# Safe Learning in the Real World via Adaptive Shielding with Hamilton-Jacobi Reachability

**Michael Lu** MICHAEL_LU_3@SFU.CA
**Jashanraj Singh Gosain** JSG31@SFU.CA
**Luna Sang** LUNA.SANG21@GMAIL.COM
**Mo Chen** MOCHEN@CS.SFU.CA
*School of Computing Science, Simon Fraser University, Burnaby BC, Canada, V5A 1S6*

**Editors:** N. Ozay, L. Balzano, D. Panagou, A. Abate

## Abstract

We present a robust shielding framework using Hamilton-Jacobi (HJ) Reachability that can be combined with any off-policy Reinforcement Learning algorithm to enable safer learning. Using an approximate model of a system dynamics, our method can capture the local model mismatch from a safety perspective. This leads to a more conservative safety filter that can adapt to the model mismatch. Compared to using a fixed safety filter, our approach leads to less safety violations. Using a Turtlebot 2, we demonstrate that our method can allow for safe learning in the real-world with minimal human intervention.

**Keywords:** Hamilton-Jacobi Reachability, Safe Reinforcement Learning

## 1. Introduction

In safe reinforcement learning (RL) (Gu et al., 2022b), an agent aims to maximize a reward function while adhering to safety constraints through trial-and-error (Sutton and Barto, 2018; Altman, 1999). These constraints typically encode desirable (or critical) behavioral boundaries, ensuring the agent operates safely (Wachi et al., 2024). However, without prior knowledge of the agent's dynamics or the environment, learning can be dangerous or unsafe. For example, an agent could unknowingly apply unsafe controls causing catastrophic damage. For high-stakes applications such as autonomous robotics (Isele et al., 2018), guaranteeing safety during learning is paramount. Towards obtaining safety guarantees, we focus on *safety filter* methods (Hsu et al.) which modifies an agent's control to ensure constraint satisfaction.

A key challenge in safety filter methods lies in designing an effective monitor and safety controller. The monitor determines when to override the agent's control, while the safety controller produces a safe alternative. Prior work addresses this by learning a safety critic (to predict violations) and a recovery policy (to generate safe controls) purely through trial-and-error (Srinivasan et al., 2020; Thananjeyan et al., 2021; Wilcox et al., 2022; Yu et al., 2022b; Thomas et al., 2022). Although these methods achieve strong empirical performance, they do not explicitly minimize the cumulative safety violations during learning. Alternatively, with partial knowledge of the dynamics, methods such as control barrier functions (CBFs) (Ames et al., 2019) or Hamilton-Jacobi (HJ) reachability (Bansal et al., 2017) can be used to construct safe controls.

When the agent's complete dynamics are known, HJ reachability provides rigorous safety guarantees under worst-case control and adversarial disturbances (Bansal et al., 2017). Specifically, HJ

rechability analysis provides a Backwards Reachable Tube (BRT) which represents the set of starting states that will inevitably lead a dynamical system into an unsafe region under adversarial control. However, BRT computation requires discretizing the state space into an grid before numerically solving. Since the number of grid points grows exponentially with respect to the dimension of the state space, the BRT is often infeasible to compute for high-dimensional systems. Recent works has leveraged neural networks to approximate BRTs (Bansal and Tomlin, 2020) via self-supervision or through RL techniques such as Q-learning (Fisac et al., 2019b; Hsu* et al., 2021).

**Contributions.** In this paper, we propose a robust HJ-CBF safety filter that leverages prior (but potentially inaccurate) dynamics models to enable safer real-world learning. Unlike in prior work, where a BRT is learned through trial-and-error, we argue that prior models of a system's dynamics are beneficial – even if it is inaccurate. We first show a lower-bound of the number of safety violations needed to approximate an accurate BRT using standard RL methods such as value iteration. Next, to address the model mismatch, we introduce a relaxed CBF-like constraint when designing a safety constraint using the mismatched dynamics. The relaxed constraint captures the local model mismatch from a safety perspective. While the immediate safety controller does not guarantee safety, our method is able to adapt the empirically estimated model's inaccuracy at runtime. Specifically, feedback from the safety constraint is used to update a safety threshold to make the safety filter more conservative. In contrast to previous approaches which uses a fixed safety threshold, our method is able to be robust to model mismatch. Finally, we empirically evaluate our method and compare it to a least-restrictive safety filter both in simulation and when training a robot in the real-word.

## 2. Related Work

**Safe Reinforcement Learning via Safety Filters** To reduce the number of times an agent must violate a safety constraint, it is common to design a safety filter to ensure only safe controls are used. For example, a human can interrupt an agent when it is unsafe (Saunders et al., 2017). Additionally, CBFs can be learned from expert demonstrations (Robey et al., 2020) which can then be combined with RL (Cheng et al., 2019; Choi et al., 2020). Since HJ reachability can provide strong safety guarantees, recent work have tried to utilize this property in safe RL methods (Ganai et al., 2024). Given access to the true dynamics, prior work has leveraged reachable sets as safety filters (Selim et al., 2022; Gu et al., 2022a; Kochdumper et al., 2023; Nakamura et al., 2025). Otherwise, when the model is not known, reachability-based safety critics have been learned through trial-and-error in simulation (Ganai et al., 2023; Hsu et al., 2023; Yu et al., 2022a; Chen et al., 2021; Kim et al., 2021).

