Proceedings of Machine Learning Research vol vvv:1–19, 2024

Safety Filters for Black-Box Dynamical Systems by Learning Discriminating Hyperplanes

Will Lavanakul* LAV.WILL@BERKELEY.EDU
University of California, Berkeley

Jason J. Choi* JASON.CHOI@BERKELEY.EDU
University of California, Berkeley

Koushil Sreenath KOUSHILS@BERKELEY.EDU
University of California, Berkeley

Claire J. Tomlin TOMLIN@EECS.BERKELEY.EDU
University of California, Berkeley *

Abstract

Learning-based approaches are emerging as an effective approach for safety filters for black-box dynamical systems. Existing methods have relied on certificate functions like Control Barrier Functions (CBFs) and Hamilton-Jacobi (HJ) reachability value functions. The primary motivation for our work is the recognition that, ultimately, enforcing the safety constraint as a control input constraint at each state is what matters. By focusing on this constraint, we can eliminate dependence on any specific certificate function-based design. To achieve this, we define a *discriminating hyperplane* that shapes the half-space constraint on control input at each state, serving as a sufficient condition for safety. This concept not only generalizes over traditional safety methods but also simplifies safety filter design by eliminating dependence on specific certificate functions. We present two strategies to learn the discriminating hyperplane: (a) a supervised learning approach, using pre-verified control invariant sets for labeling, and (b) a reinforcement learning (RL) approach, which does not require such labels. The main advantage of our method, unlike conventional safe RL approaches, is the separation of performance and safety. This offers a reusable safety filter for learning new tasks, avoiding the need to retrain from scratch. As such, we believe that the new notion of the discriminating hyperplane offers a more generalizable direction towards designing safety filters, encompassing and extending existing certificate-function-based or safe RL methodologies.

Keywords: Safety filters, Safe learning, Safe reinforcement learning, Certificate functions

1. Introduction

While learning-based control has demonstrated capabilities in solving complex tasks for uncertain dynamical systems, safety assurance remains an unresolved challenge. Safe Reinforcement Learning (RL) has made strides towards co-learning performance and safety (Garcıa and Fernández, 2015), but the absence of provable guarantees in these data-driven methods remains a significant challenge. Conversely, although model-based control theory has well-established mechanisms to enforce safety constraints, its application to uncertain systems is limited when the models deviate from the actual system dynamics.

In an effort to bridge this gap and provide safety guarantees for black-box dynamical systems with unknown closed-form expressions, researchers are actively combining model-based control techniques with data-driven methods (Brunke et al., 2022a; Wabersich et al., 2023). Traditional

* * indicate co-first authors. Extended version.

model-based methods for safety problems fundamentally involve two key components: a safe set where the trajectory can remain indefinitely, and a control input constraint that guarantees this invariance within the set. These methods typically rely on certificate functions such as Lyapunov functions, Control Barrier Functions (Ames et al., 2016), or HJ reachability value functions (Fisac et al., 2019a). Consequently, existing learning-based methods for designing safety filters for black-box systems have primarily focused on learning the certificate functions (Dawson et al., 2023).

We endorse the philosophy that control theory-based analysis can play a crucial role in devising a data-driven scheme for safety control. However, the main motivation for our work comes from the realization that the enforcement of an appropriate control input constraint at each state is the ultimate step for guaranteeing safety. By focusing on the constraint, we eliminate the need to rely on a specific certificate function-based safe set representation and the input constraint form.

Towards this end, we define a *discriminating hyperplane* that shapes the half-space constraint on control input at each state and serves as a sufficient condition to ensure control invariance of a safe set. The discriminating hyperplane-based safety constraint generalizes the well-known Nagumo-like condition (Blanchini, 1999), CBF constraint (Ames et al., 2016), and HJ reachability optimal control law (Fisac et al., 2019a), for control-affine systems.

Next, we present two approaches for learning the discriminating hyperplane for black-box systems. The first method utilizes supervised learning, where the learning is guided by labels derived from a pre-verified control invariant set. Although this assumes access to prior knowledge of the invariant set, it can be well supplemented by leveraging existing research focused on constructing or learning control invariant sets (Bertsekas, 1972; Blanchini, 1999; Fisac et al., 2019a; Wabersich and Zeilinger, 2018; Bansal and Tomlin, 2021; Cao et al., 2022). In our simulations across various systems, we find that the safety filter, based on the learned discriminating hyperplane, closely approximates a filter that could be designed using ground-truth system dynamics.

The second method, utilizing reinforcement learning (RL), circumvents the need for access to control invariant sets by learning the discriminating hyperplane directly from trajectory data. We can employ existing RL algorithms, such as Proximal Policy Optimization (PPO) (Schulman et al., 2017), for this purpose. A key strength of this approach is the ability to separate a learning-based policy into two components–task achievement and safety filtering. This separation allows the learned safety filter to be reused or adapted to various performance tasks. The concept of separating performance and safety is well-supported by existing safe RL literature (Thananjeyan et al., 2021; Wagener et al., 2021; Kim et al., 2023b). The main distinction of our work is that there is no on-off switching mechanism between performance and safety policies. Instead, we integrate these two aspects into a unified safety filter design based on the discriminating hyperplane concept.

2. Related Work

Certificate functions: Many model-based control approaches for safety problems use the concept of certificate functions (Prajna, 2006; Dawson et al., 2023). These are also referred to as safety index in Liu and Tomizuka (2014) or energy function in Wei and Liu (2019). Put simply, a certificate function is a state-dependent scalar function, whose level sets characterize the safe domain and whose gradient can impose a constraint on the control input to ensure safety. To address difficulties in designing certificate functions using classical methods, learning-based methods are being actively explored (Fisac et al., 2019b; Srinivasan et al., 2020; Huh and Yang, 2020; Lindemann et al., 2021; Thananjeyan et al., 2021; Liu et al., 2023; So et al., 2023; Castañeda et al., 2023).

Learning constraints for uncertain systems: A crucial step in designing the safety filter is to come up with a valid constraint to be imposed on the control input. Usually, this constraint is a sufficient condition for the closed-loop trajectory to stay invariant inside the safe domain. These constraints can be derived from certificate functions. For uncertain systems, these are typically designed based on a nominal model, and are combined with data-driven methods to accommodate the modeling error (Choi et al., 2020; Castañeda et al., 2021; Taylor et al., 2020, 2021; Brunke et al., 2022a; Wabersich et al., 2023). Our methodology aligns closely with these techniques. However, crucially, our approach is independent of both certificate functions and nominal models.

Safe Reinforcement Learning (RL): Numerous safe RL algorithms proposed in Garcıa and Fernández (2015); Achiam et al. (2017); Ray et al. (2019); Srinivasan et al. (2020); Thananjeyan et al. (2021); Wagener et al. (2021) address safety using a purely data-driven approach. A majority of these methods construe safety violations as an *accumulation* of safety-relevant cost, referred to as the constrained Markov Decision Process problem (Altman, 1998). While such a formulation finds applicability in certain safety contexts, it falls short in capturing instantaneous constraint violations like collision. While some safe RL algorithms incorporate control theory concepts—for example, Lyapunov theory in Chow et al. (2018) and reachability theory in Fisac et al. (2019b); Huh and Yang (2020); Hsu et al. (2023)—these ideas are applied in the algorithmic design rather than in the structural design of the learned safe policy. In contrast, our approach imposes a structural knowledge derived from the discriminating hyperplane to the learned safety filter. Approaches in Cheng et al. (2019); Emam et al. (2022) explicitly use certificate functions to achieve this.

3. Discriminating Hyperplane

3.1. Safety filters for black-box dynamical systems

We are interested in guaranteeing safety for a black-box dynamical system, which is characterized by a state trajectory, $\mathrm{x}(t)$, a solution to an ordinary differential equation (ODE) with an initial condition $\mathrm{x}(0) = x$. Specifically, we make the following assumptions about the system dynamics.

