Back to Base: Towards Hands-Off Learning via Safe Resets with Reach-Avoid Safety Filters

Sander Tonkens[†] **Nikhil Uday Shinde**[†] **Azra Begzadić**[†] **Dylan Hirsch**
Kaleb Ugalde **Michael C. Yip** **Jorge Cortés** **Sylvia L. Herbert**

University of California San Diego [†] Equal contribution.

Abstract: Designing controllers that accomplish tasks while guaranteeing safety constraints remains a significant challenge. We often want an agent to perform well in a nominal task, such as environment exploration, while ensuring it can avoid unsafe states and return to a desired target by a specific time. In particular we are motivated by the setting of safe, efficient, hands-off training for reinforcement learning in the real world. By enabling a robot to safely and autonomously reset to a desired region (e.g., charging stations) without human intervention, we can enhance efficiency and facilitate training. Safety filters, such as those based on control barrier functions, decouple safety from nominal control objectives and rigorously guarantee safety. Despite their success, constructing these functions for general nonlinear systems with control constraints and system uncertainties remains an open problem. This paper introduces a safety filter obtained from the value function associated with the reach-avoid problem. The proposed safety filter minimally modifies the nominal controller while avoiding unsafe regions and guiding the system back to the desired target set. By preserving policy performance while allowing safe resetting, we enable efficient hands-off reinforcement learning and advance the feasibility of safe training for real world robots. We demonstrate our approach using a modified version of soft actor-critic to safely train a swing-up task on a modified cartpole stabilization problem.

Keywords: Safety filters, Reachability analysis, Safe learning

1 Introduction

Reach-avoid problems have typically focused on reaching a target set in the state space as quickly as possible while avoiding a failure set. One popular value function-based approach is Hamilton-Jacobi reachability (HJR) analysis [1]. HJR encodes a differential game between the control and disturbance to the system. The result is a value function whose level sets define the reach-avoid set (or tube): the set of states from which the target can be reached safely despite worst-case disturbance at the end of (or within) the time horizon. Additionally, the gradients of the value function provide the optimal control strategy for reaching the goal *in minimum time*.

A secondary approach is through the separate use of control Lyapunov functions (CLFs) that enforce stabilizing to a desired goal, combined with control barrier functions (CBFs) that enforce safety [2]. An optimization problem (which is not necessarily feasible) incorporating both constraints is solved at each iteration, forcing the control to maintain safety (via the CBF) while progressing toward the goal (via the CLF). However, the CLF constraint only ensures *exponential stabilizability* to the goal.

In practical applications, many scenarios require safely reaching a target at or within an exact time in the future, while achieving a secondary objective such as minimizing control input. One such application is hybrid systems, where the system must safely reach a switching state within a given time [3]. Additionally, a broad range of systems require being able to return to a desired goal within a certain time, e.g., a drone has to safely return to a dock station before its battery runs out. In this

Workshop on Safe and Robust Robot Learning for Operation in the Real World at CoRL 2025.

paper, we motivate our work through the application of safely training a reinforcement learning (RL) agent in the real world, either from scratch or through fine-tuning. Real world RL training entails significant risks to both the robot and its environment if safety is not properly ensured. Furthermore, the trajectory of the robot can lead to states where it cannot continue operating due to system limitations, such as running out of battery or getting stuck [4]. This often necessitates human intervention to reset the robot to a desired state from which the robot can continue learning. This can make real world training prohibitively expensive and even infeasible in certain scenarios, like search and rescue or space exploration. For example, a drone fine-tuning its flight or search policy during deployment in an unknown environment could benefit from an approach that minimally modifies the nominal policy to allow for effective learning, while ensuring that it can always safely return to a desired target set. This would allow the drone to seamlessly resume its tasks and continue learning, enhancing the efficiency of real world RL.

We propose characterizing this problem through a time-varying reach-avoid value function. However, instead of directly applying the optimal control to safely reach the goal in minimum time, as is typically done using HJR, we view the reach-avoid tube as a time-varying safe set and use an associated CBF-like constraint on the value function to design a safety filter. This safety filter allows a user to prioritize safety while also pursuing a secondary objective (e.g. learning a performance policy using RL). Our contributions are as follows:

1. We introduce the notion of a viscosity-based control barrier function (VB-CBF), which resembles a standard CBF but has less restrictive assumptions. For control-and disturbance affine dynamics its associated time-varying safety filter is a quadratic program.
2. We prove that the HJR reach-avoid value function is a VB-CBF for the reach-avoid set, and admits a feasible control set for almost every state and time pair in the reach-avoid set. Under mild assumptions, it also serves as a VB-CBF for the reach-avoid tube.
3. We demonstrate the effectiveness of our approach for safely training an RL agent on an enhanced cartpole environment. We successfully showcase how we can use our method to train an RL agent without the need to explicitly reset the environment. Specifically, the reach-avoid VB-CBF enables both training safely and returning to a safe initial state at the end of each episode. Both are crucial components for scalable real world RL.

2 Related Work

Hamilton-Jacobi Reachability. Many safety-critical autonomous tasks can be formulated as reach-avoid problems [5, 6]. HJR offers a rigorous framework for verifying safety and reachability in dynamical systems [1]. To tackle the scalability challenges of HJR, learning-based approaches have been developed [7, 8, 9, 10], but they require specifying the desired task objectives as a part of their reach-avoid formulation *a priori*, and are hence limited to a specific subset of tasks. In contrast, [11] uses HJR to define a feasible set and constrain an RL agent to learn an unspecified objective constrained to this set. This method considers constraints only and does not enforce reaching a target.

