2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

How to Train Your Neural Control Barrier Function: Learning Safety Filters for Complex Input-Constrained Systems

Oswin So[1], Zachary Serlin[2], Makai Mann[2], Jake Gonzales[2], Kwesi Rutledge[1], Nicholas Roy[1], Chuchu Fan[1]

***Abstract*— Control barrier functions (CBFs) have become popular as a safety filter to guarantee the safety of nonlinear dynamical systems for arbitrary inputs. However, it is difficult to construct functions that satisfy the CBF constraints for high relative degree systems with input constraints. To address these challenges, recent work has explored learning CBFs using neural networks via neural CBFs (NCBFs). However, such methods face difficulties when scaling to higher dimensional systems under input constraints. In this work, we first identify challenges that NCBFs face during training. Next, to address these challenges, we propose policy neural CBFs (PNCBFs), a method of constructing CBFs by learning the value function of a nominal policy, and show that the value function of the maximum-over-time cost is a CBF. We demonstrate the effectiveness of our method in simulation on a variety of systems ranging from toy linear systems to a jet aircraft with a 16-dimensional state space. Finally, we validate our approach on a two-agent quadcopter system on hardware under tight input constraints.**

I. Introduction and Related Works

Techniques employing control barrier functions (CBFs) are powerful tools for safety-critical control of dynamical systems. In particular, CBFs can be used as a safety filter to maintain and certify the safety of any system under arbitrary inputs. This safety guarantee is crucial in order to give users the needed confidence for greater adoption of robotics in safety-critical domains such as autonomous driving [1], surgical robotics [2], and urban air mobility [3].

Despite their theoretical advantages, constructing CBFs in practice remains difficult. While it is easy to construct a *candidate* CBF, it is much harder to verify the conditions necessary to enjoy the safety guarantees of a *valid* CBF for systems with input constraints. Consequently, input constraints are often ignored when using CBFs in practice [4–7].

CBFs for High Relative Degree Systems under Input Constraints. To address the above challenges with CBFs, recent works try to simplify the construction of valid CBFs for high relative degree systems and input constraints. In particular, backup CBFs constrain the system to states where a fallback controller can maintain safety [8–11]. These approaches, however, either require knowledge of an invariant set for the fallback controller which is difficult to compute in itself, or require an appropriate predictive horizon that trades between myopic unsafe behavior and performance.

[1] Massachusetts Institute of Technology.
[2] MIT Lincoln Laboratory.
* Corresponding Author. oswinso@mit.edu.

This work was supported by NASA University Leadership initiative (grants #80NSSC20M0163 and 80NSSC22M0070), and National Science Foundation CAREER Award (grant #CCF-2238030). This article solely reflects the opinions and conclusions of its authors and not any NASA entity. ©2023 Massachusetts Institute of Technology.

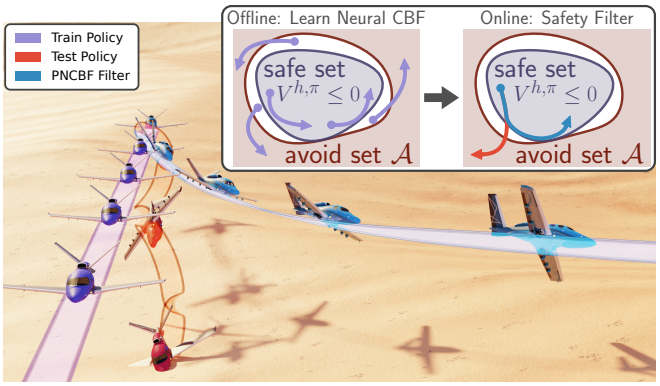

Fig. 1. We train a Policy Neural CBF (PNCBF) by learning the value function $V^{h,\pi}$ for a given **train policy** offline. The sublevel set of $V^{h,\pi}$ contains all states from which the **train policy** remains safe. The PNCBF can then be used online as a **safety filter** to ensure the safety of any **potentially unsafe test policy**. Our method avoids the pitfalls of previous Neural CBF approaches and can scale to high dimensional systems such as a jet aircraft.

Neural CBFs. Recently, learning based approaches have been used to learn neural CBFs (NCBFs) that approximate CBFs using neural networks [12], part of a more general trend of learning neural certificates [13–23]. Owing to the flexibility of neural networks, NCBFs have been extended to handle parametric uncertainties [24], obstacles with unknown dynamics [25] and multi-agent control [26–28]. However, many existing NCBF approaches do not consider input constraints. Recent work has examined incorporating input constraints into NCBFs [29], but this approach requires solving a minimax problem that can be brittle to solve in practice.

Reachability Analysis. Reachability analysis provides a powerful tool for analyzing the safety of dynamical systems. Hamilton-Jacobi (HJ) reachability analysis computes the largest control-invariant set [30] and is often computed using grid-based methods [31]. Recent works connect the HJ value function with CBFs [32], providing an alternative to Sum-of-Squares programming for automated synthesis. However, the curse of dimensionality limits the practical applicability of grid-based solvers to systems with state-dimension smaller than 5 [31].

Contributions. We summarize our contributions as follows:

1) We identify challenges that existing training methods for Neural CBFs face when under input constraints.
2) We show that the policy value function is a valid CBF. Using this insight, we propose learning Neural CBFs via the policy value function, thereby bypassing the

challenges that previous neural CBF approaches face.
3) We demonstrate our approach with extensive simulation experiments and show that our method can yield much larger control invariant sets and can scale to higher dimensional systems than current state of the art methods.
4) We validate our approach on a two-agent quadcopter system on hardware.

