize and prove a connection between value functions and control barrier functions, and analyze the resulting implications. From a practical standpoint, our method makes minimal modifications to the RL algorithm, and thus is simpler and more general.

Reward framework. The reward framework we consider was originally proposed in Safe MBPO (Thomas et al., 2022), which provided similar safety guarantees. However, the original formulation required H steps of *explicit* lookahead, and hence considered only the model-based setting as safety was estimated using model-based imaginary rollouts. By leveraging control theory, we show that we only need to perform a single step of *implicit* lookahead (through the Q function), allowing us to generalize to the model-free setting.

Learned certificates. Generally, there exists a wealth of literature on learning of neural certificates (Abate et al., 2021; Richards et al., 2018; Manek & Kolter, 2020; Gaby et al., 2021). While a full review of certificate learning is outside the scope of this paper, we refer interested readers to Dawson et al. (2022a) for a comprehensive survey. Learning methods for neural certificates typically rely on self-supervised learning, consider continuous systems, assume knowledge of dynamics, and *control-affine* dynamics. Certificates for discrete-time systems were studied in (Grizzle & Kang, 2001; Dai et al., 2020). Recent work studied certificate learning for black-box systems through learned dynamics models (Qin et al., 2022). Compared to the main body of work on certificate learning, our method is applicable to a much wider range of systems as it works with black-box dynamics, discrete-time systems, and does not need control-affineness.

Safe RL. Finally, we discuss our work in the context of the safe RL literature. As discussed in the introduction, such methods aim to preserve safety by augmenting M with safety constraints of the form $c_i(x, u) \leq 0$. Methods for learning to solve cMDPs have been widely studied, such as Lagrangian methods (Tessler et al., 2018; Stooke et al., 2020) and Lyapunov-based methods (Chow et al., 2018). Recent work considers building a trust region of policies (Achiam et al., 2017; Zanger et al., 2021), projecting to a safer policy (Yang et al., 2020), and using conservative policy updates (Bharadhwaj et al., 2021). Within this context, our results show that learned Q-functions can be directly used in a constraint of the form $c(x, u) = Q(x, u) - R \leq 0$ in order to guarantee safety. Hence, our method is orthogonal to and compatible with all of the safe RL methods discussed above.

6. Limitations and Future Work

Safety violations during exploration. Our method assumes no prior knowledge on dynamics. Hence, encountering safety violations is the sole learning signal for safety. This may not be the case for learning in real-world environments where safety must be preserved throughout the exploration. An exciting direction for future work is to reduce safety violations during exploration by using nominal (and possibly imperfect) dynamics models to pre-train a CBF solution using self-supervised learning approaches (Dawson et al., 2022a), and subsequently fine-tune using our RL-based method.

Soft safety guarantees. Despite empirically correlating well with the capacity of barrier functions to constrain unsafe policies, our validity metric can be interpreted as a statistical argument for safety, rather than a formal proof; indeed, provable guarantees are impossible so long as we assume completely black-box dynamics. By considering gray-box dynamics models instead, such as nominal models with unknown parameters, future work can explore methods that provide stronger guarantees such as rigorous verification through Lipschitz-continuity bounds (Dawson et al., 2022a), formal verification relying on symbolic logic (Xie et al., 2022), or exhaustive verification (Albarghouthi, 2021).

Discrete-action environments. This work so far considers discrete-action environments as barrier functions are parametrized in terms of Q-functions and evaluate $V(x) = \sup_{u \in \mathcal{U}} Q(x, u)$, which is more difficult in continuous-

action environments. Nevertheless, the results in Section 3.1 show the applicability to any general MDP. An analogous method for continuous-action tasks could use an actor to obtain a variational lower bound on V, or explore dueling network architectures (Wang et al., 2016) which parametrize and learn separate V, Q.