Control barrier functions. Safety is often enforced using CBFs [2]. In particular, CBFs are used as safety filters by adjusting a nominal control law to ensure the system satisfies safety constraints [12], resulting in a quadratic program. However, constructing CBFs and ensuring feasibility remain challenging [13]. To tackle feasibility challenges, backup CBFs have been proposed to guarantee feasibility using a predefined backup policy to a predefined safe set [14, 15]. While more broadly applicable to complex systems, specifying a backup set a priori limits the implicitly specified safety region and the backup CBF formulation introduces additional computational complexity. Deriving a CBF from HJR without a target (i.e., only avoiding unsafe states) is explored in [16] via a novel value function. Similarly, in [17], the authors provide a method for constructing CBFs using ideas from reachability analysis, but their approach is limited to fixed policies.

Safe & recoverable RL. Safe RL is well-explored [18] and and typically framed either as a constrained Markov decision process [19] or using control-theoretic methods to restrict the action space of the agent. These works leverage that separating task performance from safety objectives can improve both performance and safety [20]. Lyapunov-based methods, including CBFs, have been

used to guarantee safety while learning for model-based [21, 22] and model-free RL [23]. This has been extended to include system uncertainties and disturbances [24, 25]. However, these methods typically do not consider resetting to a desired state at the end of each episode, which is desired for autonomous, i.e., without human presence, training. In contrast, reach-avoid methods focus exclusively on reaching a desired goal state, which is often difficult to define in advance and limits the use of general-purpose RL algorithms. To address these challenges, we propose a time-varying CBF that encodes both safety (for all times) and recovery to a desired target set within the specified time.

3 Preliminaries

Consider a system of the form

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

where $x \in \mathbb{R}^n$ is the state, $u \in \mathcal{U} \subseteq \mathbb{R}^p$ is the control input, $d \in \mathcal{D} \subseteq \mathbb{R}^q$ is the disturbance, with convex and compact sets $\mathcal{U}$ and $\mathcal{D}$. For each initial time $t \leq 0$, we denote the sets of admissible control and disturbance signals by $\mathbb{U}(t) := \{\boldsymbol{u} : [t, 0] \to \mathcal{U} \mid \boldsymbol{u} \text{ is measurable}\}$ and $\mathbb{D}(t) := \{\boldsymbol{d} : [t, 0] \to \mathcal{D} \mid \boldsymbol{d} \text{ is measurable}\}$, respectively. Throughout this work, we make the following assumption on the dynamics.

Assumption 1. The function $f : \mathbb{R}^n \times \mathcal{U} \times \mathcal{D} \to \mathbb{R}^n$ is bounded and globally Lipschitz.

It follows from Assumption 1 that for each $t \leq 0$, $x \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{U}(t)$, and $\boldsymbol{d} \in \mathbb{D}(t)$, there exists a unique (Carathéodory) solution $\boldsymbol{x} : [t, 0] \to \mathbb{R}^n$ of (1) which satisfies $\boldsymbol{x}(t) = x$ [26]. We denote this solution by $\xi_{x,t}^{\boldsymbol{u},\boldsymbol{d}}$.

3.1 Hamilton-Jacobi Reachability

HJR determines the set of initial states from which a system can robustly reach a goal while avoiding failure states. This analysis is formulated in terms of a differential game played over the dynamics (1), where we consider the control u and disturbance d as the players of the game [1]. Player u wishes to ensure the system enters the target set $\mathcal{T} \subseteq \mathbb{R}^n$ by some final time (which we shall choose to be 0), while avoiding the failure set $\mathcal{F} \subseteq R^n$ in the process, and player d wishes for the opposite. Given an initial time $t \leq 0$, the reach-avoid tube $\mathcal{RA}(\mathcal{T}, \mathcal{F}, t)$ represents the set of initial states x for which the control can win the game. More precisely, we have

$$\mathcal{RA}(\mathcal{T},\mathcal{F},t) := \{x \in \mathbb{R}^n \mid \forall \lambda \in \Lambda(t), \exists \mathbf{u} \in \mathbb{U}(t), \exists s \in [t,0], \xi_{x,t}^{\boldsymbol{u},\lambda[\boldsymbol{u}]}(s) \in \mathcal{T} \wedge \forall \tau \in [t,s], \xi_{x,t}^{\boldsymbol{u},\lambda[\boldsymbol{u}]}(\tau) \notin \mathcal{F}\}, \tag{2}$$

where $\Lambda(t)$ represents the set of non-anticipative (i.e., causal) strategies from which we permit the disturbance player to choose. Formally, we have

$$\begin{aligned}\Lambda(t) = \{\lambda : \mathbb{U}(t) \to \mathbb{D}(t) \mid &\forall s \in [t, 0], \forall \boldsymbol{u}, \hat{\boldsymbol{u}} \in \mathbb{U}(t), \\ &\boldsymbol{u}(\tau) = \hat{\boldsymbol{u}}(\tau) \text{ a.e. } \tau \in [t, s] \implies \lambda[\boldsymbol{u}](\tau) = \lambda[\hat{\boldsymbol{u}}](\tau) \text{ a.e. } \tau \in [t, s]\}.\end{aligned} \tag{3}$$