Assumption 1 (System dynamics) The dynamics is *affine in control*, represented by an ODE

$$\dot{\mathrm{x}}(t) = f(\mathrm{x}(t)) + g(\mathrm{x}(t))\mathrm{u}(t) \quad \text{for} \quad t > 0, \qquad \mathrm{x}(0) = x, \tag{1}$$

where $x \in \mathbb{R}^n$ is an initial state, $\mathrm{x} : [0, \infty) \to \mathbb{R}^n$ is the solution to the ODE, and $\mathrm{u} : [0, \infty) \to U \subset \mathbb{R}^m$ is a control signal. $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are bounded Lipschitz continuous vector fields. The control input set U is compact.

These assumptions reflect the realistic conditions of real-world physical systems and set the minimum requirements for the structure of the systems under consideration. The control-affine nature of the dynamics applies to a variety of physical systems derived from Euler-Lagrangian mechanics, or the dynamics can often be transformed into a control-affine form through a change of variables. The last condition is typical of physical systems with bounded actuation limits.

In our framework, we do not assume access to the closed-form expressions of f and g. Instead, we assume the ability to sample state trajectories over discrete steps in time, through simulation or experiments. With this setup, we can efficiently use the collected state transition data to train the safety filter without requiring explicit knowledge of the underlying system dynamics.

The safety problems we focus on is to ensure that the system states satisfy specific constraints over an indefinite time horizon. Thus, safety for the system (1) is encoded by a *target constraint set* $X \subset \mathbb{R}^n$ that must be respected during the evolution of the system:

$$\mathrm{x}(t) \in X \text{ for all } t \geq 0. \tag{2}$$

The *safety filter* we aim to design operates as an intermediary between the reference controller $\pi_{\text{ref}} : \mathbb{R}^n \to U$ and the system, ensuring that the downstream control adheres to the safety constraint described in (2). Reference controllers are typically safety-agnostic and can take various forms, from a neural network policy optimized for performance objectives, hand-designed controllers from domain experts, to a human operator's commands. The safety filter $\pi_{\text{safe}} : \mathbb{R}^n \to U$ modifies the input signal from the reference controller when necessary, yielding a control input signal $\mathrm{u}(t) = \pi_{\text{safe}}(\mathrm{x}(t); \pi_{\text{ref}})$. We mainly focus on the *safety constraint* within the filter, which modifies the control input to guarantee safety:

Definition 1 (Safety constraint & safe domain for safety filter) We say that $c(x, u) \geq 0$ is a (valid) *safety constraint* if there exists a *safe domain* $S \subseteq X$, such that for all Lipschitz feedback policy $\pi : \mathbb{R}^n \to U$ that satisfies $c(x, \pi(x)) \geq 0$ for all $x \in S$, the trajectory resulting from the control signal $\mathrm{u}(t) = \pi(\mathrm{x}(t))$ satisfies (2) for all $x \in S$ where $\mathrm{x}(0) = x$.

Note that the Lipschitz continuity of π is required to guarantee the solution existence and uniqueness of $\mathrm{x}(\cdot)$. In words, the safety constraint is a sufficient condition that, when u satisfies the constraint, safety is guaranteed for all states in the safe domain S, a subset of the target constraint set X. Consequently, the effective design of a safety filter hinges on the choice of a valid safe domain S and a safety constraint $c(x, u) \geq 0$.

We first review how a safe domain is generally designed in the literature. Many existing design approaches (Wabersich et al., 2023) seek to find a set S that is control invariant, defined next.

Definition 2 (Control Invariance (Blanchini, 1999)) A set $S \subset \mathbb{R}^n$ in the state space is *control invariant* (under the dynamics (1)) if for all $x \in S$, there exists a control signal $\mathrm{u} \in \mathcal{U}$ such that

$$\mathrm{x}(t) \in S \text{ for all } t \geq 0. \tag{3}$$

The definition implies that a control invariant set S that is a subset of X can serve as a safe domain. The representation of the control invariant set can vary significantly depending on the chosen design methodology. It ranges from geometric structures like polytopes (Blanchini, 1999) to level sets of scalar functions such as certificate functions (Dawson et al., 2023), and extends to the concept of feasibility in receding-horizon optimal control (Wabersich and Zeilinger, 2021). This versatility underscores control invariance as a foundational concept in designing safe domains.

3.2. Discriminating hyperplane for safety constraint

Once a control invariant safe domain S is specified, a similar blueprint for characterizing a valid safety constraint exists. It is based on the geometric relationship that the control invariant set and the vector field of the dynamics satisfy at its boundary, ∂S, which is known as the "Nagumo-like" condition in the literature (Nagumo, 1942; Blanchini, 1999; Aubin et al., 2011; Ames et al., 2016):

Lemma 1 (Tangential characterization of control invariant sets Aubin et al. (2011, Theorem 11.3.4)) Let the dynamics (1) satisfy Assumption 1. Then, a closed set $S \subset \mathbb{R}^n$ is control invariant if and only if for all $x \in \partial S$,

$$\exists u \in U \text{ such that } \frac{\partial h}{\partial x}(x) \cdot (f(x) + g(x)u) \geq 0, \tag{4}$$

where $h:\mathbb{R}^n \to \mathbb{R}$ is continuously differentiable function such that $S=\{x|h(x) \geq 0\}$ and $\frac{\partial h}{\partial x}(x)$ is bounded away from 0 for all $x \in \partial S$, implying that the slope of h does not vanish at the boundary[1].

The condition (4) implies that for the set to be control invariant, there must exist a control input that renders the vector field pointing inward to the set. In fact, the condition is a geometric property that remains invariant under different choices of the distance-like function h, and (4) is merely its analytic description[2]. Extending this property, in the new notion of the discriminating hyperplane we introduce next, the resulting safety constraint also does not rely on any choice of h. In contrast, the notion of CBF in (Ames et al., 2016), which is also inspired by Lemma 1, results in a certifying constraint that is dependent on h.

From (4), a constraint which is affine in u, imposed whenever the state x is at the boundary of S, is sufficient to regulate the trajectory to not exit the set S. This serves as a starting point for constructing a control-affine constraint extended to all $x \in S$ including the interior of the set. By noticing that the control-affine inequality defines a halfspace in the control input space, we define the discriminating hyperplane as below:

Definition 3 A *discriminating hyperplane* for systems satisfying Assumption 1 and a control invariant set $S \in \mathbb{R}^n$ is a hyperplane defined in the control input space represented by $a(x)^\top u = b(x)$, for each state $x \in S$, such that the resulting half-space constraint $a(x)^\top u \geq b(x)$ is a safety constraint according to Definition 1.

In words, the discriminating hyperplane discriminates the control input between certified input that guarantees safety, and uncertified input that can potentially lead to safety violations. The hyperplane is defined in the control input space and is parameterized by each state. If such a discriminating hyperplane can be determined, we can use it effectively to construct the following safety filter:

Discriminating hyperplane-based min-norm safety filter:

$$\begin{aligned} \pi_{\text{safe}}(x) = \arg\min_{u \in U} \quad & ||u - \pi_{\text{ref}}(x)||^2 \\ \text{s.t.} \quad & a(x)^\top u \geq b(x) \end{aligned} \tag{5}$$

Note that this filter selects a control input that is certified to be safe by the discriminating hyperplane and is closest to the reference control input $\pi_{\text{ref}}(x)$. The filter becomes a quadratic program when U is a polytope. When U is a general convex set, the program is still a convex program.

Next, we present a sufficient condition for $a(x)^\top u = b(x)$ to be a discriminating hyperplane. In words, the theorem says that any discriminating hyperplane leading to the satisfaction of the Nagumo-like condition (4) is valid.

Theorem 1 For the control input set U that is polytopic, if $a : \mathbb{R}^n \to \mathbb{R}^m$ and $b : \mathbb{R}^n \to \mathbb{R}$ are Lipschitz continuous in x, and if for all $x \in \partial S$, $\{u \in U \mid a(x)^\top u \geq b(x)\}$ has a nonempty interior and is a subset of $\{u \in U \mid \frac{\partial h}{\partial x}(x) \cdot (f(x) + g(x)u) \geq 0\}$, then a and b define a discriminating hyperplane, $\{u \mid a(x)^\top u = b(x)\}$. (Proof: Appendix 7.1)

An important lemma for the theorem's proof establishes that π_{safe} in (5) is the feedback policy that shows the validity of $a(x)^\top u \geq b(x)$ as a safety constraint satisfying Definition 1:

1. Such h exists for S whose interior is not empty and boundary is continuously differentiable (Lieberman, 1985).
2. (4) can be rewritten as $\exists u \in U$ s.t. $(f(x)+g(x)u) \in T_S(x)$, where T_S is (Bouligand's) tangent cone to S (Clarke et al., 2008).

