# Learning Robust Policies via Interpretable Hamilton-Jacobi Reachability-Guided Disturbances

**Hanyang Hu**[1], **Xilun Zhang**[2], **Xubo Lyu**[1], **Mo Chen**[1],

[1]Simon Fraser University
[2]Carnegie Mellon University
hha160@sfu.ca, xilunz@andrew.cmu.edu, lvxubo92@gmail.com, mochen@cs.sfu.ca

## Abstract

Deep Reinforcement Learning (RL) has shown remarkable success in robotics with complex and heterogeneous dynamics. However, its vulnerability to unknown disturbances and adversarial attacks remains a significant challenge. In this paper, we propose a robust policy training framework that integrates model-based control principles with adversarial RL training to improve robustness without the need for external black-box adversaries. Our approach introduces a novel Hamilton-Jacobi reachability-guided disturbance for adversarial RL training, where we use interpretable worst-case or near-worst-case disturbances as adversaries against the robust policy. We evaluated its effectiveness across three distinct tasks: a reach-avoid game in both simulation and real-world settings, and a highly dynamic quadrotor stabilization task in simulation. We validate that our learned critic network is consistent with the ground-truth HJ value function, while the policy network shows comparable performance with other learning-based methods.

## Introduction

Deep Reinforcement Learning has emerged as a powerful tool in robotics, particularly within highly dynamic environments(Lyu and Chen 2020; Zhang et al. 2024; Cheng et al. 2024; Wang et al. 2023). However, its trained policies may fail when the simulation to real-world gap is large due to modeling errors and unknown disturbances (Xu et al. 2022; Huang et al. 2023). Consequently, developing a Robust RL (RRL) policy is crucial to prevent catastrophic policy failures during deployment(Molchanov et al. 2019).

To address the issues of model mismatches and unforeseen disturbances, recent advances in RRL focus on maximizing worst-case performance across various uncertainties. One widely recognized approach within RRL is Robust Adversarial Reinforcement Learning (RARL) (Pinto et al. 2017) which mitigates environmental mismatches by treating them as adversarial perturbations. RARL conceptualizes the problem as a two-player zero-sum game, in which the protagonist aims to develop a robust policy across different environments, while the adversary introduces the perturbation policy to challenge the protagonist. However, such learning-based adversarial agents often lack theoretical interpretability and may generate implausible disturbances (Brunke et al. 2022). In this paper, we aim to develop a robust adversarial RL framework that enables more interpretable and verifiable solutions for adversary and protagonist policies by leveraging robust control theory, which also achieves comparable performances with state-of-the-art under various disturbances.



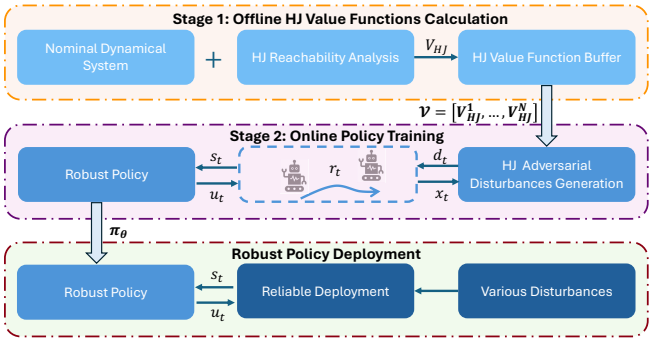


Figure 1: HJARL computes the HJ value functions offline and uses them to generate adversarial disturbances during the online training. The trained robust policy is then deployed to handle various disturbances.

Hamilton-Jacobi (HJ) reachability analysis is a robust optimal control method grounded in game theory. It treats disturbances as adversaries and provides robust minimax optimal policies for both agents, regardless of linear or nonlinear dynamics (Chen and Tomlin 2018). HJ reachability solves a differential game using the Hamilton-Jacobi-Isaacs (HJI) equation. The HJ value function has proven to be the viscosity solution to this HJI equation and can be used to calculate optimal controls and disturbances for opposing parties (Fisac et al. 2015). As a model-based approach, HJ reachability offers physically interpretable protagonist actions and adversarial disturbances. Its robustness is demonstrated by the guaranteed outcomes when initial states lie within a specific region of the value function. However, due to computational limitations, accurate solutions to the HJ reachability become intractable if the state space is higher than six dimensions (6D) (Bui et al. 2022). To mitigate this curse of dimensionality issue, we utilize the nominal dynamics for the HJ computation to circumvent the full high-dimensional

dynamics and focus on low-dimensional tractable solutions instead of using learning-based approximation tools.