Though we are mainly interested in the tube, it will be helpful to consider the reach avoid set

$$\mathcal{RA}_0(\mathcal{T},\mathcal{F},t) := \{x \in \mathbb{R}^n \mid \forall \lambda \in \Lambda(t), \exists \mathbf{u} \in \mathbb{U}(t), \xi_{x,t}^{\boldsymbol{u},\lambda[\boldsymbol{u}]}(0) \in \mathcal{T} \wedge \forall \tau \in [t,0], \xi_{x,t}^{\boldsymbol{u},\lambda[\boldsymbol{u}]}(\tau) \notin \mathcal{F}\}, \tag{4}$$

which corresponds to a game in which the controller wishes to ensure the state is in the target set *at* the final time rather than *by* the final time. By definition, $\mathcal{RA}_0(\mathcal{T}, \mathcal{F}, t) \subseteq \mathcal{RA}(\mathcal{T}, \mathcal{F}, t)$. To compute $\mathcal{RA}_0(\mathcal{T}, \mathcal{F}, t)$, one first computes the value function $V_0 : \mathbb{R}^n \times (-\infty, 0] \to \mathbb{R}$ associated with the game. We define the target and constraint functions $\ell(x), g(x) : \mathbb{R}^n \to \mathbb{R}$ such that $\ell(x) \geq 0 \iff x \in \mathcal{T}$ and $g(x) < 0 \iff x \in \mathcal{F}$. Then, V_0 is the unique viscosity solution (for details on viscosity solutions, we refer the reader to [27]) of the following variational inequality [5, 6]

$$\begin{cases} 0 = \min\{g(x) - V_0(x,t), \frac{\partial}{\partial t} V_0(x,t) + H(\nabla V_0(x,t), x)\} & \text{in } x \in \mathbb{R}^n, t < 0 \\ V_0(x,0) = \min\{\ell(x), g(x)\} & \text{on } x \in \mathbb{R}^n, \end{cases} \tag{5}$$

where the Hamiltonian $H : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is given by $H(\lambda, x) = \max_{u\in\mathcal{U}} \min_{d\in\mathcal{D}} \lambda^\top f(x, u, d)$. Then, we have the following relationship

$$V_0(x,t) \geq 0 \iff x \in \mathcal{RA}_0(\mathcal{T}, \mathcal{F}, t). \tag{6}$$

Moreover, the gradient of the value function $V_0(x,t)$ informs the optimal control law $u^*(x,t)$

$$u^*(x,t) = \arg\max_{u\in\mathcal{U}} \min_{d\in\mathcal{D}} \nabla V_0(x,t)^\top f(x,u,d). \tag{7}$$

Following this control law ensures that any state x at time t such that $V_0(x,t) \geq 0$ reaches the target while avoiding the failure set, despite the best effort from player d.

3.2 Control Barrier Functions (CBFs)

For consistency with the original work [28], we introduce CBFs in the context of control-affine dynamics with no disturbances, i.e.

$$\dot{x} = f_0(x) + g_0(x)u, \tag{8}$$

where the functions $f_0 : \mathbb{R}^n \to \mathbb{R}^n$ and $g_0 : \mathbb{R}^n \to \mathbb{R}^{n\times p}$ are globally Lipschitz. Note that this formalism has been extended to more general dynamics, such as (1), with the key ideas unchanged [29]. Moreover, the system is considered safe as long as the state remains within a safe set $\mathcal{C} \subseteq \mathbb{R}^n$, defined as the zero-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$. To ensure the safety, we introduce concept of CBFs, as defined in the following.

Definition 3.1. (**Control Barrier Functions**, [28]) A continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ such that $\mathcal{C} = \{x \in \mathbb{R}^n \mid h(x) \geq 0\}$ is a control barrier function for (8) on $\mathcal{C}$ if there is an extended class $\mathcal{K}$ function[1] α such that, for each $x \in \mathcal{C}$, there exists a control $u \in \mathcal{U}$ satisfying

$$\nabla_x h(x)^\top (f_0(x) + g_0(x)u) + \alpha(h(x)) \geq 0. \tag{9}$$

If one can identify a CBF for a system, the following result provides the desired safety guarantee:

Theorem 1. *([28]) If h is a CBF for (8) on $\mathcal{C}$, and if $\nabla h(x) \neq 0$ for all $x \in \partial C$, then the set $\mathcal{C}$ is safe under any globally Lipschitz controller $u : \mathbb{R}^n \to \mathcal{U}$ for which (9) is satisfied with $u = u(x)$ at each $x \in \mathbb{R}^n$.*

Given any nominal control law $u_{\text{nom}} : \mathbb{R}^n \to \mathbb{R}^p$ that might violate control limits and safety, CBFs can be used to minimally adjust the nominal control input with the following optimization problem:

$$\begin{aligned} u^*(x) = \arg\min_{u\in\mathcal{U}} \|u - u_{\text{nom}}(x)\|_2^2 \\ \text{s.t. } \nabla_x h(x)^\top (f_0(x) + g_0(x)u) + \alpha(h(x)) \geq 0. \end{aligned} \tag{10}$$

Note that because the dynamics (8) are control-affine, then equation (10) is a quadratic program, enabling real-time safety filtering.

4 Safe & Reset-Friendly Learning via Viscosity-Based CBFs

Joint safety, defined as avoiding failure states, and liveness, defined as reaching a target, are commonly addressed through reach-avoid HJR or combined CLF-CBF approaches. While HJR ensures safety and liveness, it lacks a framework for minimally modifying a nominal controller in a smooth CBF-like fashion. In contrast, the CLF-CBF approach facilitates the construction of a safety filter, but ensuring feasibility becomes challenging in the presence of disturbances and control bounds. As such, both of these methods are ill-suited to prioritizing finite-time safety and liveness for a system while executing a nominal controller for a secondary objective. Such a need arises, for example, when one wishes for a system to explore an environment while ensuring it can avoid failure states and return to a safe reset position (e.g., a charging station) within some time.

[1] A function $\alpha : \mathbb{R} \to \mathbb{R}$ is said to be extended class $\mathcal{K}$ if α is continuous, strictly increasing, and satisfies $\alpha(0) = 0$.