Lemma 2 If $\pi_{\text{ref}} : \mathbb{R}^n \to U$ is Lipschitz continuous in x and if a, b satisfy the conditions in Theorem 1, then π_{safe} in (5) is also Lipschitz continuous in x.

Remark 1 CBFs in (Ames et al., 2016) and HJ reachability value functions in (Fisac et al., 2019a) offer special cases of the discriminating hyperplane. For a CBF h, the hyperplane is given by $a(x) = \frac{\partial h}{\partial x}(x) \cdot g(x)$ and $b(x) = -\frac{\partial h}{\partial x}(x) \cdot f(x) - \alpha(h(x))$, where α is the comparison function associated with the CBF. For a reachability value function V, the hyperplane is given by $a(x) = \frac{\partial V}{\partial x}(x) \cdot g(x)$ and $b(x) = -\frac{\partial V}{\partial x}(x) \cdot f(x)$. As such, the discriminating hyperplane can be considered as a generalized structure of safety constraints for control-affine systems that unifies and extends the results of the existing methods.

3.3. Sample-and-hold lookahead-based discriminating hyperplane

We present a simple way to construct a discriminating hyperplane for a given control invariant set S, without having to rely on any specific distance-like function h of the set S. This is consistent with the core principle of our approach, which is to avoid reliance on any specific form of certificate functions or safe domain representation. Thus, we only rely on a minimal knowledge of the invariant set: an indicator of whether a given state is within S: $I_S(x) = 1$ if $x \in S$, and 0 otherwise.

The method we propose is to construct the discriminating hyperplane by checking whether the state trajectory exits the set S for a lookahead time Δt, under sample-and-hold of control input u. This approach is endorsed by the following theorem:

Theorem 2 For $\Delta t > 0$, there exists $a : \mathbb{R}^n \to \mathbb{R}^m$ and $b : \mathbb{R}^n \to \mathbb{R}$ that are Lipschitz continuous in x, such that $\Pi(x) = \{u \in U \mid a(x)^\top u \geq b(x)\} \subseteq \{u \in U \mid I_S(\mathrm{x}(\Delta t)) = 1\}$ where $\mathrm{x}(\cdot)$ is from (1) with $\mathrm{x}(0) = x, \mathrm{u}(\cdot) \equiv u$. Moreover, for small enough Δt, $\Pi(x)$ is not empty if $\{u \in U \mid I_S(\mathrm{x}(\Delta t)) = 1\}$ has a non-empty interior. (Proof: Appendix 7.2)

By using the lookahead approach, $\Pi(x)$ satisfying Theorem 2 anticipates into the future, which is also the main mechanism of CBFs (Choi et al., 2023) or predictive filter (Wabersich and Zeilinger, 2021). If the lookahead time is small, then the constraint would be active only on states that are close to the boundary. In contrast, larger Δt will encode additional safety margin from the boundary of S, however, it might make $\Pi(x)$ an empty set. In Section 5, we demonstrate how this design results in a "smooth braking" behavior as $\mathrm{x}(\cdot)$ approaches the boundary of the safe domain, similarly to the CBF-based design, and how different choice of the lookahead time affects this behavior.

4. Learning Discriminating Hyperplane

This section presents supervised and reinforcement learning approaches to learn the discriminating hyperplane from black-box system trajectory data.

4.1. Supervised learning approach

In our supervised learning approach, we use the lookahead-based discriminating hyperplane in Section 3.3 to construct a training label for the neural network that learns the hyperplane. We train $[a_\theta(x), b_\theta(x)] \in \mathbb{R}^{m+1}$, where θ is the neural network weights, in a supervised manner using a dataset of state and input transition pairs, and their labels detailing if an input is safe. Specifically, N states, $\{x_i\}_{i=1}^N$, are uniformly sampled from the control invariant set S. Then M inputs per state,

sampled uniformly from U, $\{u_{ij}\}_{j=1}^{M}$, are applied to the dynamical system by sample-and-hold for the lookahead time Δt, resulting in the terminal state $\mathrm{x}_{ij}(\Delta t)$. The indicator function of the set S at the terminal state, $I_S(\mathrm{x}_{ij}(\Delta t))$, determines the label of whether the next state is safe or unsafe:

$$y_{ij} = \begin{cases} 1 & \text{if } I_S\left(\mathrm{x}_{ij}(\Delta t)\right) = 1 \text{ where } \mathrm{x}_{ij} \text{ solves (1) for } \mathrm{x}_{ij}(0) = x_i, \mathrm{u} \equiv u_{ij}, \\ -1 & \text{else.} \end{cases} \tag{6}$$

In essence, we are labeling the inputs based on whether they leave S after the lookahead time.

In order for the neural network hyperplane to be a valid discriminating hyperplane, any control input that is predicted to be safe, i.e. $a_\theta(x_i)^\top u_{ij} \geq b_\theta(x_i)$, should have the label $y_{ij} = 1$. As such, we aim to minimize the misclassification rate of the neural network hyperplane, which can be achieved by minimizing the following loss function:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\gamma_{\text{pos}} \sum_{\substack{j: \\ a_\theta(x_i)^\top u_{ij} > b_\theta(x_i)}} -\min\Big\{y_{ij}(a_\theta(x_i)^\top u_{ij} - b_\theta(x_i)), 0\Big\} + \gamma_{\text{neg}} \sum_{\substack{j: \\ a_\theta(x_i)^\top u_{ij} < b_\theta(x_i)}} -\min\Big\{y_{ij}(a_\theta(x_i)^\top u_{ij} - b_\theta(x_i)), 0\Big\}\right).$$

If the prediction of safety from the neural network matches the labels, the loss is not incurred. False positive samples whose unsafe control input is predicted to be safe are counted in the first term, and false negative samples whose safe control input is predicted to be unsafe are counted in the second term, where γ_{pos} and γ_{neg} are weights for false positive and negative samples. We typically set $\gamma_{\text{pos}} > \gamma_{\text{neg}}$ as it is crucial to rule out unsafe inputs correctly.

4.2. Reinforcement learning approach

An alternative approach is to employ RL, when it is hard to find a control invariant set S for the supervised learning method. We propose a method analogous to PPO, where instead of learning a policy that maximizes reward, we use an actor parametrizing the discriminating hyperplane to minimize instances of safety violation. Our method uses the form $\pi_\theta^a(a|x), \pi_\theta^b(b|x)$ as two normal distributions to sample the hyperplane parameters. During the rollout, based on the sampled (a, b) for each x, with probability $1-\delta$, we sample u from $\{u \in U |\ a^\top u \geq b\}$ which enforces the learned safety constraint, and with probability δ, we sample u from U, allowing an exploration with a small probability. Given the target constraint set X, the reward for learning the hyperplane is designed as

$$r(x, u) = \begin{cases} 1 + d(u) & \text{if } x' \in X, \\ c & \text{otherwise,} \end{cases} \tag{7}$$

where x' is the next state after taking action u, $c \leq 0$ induces a negative reward when safety is violated, as $x' \notin X$, and $d(u) \geq 0$ is an optional bonus term that can be added to the reward if the input u is in the action space U. The bonus term can be necessary when the action space is small, which leads to the actor's predicted hyerplanes being outside of the action space. The reward is positive when safety is satisfied and the bonus term incentivizes the hyperplane to only constrain the input when needed. Let $r^a(\theta) = \frac{\pi_\theta^a(a|x)}{\pi_{\theta_{\text{old}}}^a(a|x)}$ and $r^b(\theta) = \frac{\pi_\theta^b(b|x)}{\pi_{\theta_{\text{old}}}^b(b|x)}$, where θ_{old} is the policy parameter before the update, and define the clipped surrogate objective terms, $L^a(\theta) = \mathbb{E}[\min(r^a(\theta)\hat{A}, \text{clip}(r^a(\theta), 1-\epsilon, 1+\epsilon)\hat{A})], L^b(\theta) = \mathbb{E}[\min(r^b(\theta)\hat{A}, \text{clip}(r^b(\theta), 1-\epsilon, 1+\epsilon)\hat{A})]$, where $\hat{A}$ is the estimated advantage function. We aim to maximize the combined objective $L^a(\theta) + L^b(\theta)$, and perform the policy update as PPO does. By doing so, our policy learns the neural network hyperplane that minimizes long term constraint violation while suppressing the conservativeness of the hyperplane.