II. Preliminaries

A. Problem Definition

We consider continuous-time, control-affine dynamics

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n, u \in \mathcal{U} \subseteq \mathbb{R}^m$ and f, g are locally Lipschitz continuous functions. Let $\mathcal{A} \subset \mathcal{X}$ denote a set of unsafe states. We now state the safe controller synthesis problem we wish to tackle below.

Problem 1 (Safe Controller Synthesis)**.** *Given the system* (1) *and an avoid set $\mathcal{A} \subset \mathcal{X}$, find a control policy $\pi : \mathcal{X} \to \mathcal{U}$ that prevents the system from entering the avoid set $\mathcal{A}$, i.e.,*

$$x_0 \notin \mathcal{A} \implies x_t \notin \mathcal{A}, \quad \forall t \geq 0. \tag{2}$$

Often, we have a test policy $\pi_{\text{test}} : \mathcal{X} \to \mathcal{U}$ that is performant but may not be safe. In this case, we want our policy π to *minimally modify* π_{test} to maintain safety.

Problem 2 (Safety Filter Synthesis)**.** *Solve Problem 1 with the additional desire that π is close to π_{test}. Specifically, we wish to solve the optimization problem*

$$\min_{\pi} \quad \|\pi - \pi_{\text{test}}\| \tag{3a}$$
$$\text{s.t.} \quad x_t \notin \mathcal{A}, \quad \forall t \geq 0, \tag{3b}$$

where $\|\cdot\|$ is some distance metric.

In this work, we focus on solving Problem 2 with Control Barrier Functions (CBFs), as we describe below.

B. Safety Filter Synthesis with Control Barrier Functions

We focus on CBFs [33–35] as a solution to Problem 2. Specifically, let $B : \mathcal{X} \to \mathbb{R}$ be a continuously differentiable function, $\alpha : \mathbb{R} \to \mathbb{R}$ be an extended class-κ function[1], and[2]

$$B(x) > 0, \quad \forall x \in \mathcal{A}, \tag{4a}$$
$$B(x) \leq 0 \implies \inf_{u \in \mathcal{U}} L_f B(x) + L_g B(x) u \leq -\alpha\big(B(x)\big), \tag{4b}$$

where $L_f B \coloneqq \nabla B^\mathsf{T} f$, $L_g B \coloneqq \nabla B^\mathsf{T} g$. Then, B is a CBF, and any control u that satisfies the *descent condition* (4b) renders the sublevel set of B $\{ x \in \mathcal{X} \mid B(x) \leq 0 \}$ forward-invariant, i.e., any trajectory starting from within this set remains in this set under such a choice of u. In particular, since (4b) is a linear constraint on u, we can solve Problem 2 using the following Quadratic Program (QP).

$$\begin{aligned} \min_{u \in \mathcal{U}} \quad & \|u - \pi_{\text{test}}(x)\|^2 \\ \text{s.t.} \quad & L_f B(x) + L_g B(x) u \leq -\alpha(B(x)) \end{aligned} \tag{5}$$

[1]Extended class-κ is the set of continuous, strictly increasing functions α such that $\alpha(0) = 0$

[2]Some works [35] define the unsafe set to be the zero superlevel set of B, while some use the sublevel set. We use the former definition in this work.

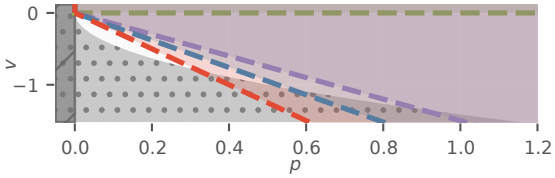

Fig. 2. **HOCBF on the double integrator.** On a double integrator with box control constraints $|u| \leq 1$ and the constraint $p \geq 0$, applying different values of the α to the HOCBF candidate $B(x) = -v - \alpha p$ results in different boundaries of the resulting safety filter. However, the only *valid* choice that satisfies the CBF descent condition (4b) is $\alpha = 0$ (green), which disallows any negative velocities and is overly conservative. All other choices of α intersect the true unsafe region (gray dotted) at some point and violate (4b).

Challenges with CBF synthesis. Define a *candidate* CBF as any function that satisfies (4a). Since (4b) is *linear* in u, if $\mathcal{U} = \mathbb{R}^m$, any candidate CBF B also satisfies (4b) if $L_g B \not\equiv 0$. When a system is of high relative degree (i.e., $L_g B(x) \equiv 0$), Higher Order CBFs (HOCBFs)[36, 37] can be used.

The main challenge to proving that a candidate CBF also satisfies (4b) occurs for bounded control sets $\mathcal{U}$ due to actuator limits [11]. Finding a function B such that (4b) verifiably holds for *arbitrary* nonlinear dynamics and avoid sets $\mathcal{A}$ is a hard problem that can be solved using Hamilton-Jacobi (HJ) reachability [38]. However, HJ reachability is computationally expensive and impractical for systems with more than 5 dimensions [31]. Consequently, many works that propose CBFs do not consider actuator limits [4–7].

One can try to use HOCBFs for automated CBF synthesis of high relative degree systems. As we show next, this can be problematic in the presence of input constraints.

Challenges of HOCBFs on the Double Integrator. Consider the double integrator $\dot{p} = v, \dot{v} = a$, the simplest high relative degree system, with the safety constraint $p \geq 0$. The HOCBF *candidate* $B(x) = -v - \alpha p$ is a *valid* CBF if and only if $\alpha = 0$ (i.e., disallowing all negative velocities). All other choices of α intersect the true unsafe region and violate (4b) (see Fig. 2).

C. Neural CBFs

To address the challenges of designing a valid CBF by hand, recent works have proposed learning a CBF using neural networks [24, 25, 29].