In this section, we develop a framework that robustly and safely reaches a goal within a desired time, while optimizing online for different performance objectives over the time horizon. In Section 4.1, we develop a generalization of a CBF for reach-avoid sets that addresses safety and liveness under disturbances. The extension of these results to reach-avoid tubes is presented in Section 4.2. The design of a safety filter using our findings is covered in Section 4.3.

4.1 Viscosity-Based Control Barrier Functions for Reach-Avoid Sets

We consider a time-varying safe set $\mathcal{C}_v : (-\infty, 0] \to 2^{\mathbb{R}^n}$ that captures both finite-time safety and liveness. To achieve this objective, we introduce viscosity-based CBF (VB-CBFs) as follows.

Definition 4.1. (**Viscosity-Based Control Barrier Function**) Consider a continuous function $h_v : \mathbb{R}^n \times (-\infty, 0] \to \mathbb{R}$, and for each $t \leq 0$, let $\mathcal{C}_v(t) = \{x \in \mathbb{R}^n \mid h_v(x,t) \geq 0\}$. Then h_v is a viscosity-based control barrier function (VB-CBF) for system (1) on $\mathcal{C}_v(\cdot)$ if there exists an extended class $\mathcal{K}$ function α such that for all $t < 0$ and all $x \in \mathcal{C}_v(t)$, the inequality

$$\frac{\partial}{\partial t} h_v(x,t) + \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \nabla_x h_v(x,t)^\top f(x,u,d) \geq -\alpha(h_v(x,t)) \tag{11}$$

holds in a viscosity sense, i.e., for each continuously differentiable $\psi : \mathbb{R}^n \times (-\infty, 0) \to \mathbb{R}$

$$\frac{\partial}{\partial t} \psi(x,t) + \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \nabla_x \psi(x,t)^\top f(x,u,d) \geq -\alpha(h_v(x,t)) \tag{12}$$

at each (x,t) where $h_v - \psi$ has a local minimum.

Unlike a traditional CBF, which satisfies the global safety condition (9), a VB-CBF satisfies the local safety condition (12) anywhere such a ψ exists (in particular anywhere h_v is differentiable; see Lemma 1.7 in [27]). Recall that the value function V_0 is the viscosity solution to the variational inequality (5). The following result formalizes the connection between VB-CBF from Definition 4.1 and the time-varying reach-avoid safety problem (4).

Theorem 2. *The value function $V_0 : \mathbb{R}^n \times (-\infty, 0] \to \mathbb{R}$ is a VB-CBF for system (1) on the reach-avoid set $\mathcal{RA}_0(\mathcal{T}, \mathcal{F}, \cdot)$.*

The proof for this theorem is in the section 7.2 of the Appendix. The above theorem provides a useful method for constructing a VB-CBF on the reach-avoid set, namely computing the value function V_0.

4.2 Viscosity-Based Control Barrier Functions for Reach-Avoid Tubes

While Theorem 2 provides a VB-CBF for the reach-avoid set $\mathcal{RA}_0(\mathcal{T}, \mathcal{F}, t)$ (i.e., safely reaching the target *at* the final time), it is often desirable to safely reach the target *by* the final time. Therefore, in this section, we extend our result from previous section to reach-avoid tubes $\mathcal{RA}(\mathcal{T}, \mathcal{F}, t)$.

Definition 4.2. (**Robust Control Invariance**) A set $\mathcal{S} \subseteq \mathbb{R}^n$ is robustly control invariant if for each $t \leq 0$ and $x \in \mathcal{S}$, for all $\lambda \in \Lambda(t)$ there exists a $\boldsymbol{u} \in \mathbb{U}(t)$ such that $\xi_{x,t}^{\boldsymbol{u},\lambda[\boldsymbol{u}]}(s) \in \mathcal{S}$ for all $s \in [t, 0]$.

For simplicity, we will assume one can safely stay within the target set and outside the failure set once the target has been reached.

Assumption 2. The set $\mathcal{T} \setminus \mathcal{F}$ is non-empty and robustly control invariant.

Note that this assumption will usually be satisfied for our intended usage, where the target serves as a resetting set (e.g., a docking station). Therefore, Assumption 2 is not restrictive in practice.

Proposition 1. Under Assumption 2, the fixed-time reach-avoid set is the same as the reach-avoid tube, i.e. $\mathcal{RA}_0(\mathcal{T}, \mathcal{F}, t) = \mathcal{RA}(\mathcal{T}, \mathcal{F}, t)$ for each $t \leq 0$.

The proof for this proposition is in the section 7.1 of the Appendix. Intuitively, this proposition states that so long as the system can safely remain in the target once it arrives, reaching the target by any time guarantees that the system can be in the target at the final time. The following result is then immediate from Theorem 2 and Proposition 1.

Theorem 3. *Under Assumption 2, the value function $V_0 : \mathbb{R}^n \times (-\infty, 0] \to \mathbb{R}$ is a VB-CBF for system (1) on the reach-avoid tube $\mathcal{RA}(\mathcal{T}, \mathcal{F}, \cdot)$.*

The above theorem implies that the value function V_0 can be computed to construct a VB-CBF on the reach-avoid tube (rather than the reach-avoid set as before), provided that Assumption 2 holds.