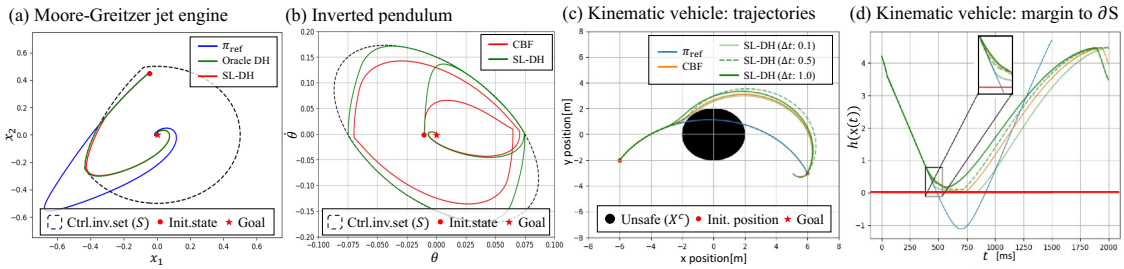

Figure 1: Simulation results of safety filter based on SL-DH, using various invariant set representation. (a) Phase plot of the jet engine system in Xue et al. (2023), under Oracle DH and SL-DH-based safety filter. The safe set is computed by HJ reachability. (b) Phase plot of the inverted pendulum under a bang-bang reference controller, filtered by CBF (Wabersich et al., 2023) and SL-DH. A Lyapunov function level set is chosen as S. (c) Position trajectories of the kinematic vehicle, where S is hand-designed. While the reference controller passes through the unsafe region, safety filters based on CBF and SL-DH are able to keep the vehicle safe. In the supplementary video, we provide an animation of the car trajectory under the SL-DH filter using a lookahead time of 0.5. (d) Margin to the boundary of S over time. $h(x) \geq 0$ represents that the state is inside S. A larger lookahead time allows the SL-DH to engage in safe actions earlier.

5. Experiments

We demonstrate the safety filter in (5) based on the discriminating hyperplane (DH) learned through methods described in Section 4 on various dynamical systems[3]. The proposed methods in the paper are each referred to as **Oracle DH** (DH based on Theorem 2, solved with support vector machine for each state), **SL-DH** (the supervised learning approach in Section 4.1), and **RL-DH** (the RL approach in Section 4.2). Through these experiments, we want to highlight

1. compatibility of DH with various representations of the safe domain S, based on HJ reachability in Xue et al. (2023), Lyapunov function, and manual design,
2. efficacy of the supervised learning approach in Section 4.1 in approximating the Oracle DH,
3. effect of lookahead time Δt on the SL-DH, and comparison against CBF-based safety filter.
4. usage of the SL-DH and RL-DH for safe RL of versatile tasks, and comparison against unconstrained PPO (Schulman et al., 2017), and PPO-Lagrangian (Ray et al., 2019).

1. Moore-Greitzer jet engine (Oracle DH vs. SL-DH, Figure 1(a)): The dynamics of this system is provided in Xue et al. (2023) (n=2, m=1), in which the maximal control invariant set of the target constraint set $X = \{x | \sqrt{x_1^2 + x_2^2} \leq 0.5\}$ is computed by the discounted infinite-horizon HJ reachability formulation. The noticeable feature of this value function is that it is flat and zero everywhere inside the verified invariant set S. Thus, the safety constraint cannot be obtained from the value function, where the DH can be of great use. π_{ref} is designed to stabilize to the origin, which can exit S and violate safety. We use the computed S to train the SL-DH, and compare the closed-loop trajectories under (a) π_{ref}, and safety filters based on (b) the Oracle DH, and (c) the SL-DH. The SL-DH safety filter closely approximates the Oracle DH safety filter.

2. Inverted Pendulum (CBF vs. SL-DH, Figure 1(b)): The reference controller π_{ref} bounces between extreme torques and turns into a stabilizing controller after 12s. It easily exits the target angle region $X = \{x | \ |\theta| < 0.3\}$. A level set of a quadratic Lyapunov function in X is chosen as S. A CBF h is also derived from this Lyapunov function. The dynamics of this system and π_{ref}, S, h are

3. Implementation details: Appendix 7.3, Supplementary video: https://youtu.be/70xxoVW8z8s
Source code: https://github.com/HJReachability/discriminating-hyperplane.git

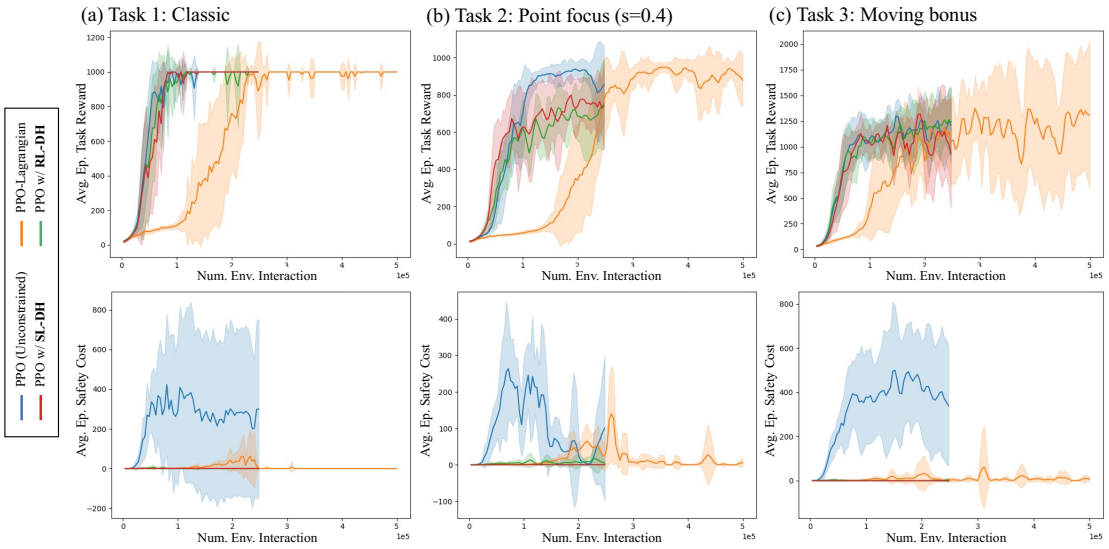

Figure 2: Cart-Pole Experiment: We compared the reward (top) and constraint cost (bottom) across training iterations of PPO, PPO-Lagrangian, and PPO filtered by SL-DH and RL-DH, across various tasks. Notably, SL-DH and RL-DH did not require retraining for each new task. The policies filtered by these methods resulted in minimal safety violations while effectively learning the task, showing performance comparable to that of the unconstrained PPO in all tasks.

detailed in Wabersich et al. (2023). Both CBF and SL-DH filters successfully maintain the system within the safe set while intervening π_{ref} smoothly before the trajectory hits the boundary. This demonstrates how SL-DH is able to achieve similar behavior to CBF without utilization of known system dynamics, assuming access to the safe set beforehand.