A naive approach to approximating the CBF B with a neural network approximation B^θ is to encourage satisfying the CBF conditions (4) by minimizing a loss L that penalizes

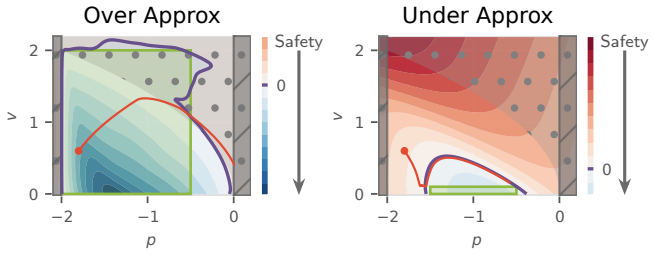

Fig. 3. **Over and Underestimation of the Safe Set.** When the safe set (in green) is *overestimated (Left)*, there are no valid CBFs that can satisfy all the loss terms in (11) simultaneously. Consequently, the Neural CBF B^θ has a zero level set (boundary in purple) that is larger than the true control-invariant set at the expense of violating the descent condition (4b). In contrast, when the safe set is *underestimated (Right)*, we can obtain a valid but overly conservative CBF. For comparison, the true unsafe set is shaded in gray with dots.

constraint violations over samples of the state space, i.e.,

$$L_{\text{unsf}}(\theta, x) = [\epsilon_{\text{unsf}} - B_\theta(x)]_+, \tag{6}$$

$$L_{\text{desc}}(\theta, x) = [L_f B_\theta(x) + L_g B_\theta(x)\pi(x) + cB_\theta(x)]_+, \tag{7}$$

$$L_1(\theta) = \sum_{x \in \mathcal{X}_{\text{unsf}}} L_{\text{unsf}}(\theta, x) + \sum_{x \in \mathcal{X}} L_{\text{desc}}(\theta, x), \tag{8}$$

where $\epsilon_{\text{unsf}} > 0$ for the strict inequality in (4a), $\alpha(\cdot)$ is chosen to be linear $x \mapsto cx$ for some $c > 0$, and $\mathcal{X}_{\text{unsf}}$ denotes some superset of $\mathcal{A}$. Successful minimization (i.e., zero loss) of (6) implies (4a), and similarly for (7) and (4b).

However, one problem is that the minimizer of (8) may have a small or even empty forward-invariant set. For example, let $\hat{B}$ be an exponential control-Lyapunov function, i.e.,

$$\inf_{u \in \mathcal{U}} L_f \hat{B}(x) + L_g \hat{B}(x) u + \hat{c}\hat{B}(x) \leq 0, \quad \hat{c} > c. \tag{9}$$

Then, $\hat{B} + d$ for all $d > 0$ small enough will also have zero loss on (8). However, the forward-invariant set of $\hat{B} + d$ is the empty set, and hence is not a useful CBF.

To address this challenge, many previous works additionally consider a loss term that enforces that $B_\theta \leq 0$ on some safe set $\mathcal{X}_{\text{safe}}$ [12, 25, 26, 39], i.e.,

$$L_{\text{safe}}(\theta, x) = [B_\theta(x)]_+, \tag{10}$$

$$L_2(\theta) = \sum_{x \in \mathcal{X}_{\text{safe}}} L_{\text{safe}}(\theta, x) + L_1(\theta). \tag{11}$$

However, the difficulty here is in finding the set $\mathcal{X}_{\text{safe}}$. In [18, 26], this is taken to be the set of initial conditions. In [12, 24], this is assumed to be available, but no details are given for how this set is found in practice. In [25], this set is evaluated by rolling out the test policy for a fixed number of timesteps. For all these cases, it is not clear whether a valid CBF B^θ exists such that $B^\theta < 0$ on $\mathcal{X}_{\text{safe}}$. The largest-possible $\mathcal{X}_{\text{safe}}$ from which a valid CBF can still be found can be obtained using reachability analysis [31]. However, the solution of the HJ reachabilty problem yields a CBF directly, rendering the NCBF unnecessary. Choosing an $\mathcal{X}_{\text{safe}}$ that is too large compromises the safety of the resulting CBF, while choosing an $\mathcal{X}_{\text{safe}}$ that is too small often results in a forward-invariant set that is too small (see Fig. 3).

An attempt to combat this issue is presented in [29], where a regularization term is added to the loss function to enlarge the sublevel set of the learned CBF B^θ. One issue with this approach is that this regularization term takes a nonzero value for any CBF (including the CBF with the largest zero sublevel set). Hence, the coefficient on this regularization term induces a trade-off between the size of the sublevel set and satisfaction of the CBF constraints (4a) and (4b). We provide comparisons against this method in the experiments section Section IV.

III. Policy Neural CBFs

To bypass the above challenges of training a Neural CBF, we now propose policy neural CBFs (PNCBFs), a different approach that does not require knowledge of the safe set but can still recover a large forward-invariant set.

A. Constructing CBFs via Policy Evaluation

We assume that the avoid set $\mathcal{A}$ can be described as the superlevel set of some continuous function $h : \mathcal{X} \to \mathbb{R}$, i.e.,

$$\mathcal{A} = \{\, x \in \mathcal{X} \mid h(x) > 0 \,\}. \tag{12}$$

Let $\pi : \mathcal{X} \to \mathcal{U}$ be an arbitrary policy, and let x_t^π denote the resulting state at time t following π. Consider the following *maximum-over-time* cost function

$$V^{h,\pi}(x_0) := \sup_{t \geq 0} h(x_t^\pi). \tag{13}$$

It can be shown that $V^{h,\pi}$ satisfies the following Hamilton-Jacobi PDE in the viscosity sense [40].

$$\max\left\{h(x) - V^{h,\pi}(x),\ \nabla V^{h,\pi}(x)^\mathsf{T}\left(f(x) + g(x)\pi(x)\right)\right\} = 0. \tag{14}$$

This immediately gives us the following two inequalities

$$V^{h,\pi}(x) \geq h(x), \tag{15a}$$

$$\nabla V^{h,\pi}(x)^\mathsf{T}\left(f(x) + g(x)\pi(x)\right) \leq 0, \tag{15b}$$

from which we have the following theorem.