4.3 Safety Filter

Given a VB-CBF h_v for system (1) and corresponding extended class $\mathcal{K}$ function α, we consider for $t \leq 0$ and $x \in \mathcal{C}_v(t)$ the feasible control set

$$\Omega(x,t) := \{u \in \mathcal{U} \mid \exists \psi : \mathbb{R}^n \times (-\infty, 0) \to \mathbb{R} \text{ continuously differentiable s.t. } h_v - \psi \text{ has a local minimum at } (x,t), \frac{\partial}{\partial t}\psi(x,t) + \min_{d \in \mathcal{D}} \nabla_x \psi(x,t)^\top f(x,u,d) \geq -\alpha(h_v(x,t))\}. \quad (13)$$

We use the feasible control set to design a safety filter that prioritizes system safety and liveness, while also addressing a secondary objective. Then, given a nominal control law $u_{\text{nom}} : \mathbb{R}^n \times (-\infty, 0] \to \mathcal{U}$ that might violate safety or prevent safely achieving the target set, whenever $\Omega(x,t)$ is non-empty, we minimally adjust the nominal control input via the following optimization problem

$$\begin{aligned} u^*(x,t) = \arg\min_{u \in \mathcal{U}} \|u - u_{\text{nom}}(x)\|_2^2 \\ \text{s.t. } \frac{\partial}{\partial t}\psi(x,t) + \min_{d \in \mathcal{D}} \nabla_x \psi(x,t)^\top f(x,u,d) \geq -\alpha(h_v(x,t)), \end{aligned} \quad (14)$$

where ψ is chosen as above. When the system dynamics are control and disturbance-affine, the optimization problem can be formulated as a quadratic program that can be solved efficiently online.

Remark 1. When computing $u^*(x,t)$ online, note that if h_v is differentiable at (x,t), it follows from Lemma 1.7 of [27] that one can simply solve the optimization problem (14) with h_v in place of ψ. Additionally, since V_0 is Lipschitz (c.f., [6]), V_0 is differentiable almost everywhere by Rademacher's Theorem. It follows from Theorems 2 and 3, that for $h_v = V_0$ we have that $\Omega(x,t)$ is non-empty almost everywhere in $\mathcal{RA}_0(\mathcal{T}, \mathcal{F}, \cdot)$ (and $\mathcal{RA}(\mathcal{T}, \mathcal{F}, \cdot)$ as well under Assumption 2). These observations justify that in practice, if one uses the value function V_0 computed via HJR as a VB-CBF, they may compute $u^*(x,t)$ by solving (14) with both ψ and h_v replaced with V_0. In this case, there will be some feasible control action at almost every x and t in the reach-avoid tube.

5 Numerical Experiments

5.1 Simulation Setup

Modified Cartpole Environment Setup: We demonstrate the the effectiveness of our approach by considering the cartpole swing-up task [30], modified such that all the mass is placed at the end of the pole, and damping and friction are removed. The state space of the cartpole system is given by $z = [x, \theta, \dot{x}, \dot{\theta}]^\top \in \mathbb{R}^4$, where x is the position of the cart, θ is the pendulum angle, and $\dot{x}$ and $\dot{\theta}$ are the velocity of the cart and the angular velocity of the pendulum. The state space of environment is constrained by $x \in [-1.8, 1.8]$. For our setting, we define position x as unsafe for $x < -1.5$ or $x > 1.5$, while the angular position θ is unsafe for $\pi/8 < \theta < \pi/4$, visualized in Figures 1 and 2 in red. These constraints limit the safe swing-up behavior of the system to the clockwise direction. These unsafe regions are not explicitly modeled in the simulator, thus safety violations do not directly impact performance. We consider the target set as a region where the robot must reset at the end of each episode. For this setup, the target region is specified as

$$\mathcal{T} = \left\{ z \mid x \in [-1.1, -0.8], \theta \in [-\pi - 0.25, \pi + 0.25], \dot{x} \in [-0.1, 0.1], \dot{\theta} \in [-0.25, 0.25] \right\}.$$

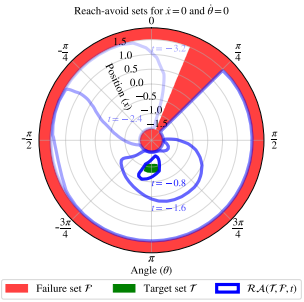

Figure 1: Reach-avoid computation for the modified cartpole environment. The reach-avoid tubes $\mathcal{RA}(\mathcal{T}, \mathcal{F}, t)$ indicate states from which the cartpole can reach the target $\mathcal{T}$ at times $t = \{-0.8, -1.6, -2.4, -3.2\}$ seconds while avoiding the failure set $\mathcal{F}$.

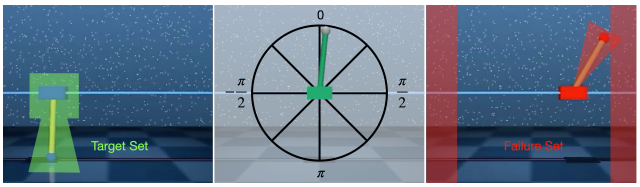

Figure 2: Our modified cartpole environment based on the Deepmind control suite. The green region in the left figure depicts the target region that we desire to reach, the center image depicts the swung up position that yields the highest reward and the red regions in the right image depict unsafe regions in the state space.

	SAC	SAC-CBF	**SAC-RACBF**	**SAC-RACBF-noreset**
% Unsafe Trajectories	99.437	0.563	0.070	0.000

Table 1: Average (over 5 seeds) percentage of unsafe trajectories during training. Our methods are bolded.