3. Kinematic Vehicle (Effect of Δt on SL-DH, Figure 1(c, d)): The state of the vehicle consists of x and y position, heading, and velocity, while the input consists of the yaw rate and acceleration (n=4, m=2). The reference controller is a feedback controller designed to navigate to a goal point. The control invariant set and the CBF is hand-designed to avoid entering an unsafe region centered at the origin. More details are available in Appendix. We train SL-DH with multiple lookahead times: $\Delta t = 0.1, 0.5, 1$. While π_{ref} passes through the unsafe region, safety filters based on SL-DH and CBF are able to prevent the vehicle from entering it. A lower lookahead time leads to a less restrictive but more myopic intervention of the filter, resulting in trajectories approaching closer to the unsafe region. With a larger lookahead time, the safety filter is more preemptive with respect to the boundary. We also note that a higher lookahead time can be beneficial for the learning in that the resulting DH varies more smoothly with respect to state. However, an excessively high value can result in the safety filter to be overly conservative.

4. Cart-Pole (Safe RL for various tasks, Figure 2): We show the utility of SL-DH and RL-DH for learning various tasks, subjected to the identical target constraint. Specifically, we compare PPO, whose policy is filtered by SL-DH and RL-DH during its task training, to unconstrained PPO and PPO-Lagrangian. The CartPole environment is defined in Towers et al. (2023) ($n = 4, m = 1$). The considered target constraint is $X = \{x| \ |s| \leq 0.5\}$, where s is the position of the cart. The classic task is to maximize the length of the trajectory before termination (Task 1). We define two new tasks: stabilizing the cart at $s = 0.4$ (Task 2), and maximizing cart speed while avoiding termination (Task 3). These new tasks challenge to approach the constraint's boundary more than the classic task. Training of the SL-DH and RL-DH is detailed in Appendix. Both SL-DH and RL-DH filtered PPOs demonstrate competitive performance compared to unconstrained PPO. Moreover,

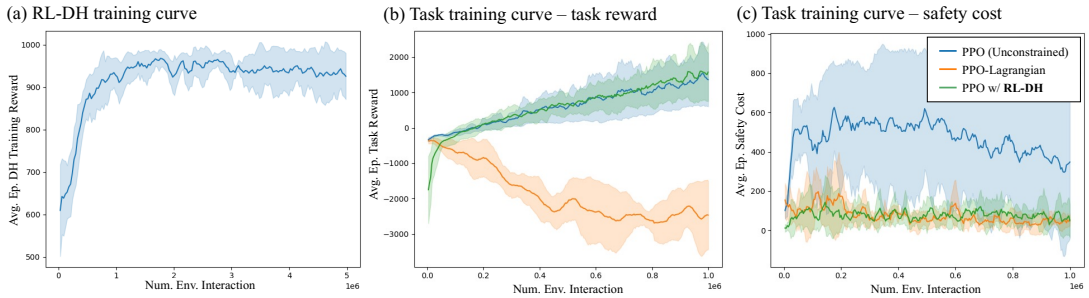

Figure 3: HalfCheetah Experiment: (a) Training curve of RL-DH, where the reward is defined in (7). The value achieving 1000 implies no constraint violation of the trajectory. (b), (c) Reward and constraint cost across iterations of PPO, PPO-Lagrangian, and PPO filtered by RL-DH during the task training.

SL-DH shows no constraint violations, while RL-DH exhibits significantly fewer violations than PPO-Lagrangian across all tasks. RL-DH also achieved zero constraint violations in many instances. We also discovered that our method results in lower constraint violations across a broader range of initial states in the state space, as visualized in Appendix. The training of the PPO-Lagrangian is slower and violates safety more compared to our methods, mainly caused by the simultaneous learning of performance and safety, associated with unstable updates of the Lagrange multiplier.

5. HalfCheetah (Safe RL for uncertain high-dimensional system, Figure 3): We show the viability of RL-DH on a higher dimensional system through HalfCheetah in Towers et al. (2023). In our setup, we utilize a safety constraint of $X = \{x | z \geq -0.3\}$ where z is the height of the robot torso. This constraint represents the requirement for the robot to stay upright during training. The environment consists of $n = 16$ states and $m = 6$ control variables, whose dynamics are hard to model due to the contact with the ground. Training SL-DH for this example is challenging, as it requires a design of a control invariant set for the complicated dynamics. PPO filtered by the trained RL-DH maintains performance competitive with that of unconstrained PPO, while achieving a level of constraint violation comparable to PPO-Lagrangian. However, we observe that RL-DH experiences a distribution shift between the training distribution of the hyperplane and that of the task-specific PPO policy. The simulation results of the trained policies can be found in the supplementary video.

6. Conclusion and Future Work

In this work, we have proposed a novel learning-based approach for the design of safety filters, focusing primarily on the control input constraint, which is key to ensuring safety. Our method, which employs a neural network to produce parameters of a *discriminating hyperplane*, offers a more general viewpoint of how to construct safety filters for general control-affine systems. We have shown that our approach is modular, working together with any control invariant set representation, and also compatible with any performance-driven objectives, thus replacing the need for safe RL approaches that combines performance and safety. While we have shifted the focus from the traditional certificate functions, we recognize that they still play an important role in verifying control invariant sets. Looking ahead, enhancing the robustness of the RL-DH safety filter against potential distribution shifts will be crucial for the improvement of our RL-based approach. Furthermore, we are keen on extending of our method for uncertain systems, as discussed in Lopez et al. (2020); Brunke et al. (2022b); Cohen et al. (2023), which may include extensions to non-affine forms of the control input constraints like second-order cone constraints, as in Castañeda et al. (2021); Dhiman et al. (2021); Taylor et al. (2021), and to multiple constraints, as in Kim et al. (2023a).

Acknowledgments

This work is supported by the National Science Foundation Grant CMMI-1944722, the DARPA Assured Autonomy and Assured Neuro Symbolic Learning and Reasoning (ANSR) programs, and the NASA ULI on Safe Aviation Autonomy. The work of Jason J. Choi received the support of a fellowship from Kwanjeong Educational Foundation, Korea. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any aforementioned organizations.

7. Appendix

7.1. Proof of Theorem 1

We prove Theorem 1 by proving Lemma 2, since from Definitions 1 and 3, Lipschitz continuity of π_{ref} implies that $a(x)^\top u \geq b(x)$ is a valid safety constraint, and thus, $\{x|a(x)^\top u = b(x)\}$ being a discriminating hyperplane. Under the conditions in Theorem 1, (5) can be written as a QP problem which satisfies the conditions in Morris et al. (2013, Thm.1), which states that its solution is unique and Lipschitz w.r.t. x. □

7.2. Proof of Theorem 2

For the proof of the theorem, we consider the distance-like function, h, which is continuously differentiable and whose derivative is bounded and Lipschitz continuous, such that $S = \{x \mid h(x) \geq 0\}$ and $\frac{\partial h}{\partial x}(x)$ is bounded away from 0 for all $x \in \partial S$. As in Lemma 1, h is solely used as a means to provide proof for the statement. Using h, we have $\{u \in U \mid I_S(\mathrm{x}(\Delta t)) = 1\} = \{u \in U \mid h(\mathrm{x}(\Delta t)) \geq 0\}$. From the boundedness of f, g and U from Assumption 1, we have

$$||\mathrm{x}(\Delta t) - (x + \Delta t(f(x) + g(x)u))\,|| \leq M_1 \Delta t^2$$

for all $u \in U$, and for some constant $M_1 > 0$. From the Lipschitz continuity of $\frac{\partial h}{\partial x}$, we have

$$\frac{\partial h}{\partial x} \cdot (y - x) + \frac{L}{2}||y - x||^2 \geq h(y) - h(x) \geq \frac{\partial h}{\partial x} \cdot (y - x) - \frac{L}{2}||y - x||^2, \tag{8}$$

where L is the Lipschitz continuity of $\frac{\partial h}{\partial x}$. Thus, we have

$$M_3 \Delta t^2 + M_4 \Delta t^4 \geq h(\mathrm{x}(\Delta t)) - h\left(x + \Delta t(f(x) + g(x)u)\right) \geq -M_3 \Delta t^2 - M_4 \Delta t^4$$

where $M_3 = HM_1$, $M_4 = \frac{1}{2}LM_1^2$, and H is the bound of $||\frac{\partial h}{\partial x}||$. Also, with $y = x + \Delta t(f(x) + g(x)u)$ in (8), we have

$$\frac{\partial h}{\partial x} \cdot (f(x) + g(x)u)\Delta t + M_5 \Delta t^2 \geq h\left(x + \Delta t(f(x) + g(x)u)\right) - h(x) \geq \frac{\partial h}{\partial x} \cdot (f(x) + g(x)u)\Delta t - M_5 \Delta t^2$$

for some constant $M_5 > 0$ for all $u \in U$, due to the boundedness of f, g, and U. Combining the two above equations, we get for all $u \in U$,

$$\begin{aligned} &\frac{\partial h}{\partial x} \cdot (f(x) + g(x)u)\Delta t + (M_3 + M_5)\Delta t^2 + M_4 \Delta t^4 \\ &\qquad \geq h(\mathrm{x}(\Delta t)) - h(x) \geq \frac{\partial h}{\partial x} \cdot (f(x) + g(x)u)\Delta t - (M_3 + M_5)\Delta t^2 - M_4 \Delta t^4. \end{aligned} \tag{9}$$