Theorem 1 (Policy value function is a CBF)**.** *The policy value function $V^{h,\pi}$ is a CBF for* (1) *for any π and $\alpha > 0$.*

Proof. (15a) and (12) implies (4a). Next, (15b) implies (4b) for *any* choice of α, since $V(x) \leq 0$ implies that

$$\nabla V^{h,\pi}(x)^\mathsf{T}\left(f(x) + g(x)\pi(x)\right) \leq 0 \leq -\alpha(V(x)). \qquad \square$$

Intuitively, the policy value function $V^{h,\pi}$ gives us an upper-bound on the worst constraint violation h in the future under the optimal policy, since using π guarantees that h will be at most $V^{h,\pi}$, and the optimal policy will do no worse. Moreover, by following the negative gradient of $V^{h,\pi}$, we can move to states where following π leads to a lower maximum value of h, i.e., safer states (see Fig. 4).

Consequently, this provides us with a method to construct CBFs via policy evaluation of *any* policy π. To make this more concrete, consider the dynamic-programming form of (13):

$$V^{h,\pi}(x_0) = \max\left\{\sup_{0 \leq s \leq T} h(x_s),\ V^{h,\pi}(x_T)\right\}. \tag{16}$$

Algorithm 1 Policy Neural CBF

input: Train Policy π
Collect dataset of tuples $\left(x_t,\, \max_{t\le s\le T} h(x_s),\, x_T\right)$
while not converged **do**
    Minimize loss (17) over samples from the dataset
end while

Given a nominal policy π, we can collect rollouts of the system and store tuples $(x_t,\, \max_{t\le s\le T} h(x_s),\, x_T)$ for different t. We then minimize the policy evaluation loss on a neural network approximation of the policy value function $V_\theta^{h,\pi}$

$$L = \left\| V_\theta^{h,\pi}(x_t) - \max\left\{ \max_{t\le s\le T} h(x_s),\, V_\theta^{h,\pi}(x_T) \right\} \right\|^2. \quad (17)$$

We summarize the above for training PNCBFs in Algorithm 1. After training, we can use $V_\theta^{h,\pi}$ via the CBF-QP (5) to minimally modify *any* (unsafe) test policy to maintain safety.

Viewing policy CBFs as policy distillation. One can interpret policy value functions as policy distillation. More specifically, when $V^{h,\pi}$ is used as a safety filter in the CBF-QP (5) with any *new* test policy $\tilde{\pi}$, the forward-invariant set of the resulting CBF-QP controller will be no smaller than that of the original train policy π, as we show next.

Theorem 2. *Let $V^{h,\pi}$ be a policy value function and let $\tilde{\pi}$ be some other policy. Then, the forward-invariant set under CBF-QP with $V^{h,\pi}$ and $\tilde{\pi}$ is a superset of the forward-invariant set under π.*

Proof. The forward-invariant set under π is exactly the zero sublevel set of $V^{h,\pi}$ $\left\{ x \;\middle|\; V^{h,\pi} \le 0 \right\}$. Since $V^{h,\pi}$ is a CBF, the CBF-QP controller will render this set forward-invariant under any *new* nominal policy $\tilde{\pi}$. □

Relationship with Hamilton-Jacobi Reachability. The policy CBF is also closely related to HJ reachability. As noted in [32, 41, 42], the (optimal) HJ value function is a CBF. This is equivalent to the policy CBF (13) with the optimal policy π^*. The policy CBF can thus be seen as a *relaxation* of optimality that remains a CBF.

For neural networks, policy evaluation can be more attractive than optimization, which requires techniques such as deep reinforcement learning (e.g., [43–45]) that can be more unstable and computationally expensive. However, as a middle ground, we next show how policy iteration can be applied to PNCBFs to achieve fast convergence without resorting to a full deep reinforcement learning setup.

B. Policy Iteration with PNCBFs

The choice of the train policy π is crucial. In light of Theorem 2, the forward invariant set of the resulting PNCBF controller (via the CBF-QP (5)) is only guaranteed to be no smaller than that of π. Hence, a poor choice of π can result in a small forward-invariant set, resulting in a poor CBF.

To resolve this problem, we use the insight that the policy value function (13) is also a (shifted) Lyapunov function.

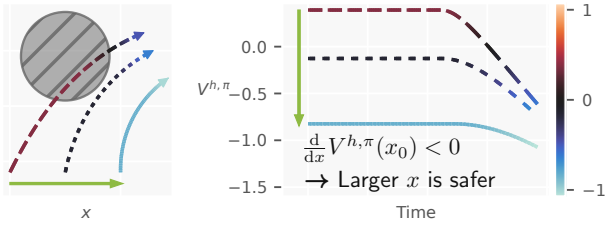

Fig. 4. **Understanding the policy value function.** *(Left)* Trajectories from a train policy π started from three values of x_0. *(Right)* The corresponding policy value functions $V^{h,\pi}$ along each trajectory. $V^{h,\pi}$ is non-increasing along (any) trajectory of π and is a CBF. Hence, the gradients of $V^{h,\pi}$ inform the CBF-QP on how to improve safety using the "knowledge" of the π.