Sample efficiency. Our work adopts a minimalist RL approach which can be sample-inefficient. Future work can improve sample efficiency through offline datasets (Fu et al., 2020), model-based reinforcement learning (Janner et al., 2019; Hafner et al., 2019), or better representation learning methods (Eysenbach et al., 2022; Wang et al., 2023).

7. Conclusion

This work presents theoretical contributions that establish a connection between barrier functions and value functions and demonstrates the feasibiliy of learning of barrier functions through an RL approach. We explore and ablate critical implementation details for learning high-quality barrier functions using our method. We demonstrate that standard RL metrics fail to evaluate the capacity of learned barrier functions to act as safety constraints. To address this gap, we propose our own novel barrier metrics.

The proposed approach is especially suitable for learning **perceptual CBFs**, where safety can be defined as a direct function of sensor inputs. In one case study, perceptual CBFs on LiDAR scans enabled safe obstacle avoidance in cluttered environments (Dawson et al., 2022b). In contrast to self-supervised learning, which requires careful handling of sensor dynamics, reinforcement learning naturally scales to end-to-end robot control (Levine et al., 2016), making it a promising alternative.

The theoretical contributions of this work have broad applicability and can extend to any MDP M with any choice of RL algorithm. This suggests that our method can be employed to learn barrier functions for safe control in diverse tasks. Future work will extend to tasks with different reward structures by defining an auxiliary safety-preserving reward for the unsafe set $\mathcal{X}_{\text{unsafe}}$ and training an auxiliary value function as the CBF. This will enable joint learning of safety constraints and task-oriented behaviours.

In summary, our work contributes to the development of general, scalable, and *verifiable* control methods that can be applied to various tasks. By introducing novel barrier metrics and leveraging reinforcement learning techniques, we provide a useful framework for developing verifiable control systems, enabling safer and more reliable autonomous behaviors in real-world environments.

References

Abate, A., Ahmed, D., Giacobbe, M., and Peruffo, A. Formal synthesis of lyapunov neural networks. *IEEE Control Systems Letters*, 5:773–778, 7 2021. doi: 10.1109/lcsys.2020.3005328. URL https://doi.org/10.1109%2Flcsys.2020.3005328.

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization, 2017.

Albarghouthi, A. Introduction to neural network verification. *Foundations and Trends in Programming Languages*, 7:1–164, 9 2021. ISSN 23251131. doi: 10.1561/2500000051. URL https://arxiv.org/abs/2109.10317v2.

Altman, E. *Constrained Markov Decision Processes*. Routledge, 1999.

Bharadhwaj, H., Kumar, A., Rhinehart, N., Levine, S., Shkurti, F., and Garg, A. Conservative safety critics for exploration, 2021.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016.

Cheng, R., Orosz, G., Murray, R. M., and Burdick, J. W. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pp. 3387–3395, 2019. ISSN 2159-5399. doi: 10.1609/AAAI.V33I01.33013387. URL https://dl.acm.org/doi/10.1609/aaai.v33i01.33013387.

Chow, Y., Nachum, O., Duenez-Guzman, E., and Ghavamzadeh, M. A lyapunov-based approach to safe reinforcement learning, 2018.

Dai, H., Landry, B., Pavone, M., and Tedrake, R. Counter-example guided synthesis of neural network lyapunov functions for piecewise linear systems. *Proceedings of the IEEE Conference on Decision and Control*, 2020-December:1274–1281, 12 2020. ISSN 25762370. doi: 10.1109/CDC42340.2020.9304201.

Dawson, C., Gao, S., and Fan, C. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods, 2022a.

Dawson, C., Lowenkamp, B., Goff, D., and Fan, C. Learning safe, generalizable perception-based hybrid control with certificates. *IEEE Robotics and Automation Letters*, 7:1904–1911, 4 2022b. ISSN 23773766. doi:

10.1109/LRA.2022.3141657. URL https://arxiv.org/abs/2201.00932v1.