From the right hand side of (9), consider $a(x)^\top = \frac{\partial h}{\partial x} \cdot g(x)\Delta t$, and $b(x) = -h(x) - \frac{\partial h}{\partial x} \cdot f(x)\Delta t + (M_3 + M_5)\Delta t^2 + M_4\Delta t^4$, which are both Lipschitz continuous in x, due to the Lipschitz continuity of h, $\frac{\partial h}{\partial x}$, f, and g. Then $h(\mathrm{x}(\Delta t)) \geq a(x)^\top u - b(x)$, for all $u \in U$. Thus, if $a(x)^\top u \geq b(x)$, then $h(\mathrm{x}(\Delta t)) \geq 0$. Therefore, $\Pi(x) = \{u \in U \mid a(x)^\top u \geq b(x)\} \subseteq \{u \in U \mid I_S(\mathrm{x}(\Delta t)) = 1\}$.

Next, from (9), we get

$$a(x)^\top u - b(x) + 2\left((M_3 + M_5)\Delta t^2 + M_4\Delta t^4\right) \geq h(\mathrm{x}(\Delta t)) \geq a(x)^\top u - b(x).$$

If $\{u \in U \mid I_S(\mathrm{x}(\Delta t)) = 1\}$ has a non-empty interior, $\exists u \in U$ such that $h(\mathrm{x}(\Delta t)) > 0$, thus, $a(x)^\top u - b(x) + 2\left((M_3 + M_5)\Delta t^2 + M_4\Delta t^4\right) > 0$. For small enough Δt, $a(x)^\top u - b(x) \geq 0$, thus, proving the second statement. □

7.3. Implementation Details

SL-DH: In the loss function, we typically set $\gamma_{\text{pos}} > \gamma_{\text{neg}}$, since in order for the learned hyperplane to be a valid discriminating hyperplane, it is imperative to not misclassify unsafe inputs. Next, due to prediction error of the neural-network output, the direct use of raw predictions for the safety constraint $a_\theta(x)^\top u \geq b_\theta(x)$ might not ensure safety. To improve safety, we employ a minor adjustment factor, $\epsilon \geq 0$, to tighten the predicted hyperplane constraint. Specifically, we use the revised constraint, $a_\theta(x)^\top u \geq b_\theta(x) + \epsilon$. This adjustment can be viewed as a straightforward calibration step, a practice commonly seen in many deep neural network applications Guo et al. (2017). It should be noted, however, that over-utilizing this calibration (i.e., setting an excessively high value for ϵ) can lead to overly conservative behavior by the safety filter. Nevertheless, in our experimental results, we observe no such over-conservatism, indicating minimal, yet effective calibration.

Inverted Pendulum: The dataset for training SL-DH consists of 10k states sampled uniformly across S, with 300 inputs sampled uniformly across U per state. The training data is resampled every epoch, where each epoch we take 5 gradient steps. For the labels, we use a lookahead time of $\Delta t = 0.05$. We use the following network parameters: 5 hidden layers, 1000 hidden layer size, ReLU activations, learning rate of 5e-4, $\gamma_{\text{pos}} = 10$, $\gamma_{\text{neg}} = 1$. We train for 400 epochs with 5 gradient steps per epoch. We use $\epsilon = 0.01$ for the calibration.

Kinematic Vehicle: The system state is defined as $[p_x, p_y, \theta, v]$ which describes the positions, heading with respect to the x-axis, and the velocity of the car. The input is defined as $[\omega, a]$ where ω is the yaw rate and a is the acceleration of the vehicle. The target constraint set is $X = \{x | \sqrt{p_x^2 + p_y^2} \geq r\}$, where $r = 2$. We use input limits $|\omega| \leq 2$, $|a| \leq a_{\max} = 1$. We use the control invariant set $S := \{x | h(x) \geq 0\}$ where h is the CBF designed as:

$$h(x) = \sqrt{\left(p_x + \frac{v^2}{4a_{\max}}\cos\theta\right)^2 + \left(p_y + \frac{v^2}{4a_{\max}}\sin\theta\right)^2} - \left(r + \frac{v^2}{4a_{\max}}\right) \tag{10}$$

The CBF is designed by adding a safety margin to X^c, based on the vehicle's current heading angle and the stopping distance, calculated based on the maximum deceleration $a_{\max}$.

The dataset for training SL-DH consists of 8k states and 500 inputs per epoch. The system dynamics are discretized under $\Delta t_{\text{sys}} = 0.05$ and we compare various lookahead times of the values $\Delta t = 0.1, 0.5, 1$. To clearly examine the effect of lookahead time, we keep all training parameters and network parameters unchanged across different lookahead times. We use the following network

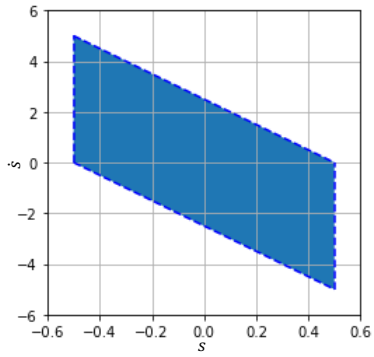

Figure 4: Hand-designed control invariant set S for the Cart-Pole experiment, used to train SL-DH. The projection of the set to $(s, \dot{s})$ is visualized.

Figure 5: Training curve of RL-DH training for the Cart-Pole Experiment. The reward reaching 2000 indicates that constraint was never violated.

parameters: 3 hidden layers, 2000 hidden layer size, ReLU activations, learning rate of 1e-4, $\gamma_{\text{pos}} = 5$, $\gamma_{\text{neg}} = 1$. We train for 400 epochs with 5 gradient steps per epoch. We use $\epsilon = 0.3$ for the calibration for all lookahead times. Since the heading angle θ is contained with the values $[0, 2\pi)$, we use $\sin$ and $\cos$ encoding of θ to preserve this continuity. Specifically, we map the state to the vector $[p_x, p_y, \sin\theta, \cos\theta, v]$ during the forward pass of the hyperplane network.

Cart-Pole: *(SL-DH training)* For SL-DH, we utilize the geometry of the system to design a control invariant set, visualized in Figure 4. We verify this design through a Monte Carlo simulation of trajectories in which we densely sample initial states near the boundary of S and verify based on random trajectories that each state satisfies the control invariance condition. To train SL-DH, we use 10k steps per epoch with 300 sampled inputs both uniformly sampled over S. We train over 200 epochs with 5 gradient steps per epoch. For the labels, we use a lookahead time of Δt_{sys} (1 steps in the system environment). For the network parameters, we use 5 hidden layers, 1000 hidden width, ReLU activations, learning rate of 5e-4, $\gamma_{\text{pos}} = 5$, $\gamma_{\text{neg}} = 1$.