**Neural Reachable Tubes.** Numerical methods (Osher et al., 2004) are often used to compute the solution of the HJB PDE to recover BRTs. These methods discrete the state and action space and are solved using dynamic programming. Unfortunately, the computational and space complexly grows exponentially with respect to the dimensional of the state space. Recent advances have used deep neural networks to obtain approximate solutions either using model-based (Bansal and Tomlin, 2020) or model-free (Fisac et al., 2019a; Hsu* et al., 2021; Wang et al., 2024) methods. Additionally, recent work have investigated verifying and correcting the accurately of approximated neural reachable tubes (Lin and Bansal, 2023a,b). In particular, Fisac et al. (2019a); Hsu* et al. (2021) proves that standard methods in RL such as value-iteration and Q-learning (Sutton and Barto, 2018) can be used to recover approximate BRTs, when the dynamics are unknown. However, these methods instead requires interactions with an environment or a simulator. Hence, cannot be used when learning solely in the real-world.

## 3. Background

### 3.1. Safe Reinforcement Learning

An infinite-horizon discounted constrained Markov decision process (CMDP) (Altman, 1999) is defined by the tuple $(\mathcal{X}, \mathcal{U}, \mathcal{P}, r, c, \rho, \gamma)$, where $\mathcal{X}$ is the set of states, $\mathcal{U}$ is the set of controls, $\mathcal{P} : \mathcal{X} \times \mathcal{U} \to \Delta_{\mathcal{X}}$ is the transition probability function, $r : \mathcal{X} \times \mathcal{U} \to [0, 1]$ is the reward function, $c : \mathcal{X} \times \mathcal{U} \to [0, 1]$ is the cost function, $\rho \in \Delta_{\mathcal{X}}$ is the initial state distribution, and $\gamma \in [0, 1)$ is the discount factor. For a policy $\pi : \mathcal{X} \times \mathcal{U} \to \Delta_{\mathcal{U}}$ where $\Delta_{\mathcal{U}}$ is the probability simplex in $\mathbb{R}^{|\mathcal{U}|}$ and reward function $r$, the *action-value function* $Q_r^\pi : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is defined as: $Q_r^\pi(x, u) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(x_t, u_t)]$ with $x_0 = x$, $u_0 = u$ and for $t \geq 1$, $x_{t+1} \sim \mathcal{P}(\cdot|x_t, u_t)$ and $u_{t+1} \sim \pi(\cdot|x_t)$. The *value function* $V_r^\pi : \mathcal{X} \to \mathbb{R}$ is defined such that $V_r^\pi(x) := \mathbb{E}_{u\sim\pi(\cdot|x)}[Q_r^\pi(x, u)]$. Additionally, for a cost function $c$, the respective action-value functions $Q_c^\pi$ and value functions $V_c^\pi$ are defined similarly. The agent's objective is to learn a policy $\pi$ that maximizes the cumulative reward subject to satisfying a safety constraint. Formally,

$$\max_\pi \mathbb{E}_{x\sim\rho}[V_r^\pi(x)] \quad \text{subject to} \quad \mathbb{E}_{x\sim\rho}[V_c^\pi(x)] \leq \Lambda \tag{1}$$

where $\Lambda \in \mathbb{R}_+$ is an upper-bound of the safety threshold. Typically to solve Equation (1), a Lagrangian is constructed and an iterative primal-dual update is used to solve the following:

$$(\pi^*, \lambda^*) = \arg\min_{\lambda\geq 0} \max_\pi \mathcal{L}(\pi, \lambda), \quad \text{where } \mathcal{L}(\pi, \lambda) = \mathbb{E}_{x\sim\rho}[V_r^\pi(x) - \lambda(V_c^\pi(x) - \Lambda)] \tag{2}$$

where $\lambda$ is the Lagrangian multiplier, $\pi^*$ is the optimal primal variable and $\lambda^*$ is the optimal dual variable. In practice $\mathbb{E}_{x\sim\rho}[V_r^\pi(x)]$ and $\mathbb{E}_{x\sim\rho}[V_c^\pi(x)]$ are approximated using Monte-Carlo or Temporal Difference Methods. In either case, agent must experience unsafe behaviour in order to learn $\mathbb{E}_{x\sim\rho}[V_c^\pi(x)]$. Finally, let $\pi_\theta$ represent a policy where $\theta$ are the parameters of a neural network.

### 3.2. Hamilton-Jacobi Reachability

To relate the CMDP formulation to HJ reachability, let $x \in \mathbb{R}^n$ be the state of a dynamical system with control $u \in \mathcal{U}$, disturbance $d \in \mathcal{D}$ and dynamics $\dot{x} = f(x, u, d)$. The sets $\mathcal{U}$ and $\mathcal{D}$ are assumed to be compact and the dynamics $f$ is assumed to be control affine, Lipschitz continuous, uniformly continuous and bounded.

To ensure our agent remains safe throughout learning, we will formalize our problem under the HJ reachability formulation. Let $\xi_x^{u,d}$ denote the trajectory achieved under the control $u(\cdot)$ and disturbances $d(\cdot)$ when starting at state $x$ over a possibly infinite time horizon. Denote the failure set which represents states we wish the agent to avoid as $\mathcal{K}$. Let $l : \mathcal{X} \to \mathbb{R}$ be a Lipschitz function such that $\mathcal{K}$ is equal to the zero sublevel set of $l$ i.e., $x \in \mathcal{K} \iff l(x) < 0$. Typically $l$ is chosen to be the signed distance function in order to make subsequent computations interpretable. It intuitively describes the safety margin of how close an agent is to an unsafe state. The overall objective is modeled as a two-player zero-sum game between the controller and disturbance policy with the following continuous payoff