Eysenbach, B., Zhang, T., Levine, S., Salakhutdinov, R., Cmu, A., and Research, G. Contrastive learning as goal-conditioned reinforcement learning. 6 2022. URL https://arxiv.org/abs/2206.07568v2.

Fu, J., Kumar, A., Nachum, O., Brain, G., Brain, G. T. G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. 4 2020. URL https://arxiv.org/abs/2004.07219v4.

Gaby, N., Zhang, F., and Ye, X. Lyapunov-net: A deep neural network architecture for lyapunov function approximation. *Proceedings of the IEEE Conference on Decision and Control*, 2022-December:2091–2096, 9 2021. ISSN 25762370. doi: 10.1109/CDC51059.2022.9993006. URL https://arxiv.org/abs/2109.13359v2.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 12 2014. URL https://arxiv.org/abs/1412.6572v3.

Grizzle, J. W. and Kang, J. M. Discrete-time control design with positive semi-definite lyapunov functions. *Systems & Control Letters*, 43:287–292, 7 2001. ISSN 0167-6911. doi: 10.1016/S0167-6911(01)00110-4.

Hafner, D., Deepmind, T. L., Ba, J., Norouzi, M., and Brain, G. Dream to control: Learning behaviors by latent imagination. 12 2019. URL https://arxiv.org/abs/1912.01603v3.

Huang, S., Dossa, R. F. J., Ye, C., Braga, J., Chakraborty, D., Mehta, K., and Araújo, J. G. M. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23:1–18, 2022. URL http://jmlr.org/papers/v23/21-1342.html.

Isidori, A. *Nonlinear control systems: an introduction*. Springer, 1985.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 32, 6 2019. ISSN 10495258. URL https://arxiv.org/abs/1906.08253v3.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with alphafold. *Nature 2021 596:7873*, 596:583–589, 7 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://www.nature.com/articles/s41586-021-03819-2.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17:1–40, 2016.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 9 2015. URL https://arxiv.org/abs/1509.02971v6.

Manek, G. and Kolter, J. Z. Learning stable deep dynamics models. *Advances in Neural Information Processing Systems*, 32, 1 2020. ISSN 10495258. URL https://arxiv.org/abs/2001.06116v1.

Margolis, G. B. and Agrawal, P. Walk these ways: Tuning robot control for generalization with multiplicity of behavior, 9 2022.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning, 2013.

Qin, Z., Sun, D., and Fan, C. Sablas: Learning safe control for black-box dynamical systems. *IEEE Robotics and Automation Letters*, 7:1928–1935, 1 2022. ISSN 23773766. doi: 10.1109/LRA.2022.3142743. URL https://arxiv.org/abs/2201.01918v2.

Richards, S. M., Berkenkamp, F., and Krause, A. The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems, 2018.

Stooke, A., Achiam, J., and Abbeel, P. Responsive safety in reinforcement learning by pid lagrangian methods. JMLR.org, 2020.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. A Bradford Book, 2018. ISBN 0262039249.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization, 2018.

Thomas, G., Luo, Y., and Ma, T. Safe reinforcement learning by imagining the near future, 2022.

Wang, T., Torralba, A., Isola, P., and Zhang, A. Optimal goal-reaching reinforcement learning via quasimetric

learning. 4 2023. URL https://arxiv.org/abs/2304.01203v3.

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., and de Freitas, N. Dueling network architectures for deep reinforcement learning, 2016.

Westenbroek, T., Agrawal, A., Castañeda, F., Sastry, S. S., and Sreenath, K. Combining model-based design and model-free policy optimization to learn safe, stabilizing controllers. *IFAC-PapersOnLine*, 54:19–24, 2021. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2021.08.468. URL https://www.sciencedirect.com/science/article/pii/S240589632101243X. 7th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2021.

Westenbroek, T., Castaneda, F., Agrawal, A., Sastry, S., and Sreenath, K. Lyapunov design for robust and efficient robotic reinforcement learning, 2022.