(RL-DH training) As RL-DH does not rely on a pre-defined safe set, we pretain RL-DH as discussed in Section 4.2, where the environment no longer terminates upon the pole falling. The training curve is reported in Figure 5. In detail, our environment uses 1000 steps without termination whereas the classic Cart-Pole environment terminates when the pole falls over, allowing for better exploration in regards to the safety constraints. As detailed in Section 4.2, we use a bonus term $d(u) = 1$ when u is inside the action space and $c = -1$ when $x \notin X$. The training details for

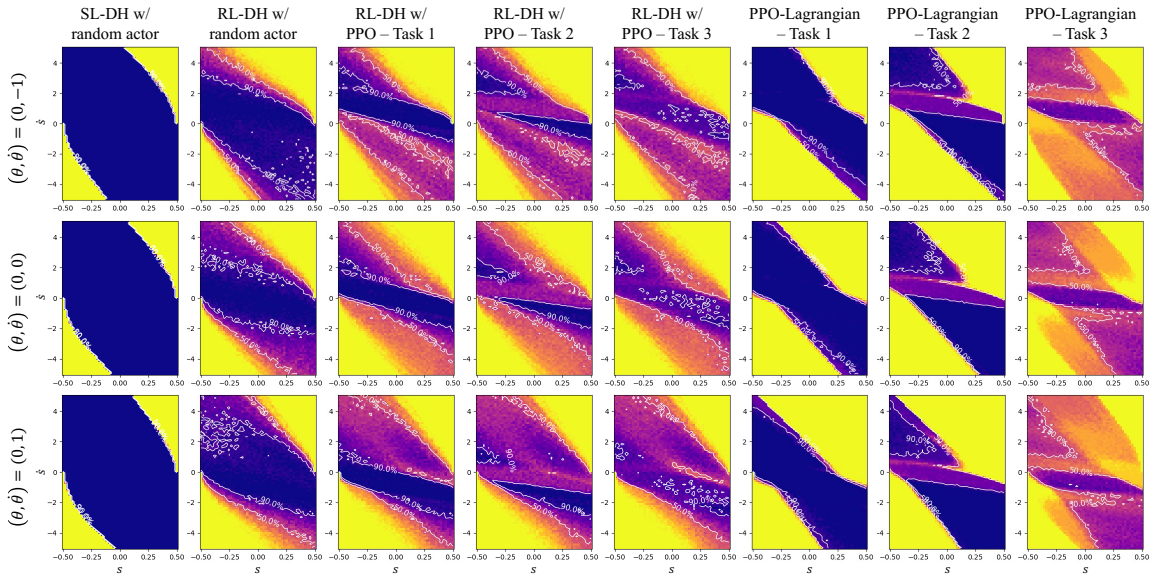

Figure 6: Constraint satisfaction rates in random rollouts of Cart-Pole trajectories: evaluation at initial states in 2D slices of the state space, at $(\theta, \dot{\theta})$=(0, -1), (0, 0), (0, 1).

RL-DH are as follows: 4000 steps per epoch, 1250 epochs, $\gamma = 0.99$, clip ratio of 0.2, actor and critic sizes of 2 hidden layers of 256 width, actor learning rate of 3e-4, critic learning rate of 1e-3, GAE λ of 0.97, 80 gradient steps per actor and critic update steps.

When utilizing the trained RL-DH for filtering PPO actors for task training, we use a truncated normal distribution whose support region is determined by the discriminating hyperplane to enforce hard safety constraint on the control input. Since $m = 1$ for Cart-Pole, we are able to determine the left and right bounds of the truncated normal distribution from the RL-DH hyperplane. Specifically, given the hyperplane parameters a, b, the left and right bounds are $[l, r] = [\min(\frac{b}{a}, 1), 1]$ when $a > 0$, and $[l, r] = [-1, \max(\frac{b}{a}, -1)]$ when $a < 0$. We then sample the control input from the truncated normal, $u \sim \mathcal{TN}(\text{clip}(\mu_\theta, l, r), \sigma_\theta, l, r)$, where $(\mu_\theta, \sigma_\theta)$ is the mean and variance determined by the actor network, to sample only safe control during rollouts.

(Safe RL experiments) The training of PPO, PPO-Lagrangian, PPO with SL-DH, and PPO with RL-DH all use the same parameters as detailed for training the RL-DH, except for the number of epochs. We train each method over 10 different random seeds. We train for 62.5 epochs for PPO, PPO with SL-DH, and PPO with RL-DH, while for PPO-Lagrangian we train for 125 epochs. We do this to save computation time since non-Lagrangian methods converge to a high-performing policy more promptly than PPO-Lagrangian. For PPO-Lagrangian, we use a learning rate of 5e-2 for the lagrangian multiplier with one gradient step per epoch. The cost function used for PPO-Lagrangian is defined as $c(x) = 1$ if $|s| < 0.5$, else $c(x) = 0$.

We evaluate all four methods for three different variations of the CartPole task. The first is the classic CartPole objective where the reward is 1 when the pole angle θ is within ± 12 degrees. The second task consists of a reward of $r(x) = 1 + |s - 0.4|$. which incentivizes the cart position to be at $s = 0.4$. This task is more challenging for the policy to achieve while satisfying the safety constraint since the desired position is close to the safety boundary. The final task uses a reward of

$r(s) = 1 + |\dot{s}|$. This incentivizes the cart to continue moving as much as possible while keeping the pole upright, making it even harder to satisfy the safety constraint.

Finally, we compare the set of safe initial states across four methods. In addition to the PPO policies filtered by RL-DH safety filters and the PPO-Lagrangian policies for each task, we also include an actor policy initialized as a random policy (with a randomly weighted neural network) that is filtered with SL-DH and RL-DH. This approach effectively isolates the safety-filtered control input from any specific performance objective. For each initial state, selected from grid points within 2D grids in the $(s, \dot{s})$ space, sliced at $(\theta, \dot{\theta}) = (0, -1), (0, 0), (0, 1)$, we rollout 50 trajectories (5 trajectories per 10 random seeds) and evaluate whether they exit the set X.

We present the success rate of constraint satisfaction for these 50 trajectories in Figure 6. Two observations in the figure are particularly noteworthy: First, the SL-DH safety filter provides the largest set of safe initial states among the four methods, with a clear demarcation between safe and unsafe initial states. Second, while the support region of the distribution of safe initial states for PPO-Lagrangian varies with the tasks, the distribution for RL-DH exhibits a similar support region across tasks. This suggests that while PPO-Lagrangian learns task-specific safe policies, the RL-DH-based safety filter is task-agnostic.

HalfCheetah: The safety constraint we impose is $X = \{x | z \geq -0.3\}$ where z is the height of the robot torso. To train RL-DH, we use a reward of 1 if $x \in X$ and $c = -2$ otherwise. The parameters used to train RL-DH are as follows: 2 hidden layers and 256 hidden width (for both policy and value function networks), discount factor $\gamma = 0.99$, 30k steps per epoch over 167 epochs, policy learning rate of 3e-4, value learning rate of 1e-3, GAE λ of 0.97, and 80 gradient steps for both the policy and value function. When using RL-DH as a filter for PPO, we take a different approach than CartPole since it is not possible to come up with a truncated normal distribution form for a higher dimensional control input space. Instead of using a truncated normal distribution, we use a normal distribution but project μ_θ onto the hyperplane if the constraint $a\mu_\theta \geq b$ is not satisfied. Even though this doesn't provide hard safety guarantees for the sampled actions, we found this to provide the best performance while still mitigating safety significantly.

The training for PPO, PPO with RL-DH, and PPO-Lagrangian use the following parameters: 2 hidden layers and 256 hidden width (for both policy and value function networks), discount factor $\gamma = 0.99$, 4k steps per epoch over 250 epochs, policy learning rate of 3e-4, value learning rate of 1e-3, GAE λ of 0.97, and 80 gradient steps for both the policy and value function. The reported results are from 5 random seeds for each methods.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 06–11 Aug 2017.

Eitan Altman. Constrained markov decision processes with total cost criteria: Lagrangian approach and dual linear program. *Mathematical Methods of Operations Research*, 1998.

Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Trans. on Automatic Control*, 62(8):3861–3876, 2016.

Jean-Pierre Aubin, Alexandre M Bayen, and Patrick Saint-Pierre. *Viability theory: new directions*. Springer Science & Business Media, 2011.