$$J^{u,d}(x) := \min_{t\in\mathbb{N}} l(\xi_x^{u,d}(t)). \tag{3}$$

The HJ value function is defined as $V(x) := \max_u \min_d J^{u,d}(x)$ which represents the minimum distance a controller policy can bring the system state, under the worst-case disturbance policy to the

failure set. Additionally, the HJ value function is known to satisfy the following two-player dynamic programming Isaacs equation (Hsu et al., 2023)

$$V(x) := \max_{u} \min_{d} \min(g(x), V(f(x,u,d))). \tag{4}$$

For a given value function $V$ that satisfies Equation (4), the BRT can be obtained as the sub-zero level set: BRT $:= \{x \in \mathcal{X} \mid V(x) \leq 0\}$. This implies that for any state $x$ such that $V(x) > 0$, there always exists a control that guarantees $f$ to be safe under any disturbance. The optimal safety control and disturbance can then be computed as follows:

$$u^*(x) = \arg\max_{u} \min_{d} \frac{\partial V(x)}{\partial x}^\top f(x,u,d), \quad d^*(x) = \arg\min_{d} \max_{u} \frac{\partial V(x)}{\partial x}^\top f(x,u,d). \tag{5}$$

Moreover, any control from the set $\{u \in \mathcal{U} \mid \min_d \frac{\partial V(x)}{\partial x}^\top f(x,u,d) \geq 0\}$ is guaranteed to be keep the agent safe, even if it is sub-optimal. For a more detailed explanation of HJ reachability, readers are encouraged to read (Bansal et al., 2017).

### 3.3. Shielding Unsafe Control Using Safety Filters

Recent work has leveraged *safety filters* (Hsu et al.) to ensure a policy $\pi_\theta$ is safe during learning. A safety filter $\phi : \mathcal{X} \times \mathcal{U} \to \mathcal{U}$ consists of two key components: a fallback policy $\pi_{\text{safe}} : \mathcal{X} \to \Delta_{\mathcal{U}}$ and a monitor. The fallback policy $\pi_{\text{safe}}$ provides a control action to ensure safety, while the monitor determines when to activate the fallback policy. Depending on the construction of the monitor, safety can be guaranteed at all times. For example, using HJ reachability, one can design a least-restrictive safety filter (Hsu et al.; Fisac et al., 2018), assuming the dynamics $f$ are known:

$$\phi(x, \pi_\theta(x)) := \begin{cases} \pi_\theta(x) & V(x) \geq \epsilon \\ \pi_{\text{safe}}(x) & \text{otherwise} \end{cases} \quad \text{(Least-Restrictive Safety Filter)} \tag{6}$$

where $\epsilon \geq 0$ is a hyper-parameter to select the level set of the $V(x)$ used for intervention. In this setting $\pi_{\text{safe}}(x)$ can either be the solution of Equation (5) or any safe admissible control, i.e., $\{u \in \mathcal{U} \mid \min_d \frac{\partial V(x)}{\partial x}^\top f(x,u,d) \geq 0\}$ (He et al., 2023; Chen et al., 2018). This approach allows policy $\pi_\theta$ to explore and learn autonomously in most of the state space. However, close to the boundary of the BRT, as indicated by the value of $V(x)$, the policy is then switched over to $\pi_{\text{safe}}$. According to Hsu et al., Corollary 1.1, the safety filter maintains all-time safety. Although $\epsilon = 0$ suffices for theoretical guarantees, $\epsilon > 0$ is often chosen to account for (dynamics) model mismatch and approximation errors in BRT computations.

There are several caveats to using a least restrictive safety controller. First, abrupt control switching in this least restrictive approach can cause the resulting policy to be jittery. This could cause unwanted wear and tear to the robot. Secondly, the resulting safe control $\pi_{\text{safe}}(x)$ may not align with the desired control from $\pi_\theta(x)$. While the safe control from $\pi_{\text{safe}}(x)$ ensures the robot will always be safe, the resulting control may not maximize reward. To better align the desired control to the safe control near the boundary of V, i.e., $V(x) = \epsilon$, Borquez et al. (2023) proposes to use the following HJ-CBF filter constraint obtain $\pi_{\text{safe}}(x)$,

$$\pi_{\text{safe}}(x) = \arg\min_{u \in \mathcal{U}} \|u - \pi_\theta(u|x)\|_2^2 \quad \text{subject to} \quad \frac{\partial V(x)}{\partial x}^\top f(x,u,d^*) \geq -\alpha V(x) \tag{7}$$

where $\alpha \geq 0$ is a hyper-parameter. The choice of $\alpha$ dictates how to blend $\pi_\theta(x)$ with the optimal control. As $\alpha \to 0$, we recover the smooth least-restrictive approach, where the closest safe control to $\pi_\theta(x)$ is chosen. When $\alpha > 0$, the addition of $V(x)$ to the RHS limits the rate of change of $V(x)$ depending on $V(x)$, even when $V(x)$ is not close to 0. Assuming that $V(x) \geq 0$, Borquez et al. (2023, Lemma 3) ensures that the resulting control is always feasible for any $\alpha \geq 0$. Additionally, if $f$ is control-affine, Equation (7) is convex with respect to the control $u$, enabling the HJ-CBF filter to be solved quickly and easily used in online settings.

Typically obtaining solving Equation (4) is numerically intractable for large state spaces ($|\mathcal{X}| > 6$) due to the curse of dimensionality (Bui et al., 2025). However, a recent line of work have leveraged approximating BRTs (Fisac et al., 2019a; Hsu et al., 2023) using common RL algorithms with function approximation using neural networks. In the following section, we will first discuss the limitations of approximating BRTs through interactions with an environment.