Hence, when using $V^{h,\pi}$ with the CBF-QP (5), we can hope that the resulting forward-invariant set will be larger than the original policy π. Nevertheless, this new controller will be no worse than the original policy π. Thus, we propose to take the PNCBF controller as the *new* train policy π^+ to train a new PNCBF, and iterate this procedure (see Fig. 5). By treating the application of a CBF-QP as an analytical *policy improvement* and the computation of $V^{h,\pi}$ as policy evaluation, we can interpret this procedure as *policy iteration*, which has been studied extensively for the normal sum-over-time cost structure [46] where it enjoys guaranteed convergence at a superlinear rate under certain assumptions [47]. While it is not clear if this convergence result holds for the maximum-over-time cost structure, we empirically observe fast convergence in only a few iterations, as we show in Section IV-C. Also, we observe that using a policy value function with a non-zero discount factor can help with convergence when π is far from optimal. We leave an analysis of the interaction between the discount factor and convergence rates to future work.

C. Discounting and Contraction

One problem with using (17) directly as a loss function is that there are undesireable solutions that satisfy this recursive equation. For example, $V^{h,\pi}(x) = a$ for a large enough minimizes (17), but is clearly not a solution to (13). This is similar to the case in Markov Decision Processes where the *undiscounted* value iteration is not contractive [48]. Hence, instead of (13), we consider the following *discounted* cost, for $\lambda \ge 0$.

$$V_\lambda^{h,\pi}(x_0) := \sup_{t\ge 0} \left\{ \tilde{h}(x_t, \lambda) + e^{-\lambda t} h(x_t) \right\}, \quad (18)$$

$$\tilde{h}(x_t, \lambda) := \int_0^t \lambda e^{-\lambda s} h(x_s) \mathrm{d}s. \quad (19)$$

Taking $\lambda = 0$ recovers the undiscounted problem (13), while $\lambda \to \infty$ yields the solution $V_\infty^{h,\pi} = h$. Hence, different choices of λ can be seen as *implicitly* choosing the horizon considered for safety. Similar to (14), it can also be shown that $V_\lambda^{h,\pi}$ satisfies the following Hamilton-Jacobi PDE in the viscosity sense (suppressing arguments for conciseness) [40]:

$$\max\left\{ h - V_\lambda^{h,\pi},\; \nabla V_\lambda^{h,\pi\mathsf{T}} (f + g\pi) - \lambda(V_\lambda^{h,\pi} - h) \right\} = 0, \quad (20)$$

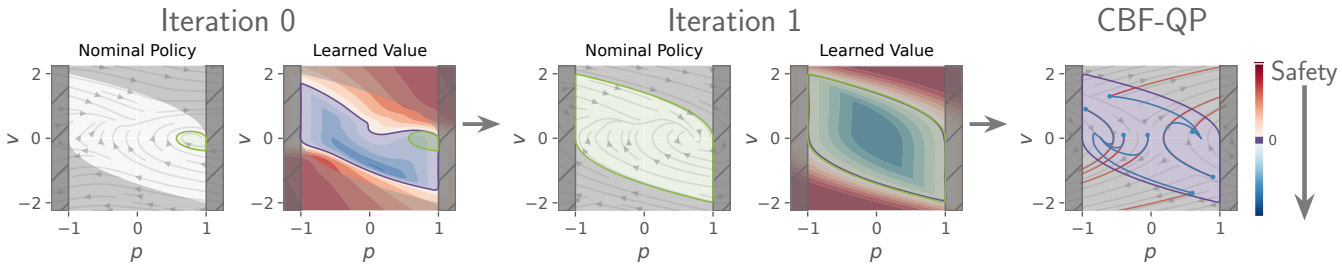

Fig. 5. **Policy iteration on a double integrator.** Starting with a suboptimal nominal policy π, we learn the value function $V^{h,\pi}$. By treating the CBF-QP of the learned $V^{h,\pi}$ as a new nominal policy, we can repeat this process to perform *policy iteration*. Here, only two iterations are needed to obtain a CBF $V^{h,\pi}$ that almost covers the true control-invariant set. The final CBF-QP controller maintains safety (blue line) under any potentially unsafe nominal policy (red line).

as well as the following dynamic programming equation

$$V_\lambda^{h,\pi}(x_0) = \max\left\{\sup_{0\le s\le t} \tilde{h}(x_s,\lambda), \tilde{h}(x_t,\lambda) + e^{-\lambda t} V_\lambda^{h,\pi}(x_t)\right\}. \tag{21}$$

While solutions to the PDE (20) no longer satisfy the CBF constraint (4b) for $\lambda > 0$, they do prevent the constant solution from being a minimizer of the corresponding discounted loss. Hence, in practice, we use (21) instead of (17). We start with a small value of λ to avoid premature convergence to the constant solution, and gradually decrease it to 0 as training progresses.

Verification of PNCBF. We stress that, without verifying that the learned PNCBF satisfies the descent condition (4b), we can not claim that the PNCBF satisfies the CBF conditions nor claim any safety guarantees. Verification of NCBFs can be performed using neural-network verification tools [49], sampling [50] or a generalization error bound [26]. However, these tools face scalability issues [49], with current methods possibly not able to verify the learned PNCBF.

Nevertheless, as we show next, empirical results show that our proposed method vastly improves the volume of both the forward-invariant set and the set where the safety filter is permissible to nominal controls compared to baseline methods, including an (unverified) HOCBF candidate.

IV. Simulation Experiments

To study the performance of PNCBFs, we perform a series of simulation experiments on high relative degree systems under box control constraints.