Xie, X., Kersting, K., and Neider, D. Neuro-symbolic verification of deep neural networks. *IJCAI International Joint Conference on Artificial Intelligence*, pp. 3622–3628, 3 2022. ISSN 10450823. doi: 10.24963/ijcai.2022/503. URL https://arxiv.org/abs/2203.00938v1. Verification of network behaviour by expressing desired behaviour as a logical formula and trying to find counter-examples using satisfiability solvers.

Yang, T.-Y., Rosca, J., Narasimhan, K., and Ramadge, P. Accelerating safe reinforcement learning with constraint-mismatched policies, 5 2020.

Zanger, M. A., Daaboul, K., and Zöllner, J. M. Safe continuous control with constrained model-based policy optimization, 2021.

Zhao, L., Gatsis, K., and Papachristodoulou, A. A barrier-lyapunov actor-critic reinforcement learning approach for safe and stable control, 2023.

A. Preliminaries

A.1. Markov Decision Processes

A Markov Decision Process M can be defined as a tuple (MDPs) $M = (\mathcal{X}, \mathcal{U}, f, r, \gamma)$, where $\mathcal{X}$ is the state space, $\mathcal{U}$ the control space, $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ the (discrete-time) dynamics, $r : \mathcal{X} \times \mathcal{U} \to [r_{min}, r_{max}]$ the reward function, and $\gamma \in [0, 1)$ the discount factor. A trajectory τ is a sequence $\{(x_t, u_t, r_t)\}_{t \in \mathbb{N}}$ satisfying $x_{t+1} = f(x_t, u_t)$ and $r_t = r(x_t, u_t)$. A policy $\pi : \mathcal{X} \to \mathcal{U}$ induces associated *closed-loop* dynamics $f_\pi(x) = f(x, \pi(x))$.

A.2. Reinforcement Learning

Reinforcement learning (RL) is a broad family of algorithms designed to solve MDPs. Given a policy $\pi : \mathcal{X} \to \mathcal{U}$, it is common to define the Q-value function Q_π and state value function V_π.

$$Q_\pi(x, u) = r(x, u) + \Big[\sum_{t=1}^{\infty} \gamma^t r(x_t, \pi(x_t))\Big]$$
$$V_\pi(x) = \sup_u \{Q_\pi(x, u)\}$$

The optimal Q^*, V^* (for the reward-maximizing policy π^*) satisfy the one-step Bellman equality:

$$Q^*(x, u) = r(x, u) + \gamma V^*(f(x, u)) \tag{5}$$

B. Training History

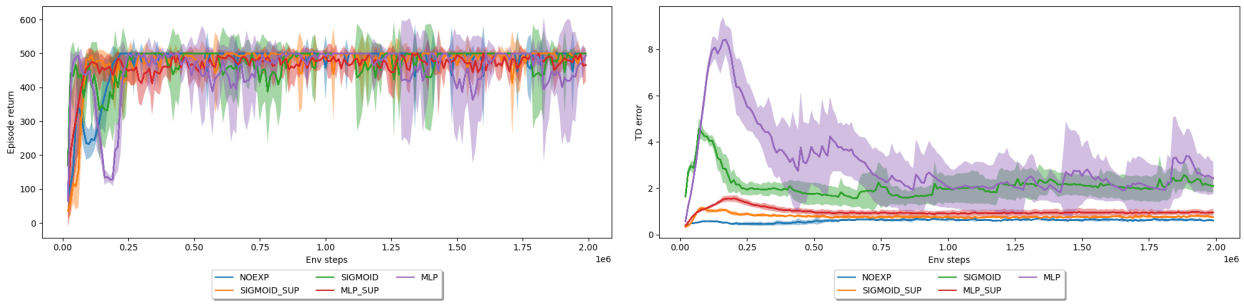

Figure 6. Left: Mean episode return over 10 rollouts. In all cases, the Q-greedy policy achieves the maximum return of 500. Right: Mean TD error across $n = 10,000$ points sampled uniformly from the state space. The architectures with bounded parametrization achieve a lower mean TD error.