## 4. Limitations of Bridging Hamilton-Jacobi Reachability and Reinforcement Learning

For simplicity, we will restrict our analysis to the single player setting, where there is an absence of adversarial disturbance. This is to abstain from finding the solution of a minmax two-player zero-sum game. The *Safety Bellman Operator* (Definition 1) introduced in Fisac et al. (2019a) provides a method to recover BRT using standard methods in RL such as value iteration or Q-learning. Since reachability analysis is formulated in continuous settings, while RL is often analyzed in the discrete setting, let $X \subseteq \mathcal{X}$ and $U \subseteq \mathcal{U}$ be finite discretizations of the respective state and control spaces. Additionally, let $\bar{f} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^{\mathcal{X}}$ be a discrete approximation of the system dynamics. In this setting, (Fisac et al., 2019a, Theorem 1) has show that the Safety Bellman Operator is a contraction mapping with modulus $\gamma$ (i.e. $\|\mathcal{T}u - \mathcal{T}v\|_\infty \leq \gamma\|u - v\|_\infty$). This implies that the discrete-time approximation of an BRT can be recovered as $\gamma \to 1$.

**Definition 1 (Safety Bellman Operator)** *For $\gamma \in (0, 1]$, the safety bellman operator $\mathcal{T} : \mathbb{R}^{|X|} \to \mathbb{R}^{|X|}$ is defined such that for vector $V \in \mathbb{R}^{|X|}$,*

$$[\mathcal{T}V](x) := (1 - \gamma)l(x) + \gamma \min\Big(l(x), \max_{u'} V(x + \bar{f}(x, u')\Delta t)\Big). \tag{8}$$

In the following theorem, we lower-bound the number of safety violations necessary to obtain an $\delta$-close safety value function where $\delta$ represents the maximum error between the approximated and true discounted value function. Here, we define a safety violation if $x \in \mathcal{K}$. Additionally, we note that the analysis can be extended to the respective optimal state-action safety value function when the dynamics $\tilde{f}$ are unknown.

**Theorem 2** *For a target $\delta > 0$, using Equation (8) with a fixed $\gamma \in (0, 1)$ requires at least $\Omega\Big(|K|\frac{\log(1/\delta\,(1-\gamma))}{1-\gamma}\Big)$ safety violations to obtain an $\delta$-close safety value function.*

**Proof** Without loss of generality, assume that $l(x) \in [0, 1]$. Since $\mathcal{T}$ is a contraction mapping, using Banach's Fixed Point Theorem, there exists a fixed point $v^*_\gamma \in \mathbb{R}^{|X|}$ such that $\mathcal{T}v^*_\gamma = v^*_\gamma$. When applying the safety bellman operator, we iterate over each state $x \in X$ and will violate the safety constraint whenever $x \in \mathcal{K}$. After each time $\mathcal{T}$ is applied to $v_t$, we have $\big\|v^*_\gamma - v_{t+1}\big\|_\infty \leq \big\|v^*_\gamma - v_t\big\|_\infty$. Applying this operation $T$ times yields, $\|v^* - v_{T+1}\|_\infty \leq \gamma^T\|v^* - v_1\|_\infty \leq \frac{\gamma^T}{1-\gamma}$. Setting $T \geq \frac{\log(1/\delta\,(1-\gamma))}{1-\gamma} \geq \frac{\log(1/(\delta\,(1-\gamma))}{\log(1/\gamma)}$ ensures $\big\|v^*_\gamma - v_{t+1}\big\|_\infty \leq \delta$. ■

We note that this is a conservative lower bound since the analysis is only for the initial set of unsafe states. We stress that using standard RL algorithms to learn reachable sets is dangerous in the real-world because agents must enter unsafe states. Therefore, it is imperative that we investigate how prior knowledge of an agent's dynamics can be utilized to help an agent learn safely. We argue that using techniques to approximate reachable sets using system decomposition (Li and Chen, 2020) or NN PDE solvers (Bansal and Tomlin, 2020) are practical even though these aforementioned methods are known not able to recover the true reachable set.

## 5. Method: The Robust HJ-CBF Filter

Recall that in Equations (6) and (7), constructing $\pi_{\text{safe}}(x)$ requires HJ value function $V$ to be computationally tractable and the dynamics $f$ to be known. If $f$ cannot be modeled exactly, the model mismatch can be addressed by selecting an appropriate bound on the disturbances $\mathcal{D}$. Unlike the approach presented in Fisac et al. (2018), we assume that $\mathcal{D}$ is fixed. For cases where the model mismatch error exceeds the disturbance bound, we will address this issue shortly. To ensure that the computation of the BRT is tractable or can be approximated, following Lyu and Chen (2020), we will assume that there exists a low-dimensional approximated model $\tilde{f} : \tilde{\mathcal{X}} \times \mathcal{U} \times \mathcal{D} :\to \tilde{\mathcal{X}}$ of $f$ where $\tilde{\mathcal{X}} \subseteq \mathcal{X}$. This implies that a subset of the full MDP state space can be effectively modeled by $\tilde{f}$. Furthermore, we assume that the control dimensions and bounds for $\tilde{f}$ are consistent with those of $f$. For example, a visual navigation controller composed of an RGB image and odometer results in the following state $x = (I(z), z)$ where $z$ is the state of the robot and $I(z)$ is the resulting RGB image. While $x$ is high-dimensional, there exists a subset of the state-space that can be low dimensional.