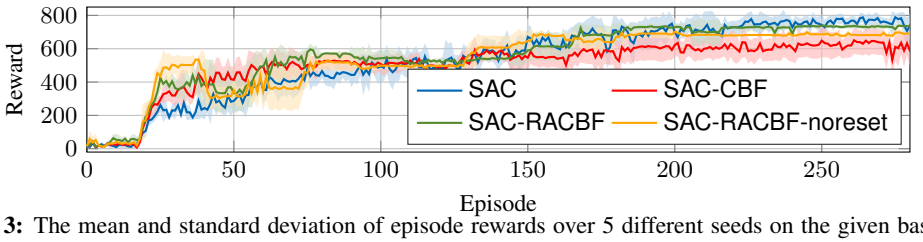

Figure 3: The mean and standard deviation of episode rewards over 5 different seeds on the given baselines (SAC, SAC-CBF) and proposed methods (SAC-RACBF and SAC-RACBF-noreset). The proposed methods achieve reward performances closely comparable to the baselines, demonstrating how the reach-avoid VB-CBF safety filter ensures safety, guarantees a safe return to the desired target set, and minimally impacts the learning process of the SAC agent. To ensure a fair comparison, we compare the reward returned over the first 10 seconds of the 15 second episode, as our method requires returning to the target set, unlike the baselines.

This target set is visualized in Figures 1 and 2 in green. Terminating an episode in this target region is not straightforward. For our system to efficiently achieve this state, it must actively dissipate energy while simultaneously moving to the desired position, balancing precise position control with energy-reducing dynamics. The zero-level sets of the reach-avoid tube $\mathcal{RA}(\mathcal{T}, \mathcal{F}, t)$ are shown over various times t in Figure 1. We consider the widely used RL method soft actor-critic (SAC) [31], which utilizes a maximum entropy RL objective to learn both a stochastic policy and a value function. We build on [32] implementation. However, to ensure safety during training, we run the output of SAC through a safety filter. By combining SAC with a reach-avoid VB-CBF safety filter we aim to learn to efficiently swing-up the cartpole to maximize reward, while avoiding safety violations and returning to the reset position (i.e. target set $\mathcal{T}$).

5.2 Results and Analysis

To evaluate the effectiveness of our approach, we compare the following methods:

1. **(Ours)** SAC with reach-avoid VB-CBF (SAC-RACBF): Our method integrates a reach-avoid VB-CBF with SAC, combining safety with a reachability objective to guide the agent.
2. **(Ours)** SAC with reach-avoid VB-CBF without resetting (SAC-RACBF-noreset): A modification of the above, where the next episode starts from the final state of the previous episode.
3. Standard SAC (SAC): The standard SAC algorithm, which does not account for safety constraints or the objective of returning to the start state, serves as our performance baseline.
4. SAC with CBF (SAC-CBF): A variant of SAC enhanced with an CBF-like safety filter. Inspired by the approach in [16], we construct a CBF-like safety filter following the same methodology as SAC-RACBF, but relying on the value function obtained from solving the avoid-only problem.

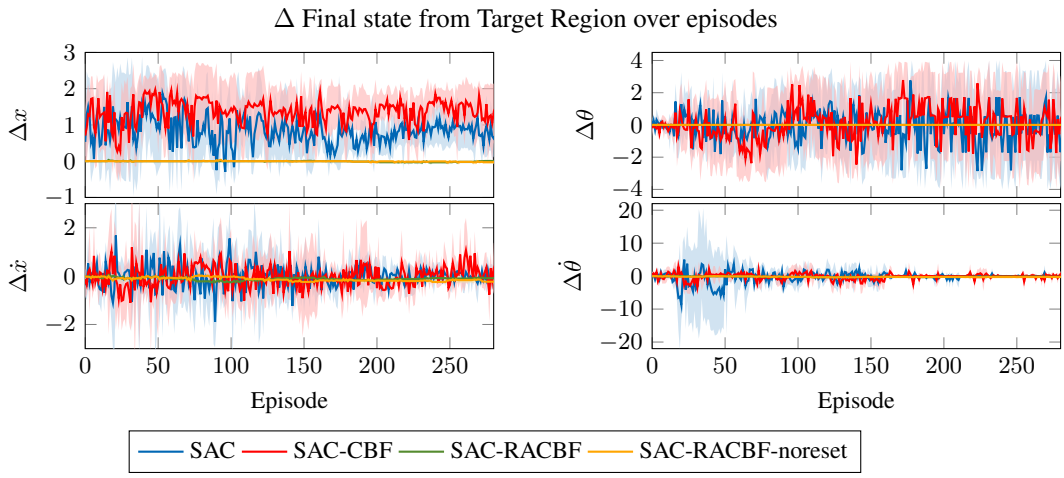

Figure 4: The mean and standard deviation of the difference between the final state and desired target region, averaged over the training episodes of 5 seeds. The proposed methods SAC-RACBF and SAC-RACBF-reset remain at Δ close to 0, indicating they successfully reach the target region, while the baselines (SAC and SAC-CBF) are distributed throughout the state space.

All methods and their results are averaged over 5 seeds. The reward graph over the course of training is illustrated in the Figure 3. These showcase the rewards accrued over the first 10 seconds of the 15 second episode, as we aim to terminate the episode at a safe reset state in the target set (which has a low reward for the swing-up objective). The safety-filter enhanced SAC agents agents accumulate rewards comparable to standard SAC, demonstrating that the safety filter based on the proposed method is minimally invasive and allows the cartpole agent to efficiently learn how to swing up. Importantly, unlike standard SAC, our methods preserve safety during learning. Table 1 shows how our methods are better at maintaining safety throughout the training, while standard SAC leads to unsafe trajectories in over 99% of the episodes. SAC-CBF provides similar levels of safety to our approaches, however, it does not provide guarantees on returning to the target set and thus terminates each episode far away from the desired safe return region. Figure 4 shows the minimum difference between the current state and target set, denoted by Δ, for each dimension. The graphs illustrate how our methods closely align with the desired target region throughout all episodes, while the baselines remain widely distributed across the state space.