In this paper, we propose a novel method HJARL, short for **H**amilton-**J**acobi-guided **A**dversarial **RL** training as shown in Fig.1. Our method differs from other learning-based robust adversarial RL methods, opting to generate adversarial disturbances through HJ reachability directly instead of learning adversarial policy networks. By leveraging nominal dynamical systems, we compute solvable HJ value functions offline in stage 1. This enables the generation of worst-case or near-worst-case disturbances during online training (stage 2), thus improving policy robustness. To smooth the learning process, we utilize the Boltzmann distribution to gradually increase the strength of the adversary to avoid overly strong adversaries in the initial training phase. The model-based nature of HJ reachability guarantees that the generated adversarial actions stay within a physically feasible bound. HJARL obtains a critic network that aligns with the HJ value function, providing an approximation to the HJ guaranteed reachable region, and achieves comparable task performances with other learning-based methods. It also requires less parameter tuning and achieves faster convergence than training adversarial policy networks. Our contributions are as follows:

1. Propose a novel method to obtain robust control policies that integrates the adversarial RL framework with HJ reachability using nominal dynamical systems.
2. Demonstrate the robustness and the consistency to the HJ reachability analysis on the low-dimensional dynamical system and achieve comparable task performances against other baseline methods on the high-dimensional dynamical system.
3. Validate our robust control policy in a real-world one vs. one reach-avoid game using TurtleBot3 robots.

## Related Work

**Robust Adversarial RL.** Motivated by $H_\infty$ control, pioneering works by (Morimoto and Doya 2005; Pinto et al. 2017; Oikarinen et al. 2021) introduced the concept of RARL which trains robust control policies against learned adversarial disturbances within the game theory framework. They formulate zero-sum Markov games as a Markov Decision Process (MDP)(Perolat et al. 2015; Puterman 1990), and generally learn a universal function approximator to approximate adversarial policies. A range of works have been proposed to address the limitations of the RARL framework. Pan et al. proposed a risk-averse RARL algorithm with a risk-averse protagonist and risk-seeking adversary to account for the probability of catastrophic events. Zhang, Hu, and Basar demonstrated that RARL cannot guarantee training stability or convergence, even in linear quadratic cases. Reddi et al. proposed an entropy-regularization-based RARL method to simplify the saddle point optimization of the original algorithm. To address the problem that employing a single adversarial network could lead to biased adversary generation, Vinitsky et al. advocated the use of multiple adversarial networks along with a performance ranking system to ensure more effective training of robust policies. Rather than formulating adversarial training as a zero-sum game, Huang et al. models RRL training as a Stackelberg game, where the adversarial agent is adaptively regularized to improve the stability of the training. Although these methods have shown promising results under the RL framework, they lack a physically interpretable analysis of the adversary, and the trade-off between training convergence and robustness still remains challenging.

**Hamilton-Jacobi Reachability.** $H_\infty$-control theory was first proposed for the design of the worst-case disturbance controller considering the linear dynamical system in the frequency domain (Zames 1981; Francis 1987; Aliyu 2011). Then, Doyle et al.; van der Schaft; van der Schaft extended the $H_\infty$-control theory to nonlinear dynamical systems in the state-space domain, and this resulting minimax optimization problem naturally connects to game theory and differential game (Başar and Olsder 1998; Başar and Bernhard 2008). HJ reachability analysis is a model-based robust optimal control method under the framework of differential games (Chen and Tomlin 2018). (Gong and Herbert 2024; Yang and Bhounsule 2024; Wang, Borquez, and Bansal 2024) provide robust control frameworks grounded in the HJ value function and the corresponding Backward Reachable Tubes (BRT). Due to the curse of dimensionality, numerical solutions to HJ reachability are limited to 6D problems using the latest toolbox (Bui et al. 2022). To circumvent this constraint, Chen, Herbert, and Tomlin introduced a dimensionality reduction technique. In addition to these numerical computation methods, researchers have used neural networks to approximate the HJ value function. Kai-Chieh et al. utilized a time-discounted Safety Bellman Equation with adversarial disturbances and built a new scheme to provide certified safe actions (Fisac et al. 2019; Hsu, Nguyen, and Fisac 2023). Somil et al. proposed the DeepReach method that uses sinusoidal networks as a partial differential equation solver to approximate the HJ value function after long offline training (Bansal and Tomlin 2021). Although these learning-based approximations can be used for higher-dimensional problems, their correctness and accuracy are hard to verify.