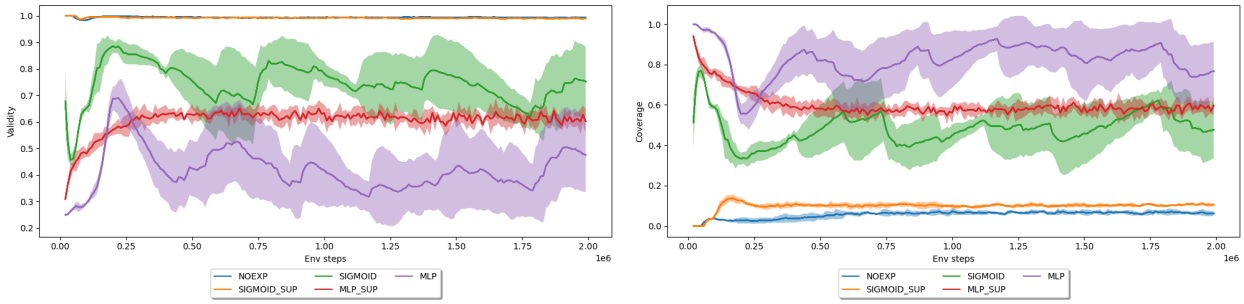

Figure 7. Training history of m_{valid} (top) and m_{cov} (bottom). The bounded and supervised value networks achieved the highest validity of approximately 100%. Enabling exploration increased coverage.

C. CartPole Schematic

We provide a schematic of the CartPole environment

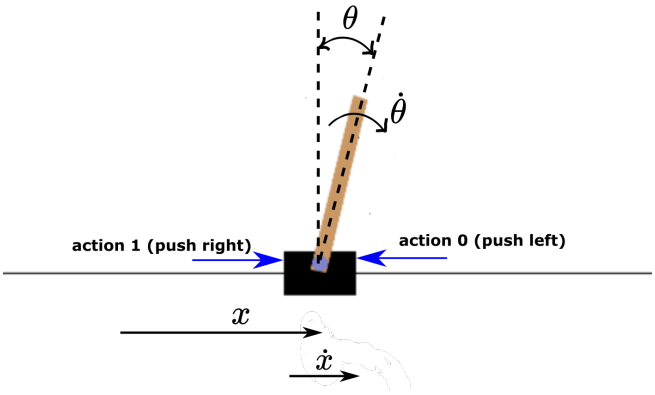

Figure 8. Schematic of the CartPole environment. The state space is parametrized as $(x, \dot{x}, \theta, \dot{\theta})$, where x is the cart position and θ is the pole angle. We use the default termination condition $c(x) = \{|x| \geq 2.4 \text{ or } |\theta| \geq 12^\circ\}$ and define $\mathcal{X}_{\text{unsafe}} = \{x : c(x) = 1\}$.

D. Proofs

Proof of Lemma 2.1.

Proof. Since $\alpha \leq 1$, if $h(x) \geq 0$ then $\sup_u\{h(f(x,u))\} \geq 0$. Thus, we can define $\pi(x) = \operatorname{argmax} h(f(x,u))$ and it is easy to see that $x \in \mathcal{X}_{\text{safe}} \implies f(x, \pi(x)) \in \mathcal{X}_{\text{safe}}$. □

Proof of Lemma 2.2.