In summary, the results in Table 1, Figures 3 and 4 demonstrate how our proposed method has minimal detrimental effect on performance and maintains safety, while enabling a key feature for real world RL, namely terminating an episode in a desired safe reset region. Furthermore, the performance of the proposed method SAC-RACBF-noreset highlights its capability for hands-off RL, enabling efficient and cost-effective robot training in real world scenarios. We acknowledge that a policy trained with the reach-avoid-based safety filter will require the continued use of this safety filter during online deployment. However, the modifications to the safety filter for a policy can be incorporated in a manner that eliminates their necessity post-training [23].

6 Conclusion

In this paper, we present a novel concept of reach-avoid VB-CBF that integrates CBFs with the HJR reach-avoid set. When combined with a safety filter, this approach prioritizes invariance with respect to the time-varying safe set while maintaining robustness to control and disturbance bounds. Moreover, we show that the Hamilton-Jacobi reach-avoid set is a reach-avoid VB-CBF. We motivate our approach with the promise of safe, efficient, hands-off RL for training robots in the real world. The effectiveness of our method is demonstrated through safe training and resetting in a cart-pole environment. For future work, we plan to carry out a comparison of our approach with other methods and leverage DeepReach to approximate the reach-avoid value function for high-dimensional and partially unknown dynamics using online updates from a conservative initial guess.

References

[1] S. Bansal, M. Chen, S. L. Herbert, and C. J. Tomlin. Hamilton-Jacobi reachability: A brief overview and recent advances. In *Proc. IEEE Conf. on Decision and Control*, 2017.

[2] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control Barrier Function based quadratic programs for safety critical systems. In *IEEE Transactions on Automatic Control*, volume 62, pages 3861–3876, 2017.

[3] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter. Multi-layered safety for legged robots via control barrier functions and model predictive control. In *Proc. IEEE Conf. on Robotics and Automation*, 2021.

[4] H. Zhu, J. Yu, A. Gupta, D. Shah, K. Hartikainen, A. Singh, V. Kumar, and S. Levine. The ingredients of real-world robotic reinforcement learning. In *Int. Conf. on Learning Representations*, 2020.

[5] K. Margellos and J. Lygeros. Hamilton–jacobi formulation for reach–avoid differential games. *IEEE Transactions on Automatic Control*, 56(8):1849–1861, 2011.

[6] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry. Reach-avoid problems with time-varying dynamics, targets and constraints. In *Hybrid Systems: Computation and Control*, 2015.

[7] S. Bansal and C. J. Tomlin. Deepreach: A deep learning approach to high-dimensional reachability. In *Proc. IEEE Conf. on Robotics and Automation*, 2021.

[8] K.-C. Hsu, V. Rubies-Royo, C. J. Tomlin, and J. F. Fisac. Safety and liveness guarantees through reach-avoid reinforcement learning. In *Robotics: Science and Systems*, 2021.

[9] G. Chênevert, J. Li, A. Kannan, S. Bae, and D. Lee. Solving reach-avoid-stay problems using deep deterministic policy gradients. *ArXiv*, abs/2410.02898, 2024.

[10] L. K. Chung, W. Jung, S. Pullabhotla, P. Shinde, Y. Sunil, S. Kota, L. F. W. Batista, C. Pradalier, and S. Kousik. Guaranteed reach-avoid for black-box systems through narrow gaps via neural network reachability. *ArXiv*, abs/2409.13195, 2024.

[11] D. Yu, H. Ma, S. Li, and J. Chen. Reachability constrained reinforcement learning. In *Int. Conf. on Machine Learning*, 2022.

[12] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. Ames, and M. N. Zeilinger. Data-driven safety filters: Hamilton-Jacobi reachability, Control Barrier Functions, and predictive methods for uncertain systems. *IEEE Control Systems*, 43:137–177, 2023.

[13] P. Mestres and J. Cortés. Feasibility and regularity analysis of safe stabilizing controllers under uncertainty. *Automatica*, 167:111800, 2024.

[14] 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*, 8:187249–187275, 2020.

[15] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames. Backup Control Barrier Functions: Formulation and comparative study. In *Proc. IEEE Conf. on Decision and Control*, 2021.

[16] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert. Robust Control Barrier-Value Functions for safety-critical control. In *Proc. IEEE Conf. on Decision and Control*, 2021.

[17] O. So, Z. Serlin, M. Mann, J. Gonzales, K. Rutledge, N. Roy, and C. Fan. How to train your neural Control Barrier Function: Learning safety filters for complex input-constrained systems. In *Proc. IEEE Conf. on Robotics and Automation*, 2024.

[18] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444, 2022.

[19] E. Altman. *Constrained Markov Decision Processes*. CRC Press, 1999.

[20] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg. Recovery RL: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3):4915–4922, 2021.

[21] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause. Safe model-based reinforcement learning with stability guarantees. In *Conf. on Neural Information Processing Systems*, 2017.

[22] J. J. Choi, F. Castañeda, C. J. Tomlin, and K. Sreenath. Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions. *ArXiv*, abs/2004.07584, 2020.

[23] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proc. AAAI Conf. on Artificial Intelligence*, 2019.

[24] Y. Emam, G. Notomista, P. Glotfelter, Z. Kira, and M. Egerstedt. Safe reinforcement learning using robust control barrier functions. *IEEE Robotics and Automation Letters*, pages 1–8, 2022.

[25] Y. Cheng, P. Zhao, and N. Hovakimyan. Safe and efficient reinforcement learning using disturbance-observer-based control barrier functions. In *Learning for Dynamics & Control*, 2022.

[26] E. A. Coddington and N. Levinson. *Theory of Ordinary Differential Equations*. McGraw-Hill, 1955.