Let $\tilde{f}$ be an approximation of $f$ under the same assumptions as described above and let $V_{\tilde{f}}$ be the BRT computed with dynamics given by $\tilde{f}$. In this setting, Equation (7) may not always be feasible. This can occur because the invariance of $\{V_{\tilde{f}}(x) \geq 0\}$ is no longer guaranteed under model mismatch. Specifically, infeasibility can occur for states within the zero sub-level set of $V_{\tilde{f}}$. To ensure feasibility, we first consider the following relaxed CBF-like constraint to determine the safe controller:

$$\pi^{\beta}_{\text{safe}}(x) = \arg\min_{u \in \mathcal{U}} \min_{\xi \geq 0} \|u - \pi_\theta(x)\|_2^2 + \beta\,|\xi| \quad \text{s.t} \quad \frac{\partial V_{\tilde{f}}(x)}{\partial x}^\top \tilde{f}(x, u, d^*) \geq -\alpha V_{\tilde{f}}(x) - \xi, \quad (9)$$

where $\beta > 0$ is an additional hyper-parameter. Compared to Equation (7), we now have the CBF-like constraint using the approximated dynamics $\tilde{f}$ with an additional slack variable $\xi$ that makes the optimization always feasible. To understand the other benefits of this relaxed constraint, let us first consider when $\alpha = 0$. Intuitively, $\xi$ captures the local model mismatch from a safety perspective. The slack variable $\xi$ is positive only when the model mismatch is detrimental to maintaining safety. Sometimes the system may be safer than the dynamics $\tilde{f}$ may imply; in these cases $\xi$ would still be 0. If a large $\xi$ was needed, then the monitor needs to switch to the safe controller earlier to ensure safety for the next time step. To act conservatively, let $\bar{\xi} := \max_t \xi_t$ be the largest slack variable over the course of the filter. All together, this results in the following robust HJ-CBF safety filter:

$$\phi(x, \pi_\theta(x)) := \begin{cases} \pi_\theta(x) & V_{\tilde{f}}(x) \geq \max\{\kappa\,\bar{\xi}, \epsilon\} \\ \pi^{\beta}_{\text{safe}}(x) & \text{otherwise.} \end{cases} \quad \text{(Robust HJ-CBF Safety Filter)} \quad (10)$$

where $\kappa \geq 0$ is a hyper-parameter to account for longer time-horizons. Note that the shield $V_{\tilde{f}}(x) \geq \max\{\kappa\,\bar{\xi}, \epsilon\}$ is different than in Equation (6) since due to potentially large model-mismatch

errors, we cannot be sure that $V_{\tilde{f}}(x) \geq 0$ ensures that $\pi_\theta(x)$ is always safe. Under the assumption that the model-mismatch error is bounded, i.e. $\left\| f - \tilde{f} \right\|_\infty \leq \epsilon$, there exists $\kappa' \geq 0$ such that $\{x \mid V_{\tilde{f}}(x) \geq \kappa' \epsilon\} \subseteq \{x \mid V(x) \geq 0\}$. In other words, some larger super level set of $V_{\tilde{f}}$ is contained within the HJ value function of the true dynamics $V$. Since $\epsilon$ and $\kappa$ are unknown, we try to estimate their values by considering the worst slack variable over time.

To summarize, since $\tilde{f}$ is only an approximation of $f$, the monitor must be proactive and intervene earlier according to Equation (9) and Equation (10). Compared to similar safety filters such as the least-restrictive approach with the optimal control (Equation (6)) or from the original HJ-CBF filter (Equation (7)), our approach explicitly adjusts the monitor over time. If the approximate model $\tilde{f}$ matches the true dynamics $f$, then our method provides the same safety guarantees as prior approaches. Otherwise, our proposed robust HJ-CBF shield can adapt to the observed local model mismatch and apply the safety controller more conservatively in a principled manner. This is beneficial when employing safety filter in continuous learning tasks in the real world, where the dynamics of the robot may unknowingly change.

These shielding methods can be applied to an off-policy deep RL algorithm. This can be done by "wrapping" the policy $\pi_\theta$ with shield $\phi$ as described in Equations (6) and (10). Since the safe control from $\pi^\beta_{\text{safe}}$ is not sampled from the current policy $\pi_\theta$, off-policy algorithms must be used. In the following section, we apply our method to train a policy safety in simulation and fully in the real-world.

## 6. Simulated Experiments

As a test bed, we first consider simulated experiments. We aim to address the following questions: (1) How many safety violations are required to learn $V$ using an DQN? (2) Under what conditions can our method still ensure that agent remains safe while learning? (3) What are the failure conditions of our method? For all tasks, we assume that $\tilde{f}$ can be used to compute the corresponding BRT $V_{\tilde{f}}$.

We first consider toy environments where the state-space is small to analysis the effects of mispecified dynamics. The goal is for a robot to reach a goal location without collide with an obstacle. In each task, we use the following Dubins car model for the dynamics:

$$\dot{x} = v\cos(\theta) \quad \dot{y} = v\sin(\theta) \quad \dot{\theta} = \omega \tag{11}$$

where $\omega$ is the control input and $v = 0.6$ is fixed velocity $(m/s)$. Here we consider `easy`, `aligned`, and `hard` variants depending on the model mismatch. In the `easy` environments, we assume that the approximate dynamics with $\omega \in [-0.75, 0.75]$ while the true dynamics has $\omega \in [-0.5, 0.5]$. Likewise, for the `hard` environment, we assume that the approximate dynamics with $\omega \in [-0.75, 0.75]$ while the true dynamics has $\omega \in [-0.5, 0.5]$ This ensures the sub zero-level set of $V_{\tilde{f}}$ may not ensure the safety of $f$. In `aligned`, there is no model mismatch. For all environments, we use the OptimizedDP (Bui et al., 2025) library to compute required BRTs.