Somil Bansal and Claire J Tomlin. Deepreach: A deep learning approach to high-dimensional reachability. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1817–1824, Xi'an, China, 2021.

Dimitri Bertsekas. Infinite time reachability of state-space regions by using feedback control. *IEEE Trans. on Automatic Control*, 17(5):604–613, 1972.

Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022a.

Lukas Brunke, Siqi Zhou, and Angela P Schoellig. Barrier bayesian linear regression: Online learning of control barrier conditions for safety-critical control of uncertain systems. In *Learning for Dynamics and Control Conference*, pages 881–892. PMLR, 2022b.

Michael Enqi Cao, Matthieu Bloch, and Samuel Coogan. Efficient learning of hyperrectangular invariant sets using gaussian processes. *IEEE Open Journal of Control Systems*, 1:223–236, 2022.

Fernando Castañeda, Haruki Nishimura, Rowan McAllister, Koushil Sreenath, and Adrien Gaidon. In-distribution barrier functions: Self-supervised policy filters that avoid out-of-distribution states. *arXiv preprint arXiv:2301.12012*, 2023.

Fernando Castañeda, Jason J. Choi, Bike Zhang, Claire J. Tomlin, and Koushil Sreenath. Pointwise feasibility of gaussian process-based safety-critical control under model uncertainty. In *IEEE Conference on Decision and Control*, pages 6762–6769, 2021.

Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3387–3395, 2019.

Jason Choi, Fernando Castañeda, Claire Tomlin, and Koushil Sreenath. Reinforcement Learning for Safety-Critical Control under Model Uncertainty, using Control Lyapunov Functions and Control Barrier Functions. In *Robotics: Science and Systems*, Corvalis, OR, 2020.

Jason J Choi, Donggun Lee, Boyang Li, Jonathan P How, Koushil Sreenath, Sylvia L Herbert, and Claire J Tomlin. A forward reachability perspective on robust control invariance and discount factors in reachability analysis. *arXiv preprint arXiv:2310.17180*, 2023.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Francis H Clarke, Yuri S Ledyaev, Ronald J Stern, and Peter R Wolenski. *Nonsmooth analysis and control theory*, volume 178. Springer Science & Business Media, 2008.

Max H Cohen, Makai Mann, Kevin Leahy, and Calin Belta. Uncertainty quantification for recursive estimation in adaptive safety-critical control. *arXiv preprint arXiv:2304.01901*, 2023.

Charles Dawson, Sicun Gao, and Chuchu Fan. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, pages 1–19, 2023.

Vikas Dhiman, Mohammad Javad Khojasteh, Massimo Franceschetti, and Nikolay Atanasov. Control barriers in bayesian learning of system dynamics. *IEEE Transactions on Automatic Control*, 2021.

Yousef Emam, Gennaro Notomista, Paul Glotfelter, Zsolt Kira, and Magnus Egerstedt. Safe reinforcement learning using robust control barrier functions. *IEEE Robotics and Automation Letters*, pages 1–8, 2022. doi: 10.1109/LRA.2022.3216996.

J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Trans. on Automatic Control*, 64(7):2737–2752, 2019a.

Jaime F Fisac, Neil F Lugovoy, Vicenç Rubies-Royo, Shromona Ghosh, and Claire J Tomlin. Bridging hamilton-jacobi safety analysis and reinforcement learning. In *Int. Conf. on Robotics and Automation (ICRA)*, pages 8550–8556, Montreal, QC, Canada, 2019b.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

Kai-Chieh Hsu, Duy Phuong Nguyen, and Jaime Fernàndez Fisac. Isaacs: Iterative soft adversarial actor-critic for safety. In *Learning for Dynamics and Control Conference*, pages 90–103. PMLR, 2023.

Subin Huh and Insoon Yang. Safe reinforcement learning for probabilistic reachability and safety specifications: A lyapunov-based approach. *arXiv preprint arXiv:2002.10126*, 2020.

Dohyeong Kim, Kyungjae Lee, and Songhwai Oh. Trust region-based safe distributional reinforcement learning for multiple constraints. In *Advances in neural information processing systems*, 2023a.

Yunho Kim, Hyunsik Oh, Jeonghyun Lee, Jinhyeok Choi, Gwanghyeon Ji, Moonkyu Jung, Donghoon Youm, and Jemin Hwangbo. Not only rewards but also constraints: Applications on legged robot locomotion. *arXiv preprint arXiv:2308.12517*, 2023b.

Gary Lieberman. Regularized distance and its applications. *Pacific journal of Mathematics*, 117(2): 329–352, 1985.

Lars Lindemann, Alexander Robey, Lejun Jiang, Stephen Tu, and Nikolai Matni. Learning robust output control barrier functions from safe expert demonstrations. *arXiv preprint arXiv:2111.09971*, 2021.

Changliu Liu and Masayoshi Tomizuka. Control in a safe set: Addressing safety in human-robot interactions. In *Dynamic Systems and Control Conference*, volume 46209. American Society of Mechanical Engineers, 2014.

Simin Liu, Changliu Liu, and John Dolan. Safe control under input limits with neural control barrier functions. In *Conference on Robot Learning*, pages 1970–1980. PMLR, 2023.

Brett T Lopez, Jean-Jacques E Slotine, and Jonathan P How. Robust adaptive control barrier functions: An adaptive and data-driven approach to safety. *IEEE Control Systems Letters*, 5(3): 1031–1036, 2020.

Benjamin Morris, Matthew J Powell, and Aaron D Ames. Sufficient conditions for the lipschitz continuity of qp-based multi-objective control of humanoid robots. In *52nd IEEE Conference on Decision and Control*, pages 2920–2926. IEEE, 2013.

M. Nagumo. Über die lage der integralkurven gewöhnlicher differentialgleichungen. *Proc. of the Physico-Mathematical Society of Japan. 3rd Series*, 24:551–559, 1942.

Stephen Prajna. Barrier certificates for nonlinear model validation. *Automatica*, 2006.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7(1):2, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Oswin So, Zachary Serlin, Makai Mann, Jake Gonzales, Kwesi Rutledge, Nicholas Roy, and Chuchu Fan. How to train your neural control barrier function: Learning safety filters for complex input-constrained systems. *arXiv preprint arXiv:2310.15478*, 2023.

Krishnan Srinivasan, Benjamin Eysenbach, Sehoon Ha, Jie Tan, and Chelsea Finn. Learning to be safe: Deep rl with a safety critic. *arXiv preprint arXiv:2010.14603*, 2020.

Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for safety-critical control with control barrier functions. In *Learning for Dynamics and Control*, pages 708–717, 2020.

Andrew J. Taylor, Victor D. Dorobantu, Sarah Dean, Benjamin Recht, Yisong Yue, and Aaron D. Ames. Towards robust data-driven control synthesis for nonlinear systems with actuation uncertainty. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6469–6476, 2021.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6 (3):4915–4922, 2021.

Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium, March 2023. URL https://zenodo.org/record/8127025.

Kim P Wabersich and Melanie N Zeilinger. Scalable synthesis of safety certificates from data with application to learning-based control. In *2018 European Control Conference (ECC)*, pages 1691–1697. IEEE, 2018.

Kim P. Wabersich, Andrew J. Taylor, Jason J. Choi, Koushil Sreenath, Claire J. Tomlin, Aaron D. Ames, and Melanie N. Zeilinger. Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 2023.

Kim Peter Wabersich and Melanie N Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 129:109597, 2021.

Nolan C Wagener, Byron Boots, and Ching-An Cheng. Safe reinforcement learning using advantage-based intervention. In *International Conference on Machine Learning*, pages 10630–10640. PMLR, 2021.

Tianhao Wei and Changliu Liu. Safe control algorithms using energy functions: A unified framework, benchmark, and new directions. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 238–243, 2019. doi: 10.1109/CDC40024.2019.9029720.

Bai Xue, Naijun Zhan, Martin Fränzle, Ji Wang, and Wanwei Liu. Reach-avoid verification based on convex optimization. *IEEE Transactions on Automatic Control*, 2023.