[27] M. Bardi and I. Capuzzo-Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Birkhäuser, 1997.

[28] 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*, 62(8):3861–3876, 2016.

[29] S. N. Y. Kolathaya and A. Ames. Input-to-state safety with control barrier functions. *IEEE Control Systems Letters*, 3:108–113, 2018.

[30] S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, N. Heess, and Y. Tassa. dm control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

[31] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine. Soft actor-critic algorithms and applications. *ArXiv*, abs/1812.05905, 2018.

[32] D. Yarats and I. Kostrikov. Soft actor-critic (sac) implementation in pytorch. https://github.com/denisyarats/pytorch_sac, 2020.

7 Appendix

7.1 Proof for Proposition 1

Proof. Fix $t < 0$ (the proof is trivial when $t = 0$). It is clear that $\mathcal{RA}_0(\mathcal{T}, \mathcal{F}, t) \subseteq \mathcal{RA}(\mathcal{T}, \mathcal{F}, t)$. We prove the reverse inclusion. Suppose $x \in \mathcal{RA}(\mathcal{T}, \mathcal{F}, t)$. Let $\lambda \in \Lambda(t)$. By the definition of $\mathcal{RA}(\mathcal{T}, \mathcal{F}, t)$, we can choose $\boldsymbol{u} \in \mathbb{U}(t)$ and $s \in [t, 0]$ such that $\xi_{x,t}^{\boldsymbol{u},\boldsymbol{d}}(s) \in \mathcal{T}$ and $\xi_{x,t}^{\boldsymbol{u},\boldsymbol{d}}(\tau) \notin \mathcal{F}$ for all $\tau \in [t, s]$, where $\boldsymbol{d} := \lambda[\boldsymbol{u}]$. Let $\boldsymbol{u}_s = \boldsymbol{u}|[t, s]$ (i.e. the restriction of $\boldsymbol{u}$ to $[t, s]$), let $\boldsymbol{d}_s = \boldsymbol{d}|[t, s]$, and let $\lambda_s : \mathbb{U}(s) \to \mathbb{D}(s)$ be given by $\lambda_s : \boldsymbol{u}_0 \mapsto \lambda(\langle \boldsymbol{u_s}, \boldsymbol{u_0} \rangle)|[s, 0]$, where $\langle \boldsymbol{u_s}, \boldsymbol{u_0} \rangle$ represents the concatenation of $\boldsymbol{u_s}$ and $\boldsymbol{u_0}$. It can be readily checked that non-anticipativity of λ_s follows from non-anticipativity of λ. Let $x_s = \xi_{x,t}^{\boldsymbol{u},\boldsymbol{d}}(s)$. By Assumption 2, we can choose $\boldsymbol{u}_0^* \in \mathbb{U}(s)$ such that $\xi_{x_s,s}^{\boldsymbol{u}_0^*,\boldsymbol{d}_0^*}(\tau) \in \mathcal{T} \setminus \mathcal{F}$ for all $\tau \in [s, 0]$, where $\boldsymbol{d}_0^* := \lambda_s[\boldsymbol{u}_0^*]$. Letting $\boldsymbol{u}^* = \langle \boldsymbol{u}_s, \boldsymbol{u}_0^* \rangle$ and $\boldsymbol{d}^* = \langle \boldsymbol{d}_s, \boldsymbol{d}_0^* \rangle$, we then have by non-anticipativity of λ and the definition of λ_s that $\lambda[\boldsymbol{u}^*] = \boldsymbol{d}^*$. Thus $\xi_{x,t}^{\boldsymbol{u}^*,\boldsymbol{d}^*}(\tau) = \xi_{x,t}^{\boldsymbol{u},\boldsymbol{d}}(\tau) \notin \mathcal{F}$ for all $\tau \in [t, s]$. Moreover, $\xi_{x,t}^{\boldsymbol{u}^*,\boldsymbol{d}^*}(\tau) = \xi_{x_s,s}^{\boldsymbol{u}_0^*,\boldsymbol{d}_0^*}(\tau) \in \mathcal{T} \setminus \mathcal{F}$ for all $\tau \in [s, 0]$. In other words, $\boldsymbol{u}^* \in \mathbb{U}(t)$ is such that $\xi_{x,t}^{\boldsymbol{u}^*,\lambda[\boldsymbol{u}^*]}(0) \in \mathcal{T}$ and $\xi_{x,t}^{\boldsymbol{u}^*,\lambda[\boldsymbol{u}^*]}(\tau) \notin \mathcal{F}$ for all $\tau \in [t, 0]$. Thus $x \in \mathcal{RA}_0(\mathcal{T}, \mathcal{F}, t)$, completing the proof. □

7.2 Proof for Theorem 2

Proof. First, observe that by (6), $\mathcal{RA}_0(\mathcal{T}, \mathcal{F}, t)$ is indeed the zero-superlevel set of $V_0(\cdot, t)$ for each $t \leq 0$. Fix $t < 0$ and $x \in \mathcal{RA}_0(\mathcal{T}, \mathcal{F}, t)$. Let $\psi : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function such that $V_0 - \psi$ has a local minimum at (x, t). Since V_0 is a viscosity solution of (5), then

$$0 \leq \min\{g(x) - V_0(x, t), \frac{\partial}{\partial t}\psi(x, t) + \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \nabla\psi(x, t)^\top f(x, u, d)\}.$$

In particular, we obtain

$$\frac{\partial}{\partial t}\psi(x, t) + \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \nabla\psi(x, t)^\top f(x, u, d) \geq 0 \geq -\alpha(V_0(x, t)),$$

for any extended class $\mathcal{K}$ function α, where the second inequality holds because $V_0(x, t) \geq 0$. □