Unlike previous work on adversarial training, HJARL provides a physically theoretical bound for the disturbance, generating the worst or near-worst adversarial disturbances incrementally during training. This approach also provides an interpretable viewpoint on the robustness of the learned critic network compared to previous methods while achieving comparable task performances.

## Preliminaries

### Hamilton-Jacobi Reachability Analysis

HJ reachability analysis constitutes a model-based approach to robust optimal control. The following ordinary differential equations govern the **nominal dynamical system**:

$$\dot{x} = f(x, u, d), \quad x(0) = x_0 \tag{1}$$

where $x \in \mathbb{R}^n$ is the state we use for the HJ value function computation, $f$ represents the nominal dynamical system, which may either encompass the complete dynamics or only the partial dynamics relevant to the states of interest,

particularly in cases where computational intractability is a concern, $u \in \mathcal{U}$ is the control input, $d \in \mathcal{D}$ is the disturbance to the system, and $x_0$ is the initial state. The sets $\mathcal{U}$ and $\mathcal{D}$ are the sets of measurable functions.

HJ reachability analysis resides at the intersection of optimal control and differential games, enabling adversarial parties in a minimax game to achieve their optimal control objectives respectively (Chen and Tomlin 2018). In our setting, we refer to the disturbance of the control system as the adversary in a general differential game. The goal of the control input is to push the system to the target set $R$. The target set $R$ can then be defined by an implicit surface function $l(x) : \mathbb{R}^n \to \mathbb{R}$ such that $R = \{x \in \mathbb{R}^n \mid l(x) \leq 0\}$ using the level set method. Similarly, the avoid set $A$ that stands for the constraints can also be formulated by the function $g(x) : \mathbb{R}^n \to \mathbb{R}$ such that $A = \{x \in \mathbb{R}^n \mid g(x) > 0\}$. Given these expressions, let the value function $V_{HJ} : \mathbb{R}^n \times [-T, 0] \to \mathbb{R}$ be the viscosity solution to the HJI partial differential equation within time $[-T, 0]$ (Crandall and Lions 1983):

$$\begin{aligned} \max\{ \min \left\{ \frac{\partial V_{HJ}}{\partial t} + H\left(x, p\right), l(x) - V_{HJ}(x, t) \right\}, \\ g(x) - V_{HJ}(x, t)\} = 0, \quad t \in [-T, 0] \end{aligned} \tag{2}$$

where the optimal Hamiltonian $H$ is calculated as:

$$H(x, p) = \min_{u \in \mathcal{U}} \max_{d \in \mathcal{D}} p^T f(x, u, d)$$

where $p = \frac{\partial V_{HJ}}{\partial x}$. If the avoid set $A$ does not exist, the $g(x)$ can be set to negative infinite. With the solved value function $V_{HJ}$, one can construct the BRT where the value of the state is negative. Starting from the BRT, the agent will reach the target set under optimal control regardless of the disturbance within the time interval.

Since there are no general analytical solutions for Eq.(2) mostly, one must rely on numerical solutions. These methods hinge on discretizing the state space and employing dynamic programming iterations. Consequently, computational and spatial complexity grows exponentially with the expansion of state quantities. Numerical computation tools like the OptimizedDP (Bui et al. 2022) leverage contemporary computational capabilities to effectively tackle Eq. (2) with grid resolutions of reasonable quality, extending to six dimensions.

If we take the time interval $T \to \infty$, we will obtain control policies that are independent of the time $t$. Then, the optimal control and the worst disturbance can be calculated as (Huang et al. 2011):

$$\begin{aligned} u^*(x) &= \arg\min_{u \in \mathcal{U}} \max_{d \in \mathcal{D}} p(x)^\top f(x, u, d) \\ d^*(x) &= \arg\max_{d \in \mathcal{D}} p(x)^\top f\left(x, u^*, d\right) \end{aligned} \tag{3}$$

In this way, as for the nominal dynamics Eq.(1), given the pre-computed value function and the agent's current state, the optimal adversarial control working as the worst disturbance to the protagonist is obtained by Eq. (3).