### 6.1. Results Analysis

To focus on mainly on how well various methods shielding methods can keep an agent safe, we train a policy using DroQ (Hiraoka et al., 2021) with Lagrangian constraints from the following implementation (Bradbury et al., 2018). For all environments, when using the robust HJ-CBF safety filter, we set $\beta = 1$, $\kappa = 1$, $\gamma = 1.0$ and $\epsilon = 0.1$. As a baseline, we compare to a least-restrictive

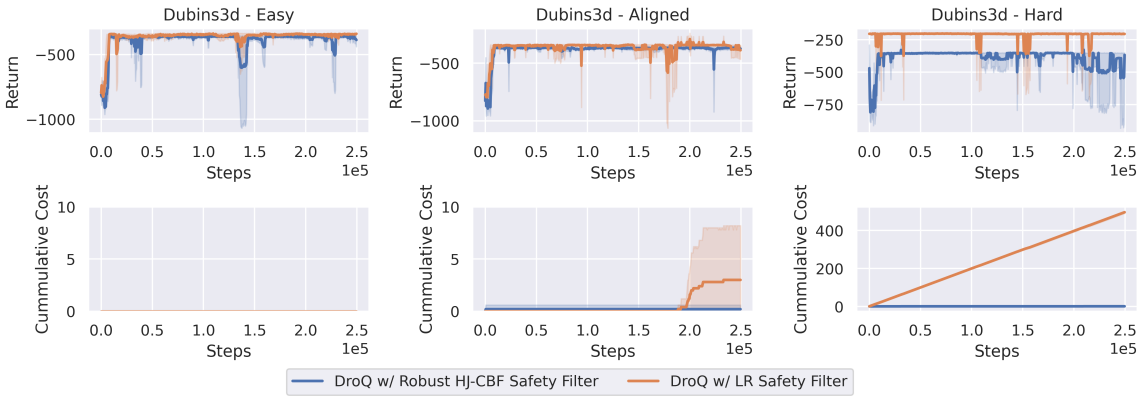


Figure 1: Results of applying DroQ with different shields. We report the average return and cumulative safety violations over 250k samples across 5 random seeds. We find that our proposed method minimizes the cumulative cost (safety violations) when there is high model mismatch.

shielding defined in Equation (6) with a fixed $\epsilon = 0.1$. From Figure 1, we conclude that robust HJ-CBF Safety filter results in fewer safety violations when compared to a least-restrictive approach. While both methods are able to quickly learn a good policy to reach a target goal, their differ in how well they can adapt to model mismatch.

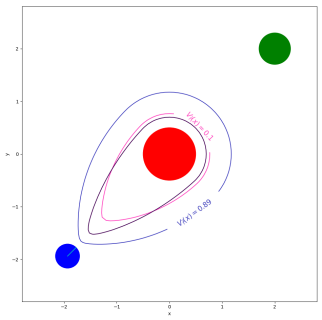

Figure 2: Various Level Sets of `Dubins3d-hard`

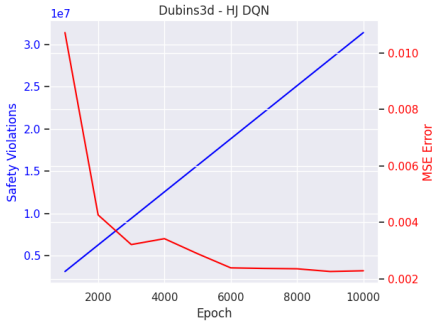


Figure 3: The effect of feature conditions on the global convergence.

In Figure 2, we depict the `Dubins3d-hard` environments with various level sets. The black contour represents the zero-level set of the BRT when using the true dynamics, i.e $V(x) = 0$. The other pink and blue contours are the level sets of $V_{\tilde{f}}$ of various safety filters. Since $\epsilon$ is fixed, the least-restrictive approach switches to the safe controller when the robot is within pink contour. Since pink level does not fully contain the black contour, safety violations are still possible. In comparison, our robust HJ-CBF safety filter saturates at a more conservative level-set. This allows the agent to learn with minimal safety violations.

In Figure 3, we depict the a learned BRT in the `Dubins3d-hard` environment using an DQN (Mnih et al., 2013). We compare the learned BRT to an numerically obtained BRT and report the mean squared error (MSE) and number of safety violations.

## 7. Real-World Experiments

### 7.1. Setup

Figure 4: The goal of the robot is to navigate towards from the starting to end pose while avoiding orange cones and leaving the designated learning boundary. The green tape represents the boundaries where the robot should not cross.

To evaluate our method in the real-world, we train a Turtlebot 2 on a reach-avoid task (depicted in Figure 4). The robot is placed in a $4\text{m} \times 6\text{m}$ room and is tasked with reaching two alternating target goals. We consider that a safety constraint must be violated if the robot comes within $0.1\text{m}$ of a static obstacle of $(0, 0)$ of radius $0.36$ m or exits the designated training area. To computing the BRT of the Turtlebot 2, we use the following Dubins car model for its dynamics:

$$\dot{x} = v\cos(\theta) + d_1 \quad \dot{y} = v\sin(\theta) + d_2 \quad \dot{\theta} = \omega + d_3 \tag{12}$$

where $d_1, d_2, d_3 \in [-0.05, 0.05]$ are the disturbances are added to account for inaccuracies when estimating the state. To purposefully mispecify the dynamics, we use $\omega \in [-1.1, 1.1]$ when calculating the BRT, but use $\omega \in [-0.7, 0.7]$ when controlling the robot in the real-world. Additionally, we also add a small random delay ($\sim 0.25$s) before sending the control to the robot. In this setting, the model error is greater than the assumed disturbance bound.