Proof. We prove that $\tilde{h} = w \circ h$ satisfies both conditions discussed in (1) to be a valid CBF. First, note:

$$\begin{aligned} x \in \mathcal{X}_{\text{unsafe}} &\implies h(x) < 0 && \text{by (1)(i)} \\ &\iff w(h(x)) < 0 && \text{by (2)(iii)} \end{aligned}$$

Hence $\tilde{h}$ satisfies (1)(i). Note that $w(h(x) \geq 0) \iff h(x) \geq 0$. Now, for $x : \tilde{h}(x) \geq 0$:

$$\begin{aligned} \sup_u\{w(h(f(x,u))) - w(h(x))\} &\geq C \sup_u\{h(f(x,u)) - h(x)\} && \text{by (2)(ii)} \\ &\geq -C\alpha h(x) && \text{by (1)(ii)} \\ &\geq -\alpha w(h(x)) && \text{by (2)(i)} \end{aligned}$$

Hence $\tilde{h}$ satisfies (1)(ii). □

Proof of Theorem 3.1

Proof. We consider the two conditions presented in (1) that CBFs must satisfy. From (3), it is clear that (1)(i) is satisfied. Similarly, we note that $h(x) \geq 0$ implies x is indefinitely safe; then by definition there exists a control such that $h(f(x,u)) = \frac{1}{1-\gamma} - R \geq (1-\alpha)(\frac{1}{1-\gamma} - R) = (1-\alpha)h(x)$. This proves condition (1)(ii). □

Proof of Theorem 3.2

Proof. We prove that $h = V - R$ satisfies both conditions discussed in (1) to be a valid CBF. First, let $x \in \mathcal{X}_{\text{unsafe}}$; then $V(x) \leq V^*(x) + \epsilon = \frac{1-\gamma^H}{1-\gamma} + \epsilon < R$. Hence $h(x) < 0$ and condition (1)(i) is satisfied.

Now, let $h(x) \geq 0$. Then $x \in \mathcal{X}_{\text{safe}}$, thus $\sup_u h(f(x,u)) \geq V^*(x) - \epsilon - R = \frac{1}{1-\gamma} - \epsilon - R$. Similarly, we have $h(x) \leq V^*(x) + \epsilon - R = \frac{1}{1-\gamma} + \epsilon - R$. Then, to satisfy condition (1)(ii), it suffices that:

$$\frac{1}{1-\gamma} - \epsilon - R \geq (1-\alpha)(\frac{1}{1-\gamma} + \epsilon - R)$$
$$\implies \alpha \geq \frac{2\epsilon}{\frac{1}{1-\gamma} + \epsilon - R}$$

Note that this can be satisfied because $R < \frac{1}{1-\gamma} - \epsilon$; hence the R.H.S is strictly smaller than 1. Hence condition (1)(ii) is satisfied under assumptions (a)-(e), which completes the proof. □

Proof of Remark 3.3 We first note that $g^{-1} = (1-\gamma)x$ is CBF preserving with $C = (1-\gamma)$. Conditions (2.2)(i), (ii), (iii) are all trivially verifiable.

Next, we show that $\tilde{\sigma}^{-1}$ is CBF-preserving with $C = 1$, where $\tilde{\sigma}(x) = \sigma(x) - 0.5$. First, note that $x \geq 0 \iff \sigma(x) \geq 0.5$ and hence (i) is satisfied by substituting $x \to \tilde{\sigma}^{-1}(x)$. Next, note the identity $\sigma(x) \leq x + 0.5$; this implies $\sigma(\tilde{x}) \leq x$. Again, by substituting $x \to \tilde{\sigma}^{-1}(x)$, we observe that condition (ii) is satisfied. Lastly, note that σ is Lipschitz-continuous with $L = 1$. This implies that $\sigma(x) - \sigma(y) \leq x - y$ for $x > y$; hence $\tilde{\sigma}(x) - \tilde{\sigma}(y) \leq x - y$. By substituting $x \to \tilde{\sigma}^{-1}(x), y \to \tilde{\sigma}^{-1}(y)$ we see that condition (iii) is satisfied.

Lastly, since both $g^{-1}, \tilde{\sigma}^{-1}$ are CBF-preserving, if h is a CBF then $g^{-1} \circ h$ is also a CBF, and then $\tilde{\sigma}^{-1} \circ g^{-1} \circ h$ is also a CBF. Hence w is CBF-preserving as claimed.