### Robust Adversarial RL

The adversarial RL aims to learn a policy that maximizes the expected rewards under the worst disturbances. The objective function under the disturbance is as follows :

$$\max_{\theta} \min_{\phi} \mathbb{E}\left[ \sum_{t=0}^{T} \gamma^t r\left(s_t, u_t, d_t\right) \mid \pi_\theta, \pi_\phi \right] \tag{4}$$

where $\pi_\theta$ is the goal robust policy parameterized by $\theta$, $u_t \sim \pi_\theta(s_t)$ is the action sampled from the policy given the discrete state $s_t$ at the time step $t$, $\pi_\phi$ is the learnable adversary policy parameterized by $\phi$, and $d_t \sim \pi_\phi(s_t)$ is the disturbance sampled from this adversarial policy.

## Methodology

### Problem Formulation

**Markov Decision Process.** We consider a MDP defined by the 6-tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma, \mu_0)$, where $\mathcal{S}$ is the set of states, and $\mathcal{A}$ is the set of actions, which can be continuous or discrete. The function $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ represents the transition probability, where $\Delta(\mathcal{S})$ denotes the distribution over the state space $\mathcal{S}$. The reward function is defined as $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, $\gamma \in [0, 1)$ is the discount factor and $\mu_0 : \mathcal{S} \to \mathbb{R}_+$ represents the initial state distribution. The objective in RL is to find a policy $\pi_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, parameterized by $\theta$, that maximizes the expected return $\mathbb{E}_{\tau \sim \pi_\theta}\left[R(\tau)\right]$, where the return $R(\tau)$ is defined as:$R(\tau) := \sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)$, and $\tau$ denotes the trajectories sampled using the policy $\pi_\theta$.

### HJ Reachability-Guided Adversarial Training

HJARL adds adversarial disturbances generated by the HJ value function $V_{HJ}$ directly to the agent's actions. To ensure the generalization of the learned robust policy, $V_{HJ}$ is calculated through a sequence of increasing upper bounds of disturbance. We denote nominal dynamical systems as $[f^1, \ldots, f^i, \ldots, f^N]$ where the upper bound of the disturbances increases uniformly. These nominal dynamical systems then construct a list of Hamiltonians $[H^1(x, p^1), \ldots, H^i(x, p^i), \ldots, H^N(x, p^N)]$, and finally obtain a HJ value function buffer: $\mathcal{V} = [V^1_{HJ}, \ldots, V^i_{HJ}, \ldots, V^N_{HJ}]$. The upper bound of the disturbance is selected based on physically feasible significance and keeps the same order of magnitude as the allowed control input. Given the certain value function $V^i_{HJ}$, the current action of the protagonist $u_t$, and the current state of the agent $s_t$, we can compute the disturbance $d_t$ from the corresponding deterministic disturbance-generated policy $d^i$ based on the $f^i$:

$$d_t = d^i(s_t | V^i_{HJ}) = \arg\max \left( \frac{\partial V^i_{HJ}}{\partial x} \right)^\top \Bigg|_{x = x_t} f^i\left(x_t, u_t, d_t\right) \tag{5}$$

where $x_t$ is the nominal state which is the same or a part of the full state $s_t$ at the time step $t$.

HJARL aims to learn a policy that maximizes the expected rewards in the presence of disturbances generated by Eq.(5). The resulting optimization problem under these disturbances is formulated as follows:

$$\max_{\pi_\theta} \mathbb{E}_{i \sim B(1, N)} \left[ \sum_{t=0}^{T} \gamma^t r\left(s_t, u_t, d_t\right) \mid \pi_\theta, d^i \right] \tag{6}$$

Algorithm 1: HJ Reachability Guided Adversarial Training

1: **Value Function Generation with HJ Reachability**
2: Initialize the value function buffer $\mathcal{V}$, the number of the disturbance levels $N$, the nominal dynamical system $f$;
3: **for** $i = 1, \ldots, N$ **do**
4: solve for $V^i_{HJ}$ based on $f^i$ according to Eq.(2);
5: collect the value function $\mathcal{V} \leftarrow \mathcal{V} \cup V^i_{HJ}$;
**end for**
6: **Adversarial RL Training**
7: Initialize $\theta$; Environment $\mathcal{E}$
8: **while** not converged **do**
9: $\tau_{\text{traj}} = \{\}$
10: **for** rollout $j = 1, \ldots, M$ **do**
11: sample $i \sim B(1, N)$, construct $d^i$ based on $V^i_{HJ}$;
12: $\tau_j \leftarrow$ run policy $\pi_\theta$ and $d^i$ until termination;
13: collect trajectory: $\tau_{\text{traj}} \leftarrow \tau_{\text{traj}} \cup \tau_j$;
**end for**
14: Update $\pi_\theta \leftarrow$ PPO (Schulman et al. 2017)$(\tau_{\text{traj}}, \pi_\theta)$
**end while**