To efficiently learn in the real-world, we use DroQ (Hiraoka et al., 2021) to learn the policy. Following Smith et al. (2022), the robot first collects 1000 samples using a uniform policy. Then, after each action is executed by the robot, the policy is then synchronously updated using an update-to-data ratio of 20. The policy is trained using an NVIDIA GeForce RTX GPU 1060 Mobile and is controlled at 10Hz. We note that synchronously training and controlling the robot on minimal hardware is feasible due to using just-in-time compilation from JAX (Bradbury et al., 2018). In total, 15,000 samples were collected to train the policy over 1 hour.

### 7.2. Results Analysis

In Table 1, we compare our method to a least-restrictive safety filter approach defined in eq. (6). To measure how well each approach keeps the Turtlebot 2 safe, we keep a running total of the

| Shielding Method | Initial Safety Threshold | Final Safety Threshold | Total Number of Safety Violations | Success Rate |
|---|---|---|---|---|
| Robust HJ-CBF | 0.1 | 0.61 | 2 | 80.0% |
| Least-Restrictive | 0.1 | 0.1 | 47 | 20.0 % |

Table 1: Comparative evaluation of the total number of safety violations and final success rate for two methods: Robust HJ-CBF (ours), and Least-Restrictive Safety Filter. The success rate is average of 5 episodes from a fixed starting of the final policy after $15,000$ samples were collected. Videos of the experiment and code can be found at https://github.com/sudo-michael/robust-hj-cbf-safe-rl/tree/main

number of safety violations. For both methods, we start with an initial threshold value of $\epsilon = 0.1$. We find that our method is able to adapt to the mispecified dynamics. This leads to a larger safety threshold, making the safe controller act earlier, resulting in zero safety violations during training. In comparison, the least-restrictive shielding approach lead to more safety violations due to the model mismatch while training. This lead to the robot frequently violating the safety constraint by exiting the designated training boundary. Additionally, the robot could not properly explore the environment since the robot needed to be reset frequently. This prevented the robot from learning the correct behaviour of reaching the goal. In comparison a higher safety threshold resulted in the robot navigating safely along the boundary at a safe distance. This behavior allowed for more useful exploration within the environment, leading to higher success rate of reaching the goal.

## 8. Conclusion and Limitations

In conclusion, we present a robust HJ-CBF safety filter for safe learning in the real-world. Our method can be applied to any off-policy RL algorithm. Using only an approximate model of the agent's dynamics, our method automatically adjusts the sensitivity of the monitor. The resulting safe controller is used based on a maximum slack's variable needed to enforce a safety constraint. This results in a more conservative monitor compared to prior work. We demonstrate that the robust HJ-CBF safety filter can adapt to uncertain dynamics in the real-world and obtain minimal safety violations when compared to a least-restrictive approach.

**Limitations:** We note that our approach is empirical and cannot guarantee safety. Safety violations while using the robust HJ-CBF safety filter can still occur when the difference between $\tilde{f}$ and $f$ is large. Additionally, we require the target set to be static and known ahead of time. To enable safe learning solely learning in the real-world, future work include updating either the approximate model $\tilde{f}$ or $V_{\tilde{f}}$ using data collected online to obtain an accurate BRT that reflects the true agent's dynamics.

## Acknowledgments

This work was supported by the Canada CIFAR AI Chairs and NSERC Discovery Grants Programs.

## References

E. Altman. *Constrained Markov Decision Processes*. Chapman and Hall, 1999.

Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pages 3420–3431. IEEE, 2019.

Somil Bansal and Claire Tomlin. Deepreach: A deep learning approach to high-dimensional reachability, 2020.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J. Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances, 2017.

Javier Borquez, Kaustav Chakraborty, Hao Wang, and Somil Bansal. On safety and liveness filtering using hamilton-jacobi reachability analysis. *arXiv preprint arXiv:2312.15347*, 2023.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Minh Bui, George Giovanis, Mo Chen, and Arrvindh Shriraman. Optimizeddp: An efficient, user-friendly library for optimal control and dynamic programming. *arXiv preprint arXiv:2204.05520*, 2025.

Bingqing Chen, Jonathan Francis, Jean Oh, Eric Nyberg, and Sylvia L. Herbert. Safe autonomous racing via approximate reachability on ego-vision, 2021.

Mo Chen, Qizhan Tam, Scott C Livingston, and Marco Pavone. Signal temporal logic meets reachability: Connections and applications. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 581–601. Springer, 2018.

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 Castaneda, Claire J Tomlin, and Koushil Sreenath. Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions. *arXiv preprint arXiv:2004.07584*, 2020.

Jaime F Fisac, Anayo K Akametalu, Melanie N Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2018.

Jaime F Fisac, Neil F Lugovoy, Vicenç Rubies-Royo, Shromona Ghosh, and Claire J Tomlin. Bridging hamilton-jacobi safety analysis and reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8550–8556. IEEE, 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 *2019 International Conference on Robotics and Automation (ICRA)*, pages 8550–8556, 2019b. doi: 10.1109/ICRA.2019.8794107.