Baselines. We compare against the following safety filters.

- **Neural CBF (NCBF) [12, 24]:** Learning a Neural CBF using (11). We choose the safe set to be the set containing the equilibrium point under the train policy π.
- **Non-Saturating Neural CBF (NSCBF) [29]:** A recent approach that explicitly tackles the problem of input constraints for CBFs by learning a Neural CBF. However, instead of enforcing the derivative condition (4b) over the entire state-space as in [24], this is only enforced on the boundary as in barrier certificates [51].
- **Handcrafted Candidate CBF (CBF) [36, 37]:** We construct a *candidate* CBF via a Higher-Order CBF on h without considering input constraints.
- **Approximate MPC-based Predictive Safety Filter (MPC) [52]:** A trajectory optimization problem is solved, imposing the safety constraints while penalizing deviations from the test policy. We do not assume access to a known forward-invariant set and hence do not impose this terminal constraint.
- **Sum-of-Squares Synthesis (SOS) [53]:** When the dynamics are polynomial, a sequence of convex optimization problem can be solved to construct a CBF.

All neural networks are trained until convergence and use 3 layers of 256 neurons with $\tanh$ activations. For PNCBFs, we perform at most 3 iterations of policy iteration.

A. Qualitative Behavior on a Double Integrator, Segway

We first perform a general comparison between the different methods on a double integrator and a Segway, two simple systems that can be easily visualized. On the double integrator, safety is defined via position bounds ($|p| \le 1$), while the Segway must stay upright ($|\theta| \le 0.3\pi$) and remain within position bounds ($|p| \le 2.0$). We use a different test policy (i.e., zero control) than the policy used during training for PNCBFs.

We visualize the results in Fig. 6, plotting the region of the state space from where the safety filter preserves safety (Safe Region) and where the test policy can influence the output of the safety filter (Filter Boundary). For CBF-based methods, the filter boundary corresponds to the zero level set of the CBF. On the double integrator, all methods induce forward-invariance on some region of the state space but only our method is both maximally safe and permissive. This trend is even more pronounced on the Segway, where our method is able to find a significantly larger safe set and filter boundary.

B. Scalability to high-dimensional systems with a jet aircraft

Next, we explore the scalability of PNCBF to high dimensional systems. We consider a ground collision avoidance example involving a jet aircraft [54, 55]. Since this system is not control-affine in the throttle, we leave the throttle as the output of a P controller, resulting in a 16-dimensional state space and a 3-dimensional control space. We define safety as a box constraint on the aircraft's altitude. During testing, we apply an adversarial test policy that commands the aircraft to dive nose-first into the ground.

We visualize the results in Fig. 6, showing a 2D slice of the state space. Even on a 16-dimensional state space, we

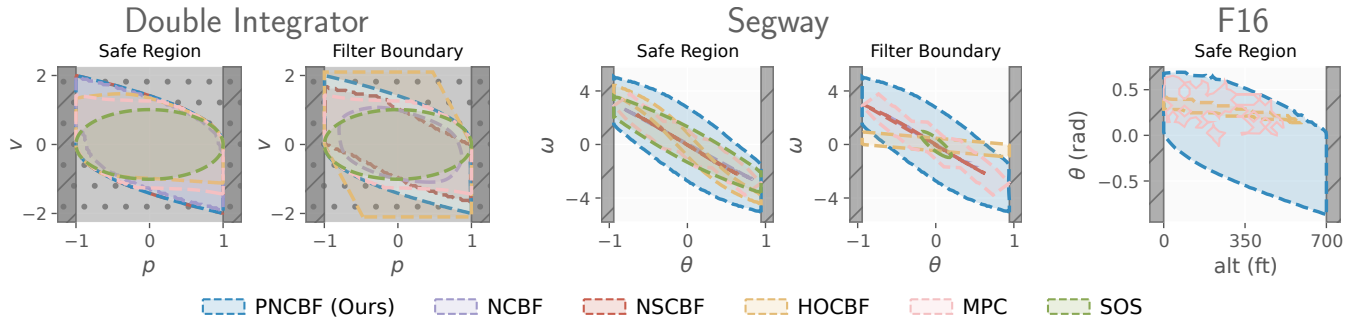

Fig. 6. **Safe set and filter boundary on the double integrator, Segway, and F16** We plot the initial states from where the safety filter can preserve safety (Safe Region), and states where the nominal policy can influence the output of the safety filter (Filter Boundary). The true unsafe region is shaded in gray dots. On the double integrator, ours is the only method that is both maximally safe and permissive. For more complex systems, the performance gap between our method and baseline methods becomes more pronounced, showcasing the benefit of PNCBFs on high-dimensional nonlinear systems.

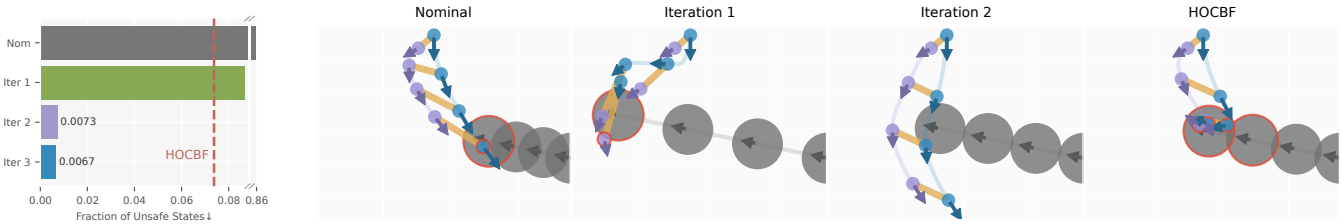

Fig. 7. **Policy Iteration on a Two-Agent Quadcopter System.** *(Left)* In only three iterations, we achieve the smallest volume of unsafe states compared to baseline methods and greatly improving the safety of the original unsafe nominal policy. Since we may sample states in the true unsafe region, the optimal safety filter will not be able to achieve safety for all sampled states. *(Right)* An example of an initial state from which only our method is able to filter out unsafe controls from the nominal policy and prevent a collision (highlighted in red) with the moving obstacle.

observe that PNCBFs are able to recover a significantly larger region of the safe set compared to other baseline methods.