where $\pi_\theta$ is the task policy parameterized by $\theta$, $u_t \sim \pi_\theta(s_t)$ is the action sampled from the policy. The Boltzmann distribution $i \sim B(1, N)$ is used to obtain a smooth curriculum learning process by sampling a value function $V^i_{HJ}$ from the HJ value function buffer $\mathcal{V}$ for each episode.

Algorithm 1 outlines the HJARL in detail. In the first stage (line 1 to line 5), the algorithm precomputes value functions that account for varying levels of disturbances in the environment. Specifically, we define the disturbance levels and calculate the corresponding value function $V_i$ for the nominal dynamical system $f$. The calculation is performed with the OptimizedDP numerical toolbox (Bui et al. 2022). Upon completion of these computations, the resulting value functions are aggregated into a value function buffer $\mathcal{V}$. In the second stage (line 6 to the end), we perform online adversarial RL training, intending to train a policy $\pi_\theta$ that can effectively handle disturbances characterized by the value functions generated in the first stage. During each rollout, we sample a value function $V^i_{HJ}$ from the value function buffer $\mathcal{V}$ at the beginning of each trajectory $\tau_j$; the corresponding disturbances $d^i$ are generated based on the sampled value function $V^i_{HJ}$ and are applied throughout the trajectory. After collecting $M$ trajectories, we choose Proximal Policy Optimization (PPO) (Schulman et al. 2017) to update our policy. This iterative update process continues until the policy converges.

## Numerical Simulations

We evaluated HJARL in two simulated tasks: a one-vs-one reach-avoid game and quadrotor stabilization. The former involves a joint 4D dynamical system, which the OptimizedDP (Bui et al. 2022) can solve in the full state space with sufficient precision for the robust policy. The latter task involves a 12D quadrotor system that is computationally intractable for numerical HJ reachability solvers, where we focus on task-specific states to maintain computational accuracy. We demonstrate the consistency of the robust policy obtained from HJARL by comparing it with the HJ value function in the reach-avoid game, and it delivers a comparable performance to other robust adversarial RL baseline methods.

### One vs. One Reach-Avoid Game

In the one vs. one reach-avoid game, the defender aims to capture the attacker while the attacker seeks to arrive at the destination without being captured. We follow a similar game pattern using the single integrator (SIG) dynamics to (Hu, Bui, and Chen 2023) but without obstacles. The SIG dynamics are as follows:

$$\begin{aligned} \dot{x}_A(t) &= v_A u(t), \quad x_A(0) = x_{A0} \\ \dot{x}_D(t) &= v_D d(t), \quad x_D(0) = x_{D0} \end{aligned} \tag{7}$$

where $x_A$ and $x_D$ are the 2D states that represent the positions of the attacker and the defender respectively, $x_{A0}$ and $x_{D0}$ are their initial positions, $v_A$ and $v_D$ are the constant speed of the attacker and the defender, in this game, we set $v_A = 1.0$ and $v_D = 1.5$ respectively, $u(t)$ and $d(t)$ are the control inputs of the attacker and the defender respectively at the time $t$. The target set $R^{11}$ and the avoid set $A^{11}$ for this one vs. one reach-avoid game are defined as follows:

$$\begin{aligned} R^{11} &= \{x_A \in \mathcal{T}\} \cap \{\|x_A - x_D\|_2 > r\} \\ A^{11} &= \{\|x_A - x_D\|_2 \leq r\} \end{aligned} \tag{8}$$

where $\mathcal{T}$ is the destination to the attacker, and $r$ is the capture radius of the defender. Given these two sets and the dynamics defined in Eq.(7), we can solve Eq. 2 and obtain a 4D BRT $\mathcal{RA}^{11}_\infty(R^{11}, A^{11})$. The attacker will win if the initial joint state $(x_{A0}, x_{D0})$ lies within the $\mathcal{RA}^{11}_\infty(R^{11}, A^{11})$ using the optimal control (Chen, Zhou, and Tomlin 2014):