Milan Ganai, Zheng Gong, Chenning Yu, Sylvia Herbert, and Sicun Gao. Iterative reachability estimation for safe reinforcement learning. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 69764–69797. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/dca63f2650fe9e88956c1b68440b8ee9-Paper-Conference.pdf.

Milan Ganai, Sicun Gao, and Sylvia Herbert. Hamilton-jacobi reachability in reinforcement learning: A survey. *IEEE Open Journal of Control Systems*, 2024.

Shangding Gu, Guang Chen, Lijun Zhang, Jing Hou, Yingbai Hu, and Alois Knoll. Constrained reinforcement learning for vehicle motion planning with topological reachability analysis. *Robotics*, 11(4):81, 2022a.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, Yaodong Yang, and Alois Knoll. A review of safe reinforcement learning: Methods, theory and applications. *arXiv preprint arXiv:2205.10330*, 2022b.

Chong He, Zheng Gong, Mo Chen, and Sylvia Herbert. Efficient and guaranteed hamilton-jacobi reachability via self-contained subsystem decomposition and admissible control sets. *IEEE Control Systems Letters*, 2023.

Takuya Hiraoka, Takahisa Imagawa, Taisei Hashimoto, Takashi Onishi, and Yoshimasa Tsuruoka. Dropout q-functions for doubly efficient reinforcement learning. *arXiv preprint arXiv:2110.02034*, 2021.

Kai-Chieh Hsu, Haimin Hu, and Jaime F Fisac. The safety filter: A unified view of safety-critical control in autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 7.

Kai-Chieh Hsu*, Vicenç Rubies-Royo*, Claire Tomlin, and Jaime Fisac. Safety and liveness guarantees through reach-avoid reinforcement learning. *Robotics: Science and Systems XVII*, Jul 2021. doi: 10.15607/rss.2021.xvii.077. URL http://dx.doi.org/10.15607/RSS.2021.XVII.077.

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.

David Isele, Alireza Nakhaei, and Kikuo Fujimura. Safe reinforcement learning on autonomous vehicles. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–6. IEEE, 2018.

Jeongho Kim, Jaeuk Shin, and Insoon Yang. Hamilton-jacobi deep q-learning for deterministic continuous-time systems with lipschitz continuous controls. *Journal of Machine Learning Research*, 22(206):1–34, 2021.

Niklas Kochdumper, Hanna Krasowski, Xiao Wang, Stanley Bak, and Matthias Althoff. Provably safe reinforcement learning via action projection using reachability analysis and polynomial zonotopes. *IEEE Open Journal of Control Systems*, 2:79–92, 2023.

Anjian Li and Mo Chen. Guaranteed-safe approximate reachability via state dependency-based decomposition. In *2020 American Control Conference (ACC)*, pages 974–980. IEEE, 2020.

Albert Lin and Somil Bansal. Generating formal safety assurances for high-dimensional reachability. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10525–10531. IEEE, 2023a.

Albert Lin and Somil Bansal. Verification of neural reachable tubes via scenario optimization and conformal prediction. *arXiv preprint arXiv:2312.08604*, 2023b.

Xubo Lyu and Mo Chen. Ttr-based reward for reinforcement learning with implicit model priors. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5484–5489. IEEE, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Kensuke Nakamura, Lasse Peters, and Andrea Bajcsy. Generalizing safety beyond collision-avoidance via latent-space reachability analysis. *arXiv preprint arXiv:2502.00935*, 2025.

Stanley Osher, Ronald Fedkiw, and K Piechor. Level set methods and dynamic implicit surfaces. *Appl. Mech. Rev.*, 57(3):B15–B15, 2004.

Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V Dimarogonas, Stephen Tu, and Nikolai Matni. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3717–3724. IEEE, 2020.

William Saunders, Girish Sastry, Andreas Stuhlmueller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention, 2017.

Mahmoud Selim, Amr Alanwar, Shreyas Kousik, Grace Gao, Marco Pavone, and Karl H Johansson. Safe reinforcement learning using black-box reachability analysis. *IEEE Robotics and Automation Letters*, 7(4):10665–10672, 2022.

Laura Smith, Ilya Kostrikov, and Sergey Levine. A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. *arXiv preprint arXiv:2208.07860*, 2022.

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

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. 2018.

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.

Garrett Thomas, Yuping Luo, and Tengyu Ma. Safe reinforcement learning by imagining the near future, 2022.

Akifumi Wachi, Xun Shen, and Yanan Sui. A survey of constraint formulations in safe reinforcement learning. *arXiv preprint arXiv:2402.02025*, 2024.

Hao Wang, Javier Borquez, and Somil Bansal. Providing safety assurances for systems with unknown dynamics. *IEEE Control Systems Letters*, 2024.

Albert Wilcox, Ashwin Balakrishna, Brijen Thananjeyan, Joseph E Gonzalez, and Ken Goldberg. Ls3: Latent space safe sets for long-horizon visuomotor control of sparse reward iterative tasks. In *Conference on Robot Learning*, pages 959–969. PMLR, 2022.

Dongjie Yu, Haitong Ma, Shengbo Li, and Jianyu Chen. Reachability constrained reinforcement learning. In *International Conference on Machine Learning*, pages 25636–25655. PMLR, 2022a.

Haonan Yu, Wei Xu, and Haichao Zhang. Towards safe reinforcement learning with a safety editor policy. *Advances in Neural Information Processing Systems*, 35:2608–2621, 2022b.