C. Performance of Policy Iteration

Finally, to investigate the ability of PNCBFs to learn a safe and permissible safety filter from an initially unsafe train policy, we consider a two-agent quadcopter system with a 12D state and 4D control space that must stay within communication radius while avoiding collisions with a dynamic obstacle. We model each quadcopter as a double integrator with a velocity tracking controller. The obstacle is assumed to move with constant velocity and direction. The train policy moves each quadcopter anticlockwise around a circle, ignoring all constraints. The obstacle can achieve higher velocities than each quadcopter. Additionally, the velocity tracking controller has a slow response time. Hence, the quadcopters must react well in advance to avoid collisions with the obstacle while staying within communication radius, resulting in a problem with complex safety constraints despite the simple dynamics.

Although the train policy is unsafe, policy iteration is able to significantly reduce the unsafe fraction to near 0 in only two iterations, representing a 90% reduction in unsafe states compared to the next best method (see Fig. 7).

V. Hardware Experiments

We further validate our approach in a two-agent quadrotor hardware experiment mirroring Section IV-C. We use two custom drones and use Boston Dynamics's Spot as a dynamic obstacle. Velocity setpoints are sent to the drones through the PX4 flight stack. The PNCBF filters the drone's unsafe test policy to avoid collisions with Spot while remaining within communication radius (Fig. 8).

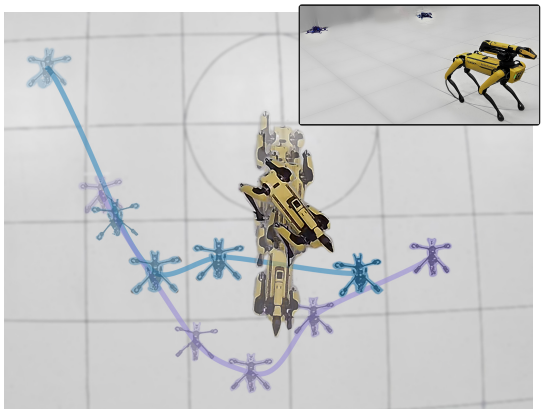

Fig. 8. **Two-agent Quadrotor System with Moving Quadruped Obstacle.** Snapshots from a hardware realization of the setup from Section IV-C.

VI. Discussion and Conclusion

By learning the policy value function, we are able to learn Neural CBFs for high relative degree systems under input constraints. Extensive simulation experiments show that our method is more scalable and can yield much larger forward invariant sets compared to existing methods of constructing safety filters, while hardware experiments suggest the robustness of our method to noise in physical systems.

One limitation of our method is that it requires an accurate dynamics model. Model errors may cause the learned safety filter to be unsafe on the real system. While this was not a major issue in our hardware experiments, we plan to investigate methods to improve robustness to model errors in future work, such as by learning *robust* CBFs as in [24], or by incorporating this into a reinforcement learning setup simuilar to [45].

References

[1] J. Betz, A. Heilmeier, A. Wischnewski, T. Stahl, and M. Lienkamp, "Autonomous driving—a crash explained in detail," *Applied Sciences*, vol. 9, no. 23, p. 5126, 2019.

[2] T. Haidegger, "Autonomy for surgical robots: Concepts and paradigms," *IEEE Transactions on Medical Robotics and Bionics*, vol. 1, no. 2, pp. 65–76, 2019.

[3] M. Connors, "Understanding risk in urban air mobility: Moving towards safe operating standards," NASA, Ames Research Center, Technical Memorandum NASA/TM-2020-5000604, February 2020.

[4] B. Xu and K. Sreenath, "Safe teleoperation of dynamic uavs through control barrier functions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7848–7855.

[5] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE control systems letters*, vol. 3, no. 1, pp. 96–101, 2018.

[6] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.

[7] M. A. Pereira, A. D. Saravanos, O. So, and E. A. Theodorou, "Decentralized safe multi-agent stochastic optimal control using deep fbsdes and admm," *arXiv preprint arXiv:2202.10658*, 2022.

[8] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron, and A. D. Ames, "A scalable safety critical control framework for nonlinear systems," *IEEE Access*, vol. 8, pp. 187 249–187 275, 2020.

[9] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, "Backup control barrier functions: Formulation and comparative study," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6835–6841.

[10] E. Squires, P. Pierpaoli, and M. Egerstedt, "Constructive barrier certificates with applications to fixed-wing aircraft collision avoidance," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2018, pp. 1656–1661.

[11] J. Breeden and D. Panagou, "High relative degree control barrier functions under input constraints," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6119–6124.

[12] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods," *arXiv preprint arXiv:2202.11762*, 2022.

[13] Y.-C. Chang, N. Roohi, and S. Gao, "Neural lyapunov control," *Advances in neural information processing systems*, vol. 32, 2019.

[14] J. V. Deshmukh, J. P. Kapinski, T. Yamaguchi, and D. Prokhorov, "Learning deep neural network controllers for dynamical systems with safety guarantees," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–7.

[15] M. Saveriano and D. Lee, "Learning barrier functions for constrained motion planning with dynamical systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 112–119.

[16] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela, "Synthesis of control barrier functions using a supervised machine learning approach," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7139–7145.

[17] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 3717–3724.

[18] A. Peruffo, D. Ahmed, and A. Abate, "Automated and formal synthesis of neural barrier certificates for dynamical models," in *International conference on tools and algorithms for the construction and analysis of systems*. Springer, 2021, pp. 370–388.

[19] Z. Yang, Y. Zhang, W. Lin, X. Zeng, X. Tang, Z. Zeng, and Z. Liu, "An iterative scheme of safe reinforcement learning for nonlinear systems via barrier certificate generation," in *International Conference on Computer Aided Verification*. Springer, 2021, pp. 467–490.

[20] H. Zhao, X. Zeng, T. Chen, Z. Liu, and J. Woodcock, "Learning safe neural network controllers with barrier certificates," *Formal Aspects of Computing*, vol. 33, pp. 437–455, 2021.

[21] L. Lindemann, H. Hu, A. Robey, H. Zhang, D. Dimarogonas, S. Tu, and N. Matni, "Learning hybrid control barrier functions from data," in *Conference on Robot Learning*. PMLR, 2021, pp. 1351–1370.

[22] W. S. Cortez, J. Drgona, A. Tuor, M. Halappanavar, and D. Vrabie, "Differentiable predictive control with safety guarantees: A control barrier function approach," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 932–938.

[23] K. Garg, S. Zhang, O. So, C. Dawson, and C. Fan, "Learning safe control for multi-robot systems: Methods, verification, and open challenges," *arXiv preprint arXiv:2311.13714*, 2023.

[24] C. Dawson, Z. Qin, S. Gao, and C. Fan, "Safe nonlinear control using robust neural lyapunov-barrier functions," in *Conference on Robot Learning*. PMLR, 2022, pp. 1724–1735.

[25] H. Yu, C. Hirayama, C. Yu, S. Herbert, and S. Gao, "Sequential neural barriers for scalable dynamic obstacle avoidance," *arXiv preprint arXiv:2307.03015*, 2023.

[26] 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.

[27] S. Zhang, K. Garg, and C. Fan, "Neural graph control barrier functions guided distributed collision-avoidance multi-agent control," in *7th Annual Conference on*

Robot Learning, 2023.

[28] S. Zhang, O. So, K. Garg, and C. Fan, "Gcbf+: A neural graph control barrier function framework for distributed safe multi-agent control," *arXiv preprint arXiv:2401.14554*, 2024.

[29] S. Liu, C. Liu, and J. Dolan, "Safe control under input limits with neural control barrier functions," in *Conference on Robot Learning*. PMLR, 2023, pp. 1970–1980.

[30] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[31] I. M. Mitchell, "The flexible, extensible and efficient toolbox of level set methods," *Journal of Scientific Computing*, vol. 35, pp. 300–329, 2008.

[32] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier–value functions for safety-critical control," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6814–6821.

[33] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[34] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[35] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[36] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 322–328.

[37] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *2019 IEEE 58th conference on decision and control (CDC)*. IEEE, 2019, pp. 474–479.

[38] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018, pp. 98–106.

[39] B. Dai, P. Krishnamurthy, and F. Khorrami, "Learning a better control barrier function," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 945–950.

[40] A. Altarovici, O. Bokanowski, and H. Zidani, "A general hamilton-jacobi framework for non-linear state-constrained control problems," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 19, no. 2, pp. 337–357, 2013.

[41] S. Tonkens and S. Herbert, "Refining control barrier functions through hamilton-jacobi reachability," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 355–13 362.

[42] S. Tonkens, A. Toofanian, Z. Qin, S. Gao, and S. Herbert, "Patching neural barrier functions using hamilton-jacobi reachability," *arXiv preprint arXiv:2304.09850*, 2023.

[43] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, S. Ghosh, and C. J. Tomlin, "Bridging hamilton-jacobi safety analysis and reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8550–8556.

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

[45] O. So and C. Fan, "Solving stabilize-avoid optimal control via epigraph form and deep reinforcement learning," *arXiv preprint arXiv:2305.14154*, 2023.

[46] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[47] M. S. Santos and J. Rust, "Convergence properties of policy iteration," *SIAM Journal on Control and Optimization*, vol. 42, no. 6, pp. 2094–2115, 2004.

[48] A. Federgruen, P. J. Schweitzer, and H. C. Tijms, "Contraction mappings underlying undiscounted markov decision problems," *Journal of Mathematical Analysis and Applications*, vol. 65, no. 3, pp. 711–730, 1978.

[49] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, M. J. Kochenderfer *et al.*, "Algorithms for verifying deep neural networks," *Foundations and Trends® in Optimization*, vol. 4, no. 3-4, pp. 244–404, 2021.

[50] R. Bobiti and M. Lazar, "Automated-sampling-based stability verification and doa estimation for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3659–3674, 2018.

[51] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *International Workshop on Hybrid Systems: Computation and Control*. Springer, 2004, pp. 477–492.

[52] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, 2021.

[53] P. Zhao, R. Ghabcheloo, Y. Cheng, H. Abdi, and N. Hovakimyan, "Convex synthesis of control barrier functions under input constraints," *IEEE Control Systems Letters*, 2023.

[54] P. Heidlauf, A. Collins, M. Bolender, and S. Bak, "Verification challenges in f-16 ground collision avoidance and other automated maneuvers." in *ARCH@ ADHS*, 2018, pp. 208–217.

[55] B. L. Stevens and F. L. Lewis, "Aircraft control and simulation, john willey& sons," *Inc., New York*, pp. 309–316, 1992.