$$u^*(x_A, x_D, t) = -v_A \frac{p_a(x_A, x_D, -t)}{\|p_a(x_A, x_D, -t)\|_2} \tag{9}$$

where $p_a = \frac{\partial V_{HJ}}{\partial x_A}$ is the partial derivative of the one vs. one HJ value function to the attacker. When the initial joint state $(x_{A0}, x_{D0})$ lies out of the $\mathcal{RA}^{11}_\infty(R^{11}, A^{11})$, the defender will guarantee to capture the attacker following the optimal control(Chen, Zhou, and Tomlin 2014):

$$d^*(x_A, x_D, t) = v_D \frac{p_d(x_A, x_D, -t)}{\|p_d(x_A, x_D, -t)\|_2} \tag{10}$$

where $p_d = \frac{\partial V_{HJ}}{\partial x_D}$ is the partial derivative of the one vs. one HJ value function to the defender.

HJARL considers the optimal attacker policy from Eq.(10) as an adversarial disturbance and aims to obtain a robust policy for the defender. In the adversary generation phase, we set the attacker and defender with the same control input range, $N = 1$. To avoid the problem that the gradient of the one vs. one HJ value function is almost zero when the attacker lies outside of $\mathcal{RA}^{11}_\infty(R^{11}, A^{11})$ causing the attacker not to move, we also compute a one vs. zero HJ value function where only one attacker is in the game so that the attacker will continue moving to the destination

when it lies outside of the $\mathcal{RA}_\infty^{11}(R^{11}, A^{11})$. Then, in the adversarial RL training stage, we collect a total of $10^7$ steps to train the defender policy. The reward function consists of three components: a bonus of 200 is awarded if the defender captures the attacker; a penalty of 200 is applied if the attacker reaches the destination; and the relative distance between the two is subtracted as an additional penalty. As for RARL(Pinto et al. 2017), and RAP(Vinitsky et al. 2020), the control policy of the attacker is represented by adversarial neural networks. Finally, we compare the learned robust controllers trained through HJARL with RARL(Pinto et al. 2017) and RAP(Vinitsky et al. 2020).

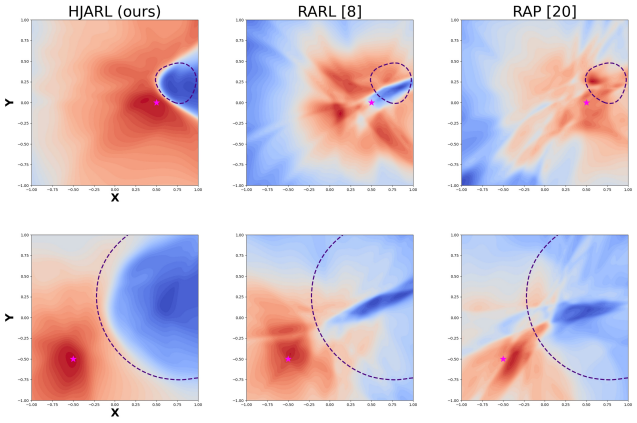


Figure 2: Trained critic networks heatmaps and the zero-level $\mathcal{RA}_\infty^{11}(R^{11}, A^{11})$ (purple dash lines) with SIG dynamics. The first and the second rows show the values of the defender's initial positions at $[0.5, 0.0]$ and $[-0.5, -0.5]$ respectively (magenta stars).

**Consistency to the HJ Value Function.** As illustrated in Fig.2, HJARL demonstrates the strong consistency with the HJ value function through highly overlapping regions. Given that PPO is an actor-critic algorithm, we generate the heatmaps using the trained critic network in HJARL alongside the zero-level $\mathcal{RA}_\infty^{11}(R^{11}, A^{11})$ obtained from the HJ value function with two initial defender positions. In particular, regions with low values on the heat maps correspond closely to the regions enclosed by the zero-level $\mathcal{RA}_\infty^{11}(R^{11}, A^{11})$. When the attacker lies outside the zero-level $\mathcal{RA}_\infty^{11}(R^{11}, A^{11})$, the defender is guaranteed to capture the attacker under the optimal HJ control, regardless of the attacker's control policy. Hence, with this consistency, the trained critic network of HJARL can be used as an approximation to the HJ value function and work as a rough guaranteed attacker-winning region. In addition, this trained critic network can help to check the degree of convergence of the training policy. In contrast, although RARL and RAP also leverage Nash Equilibrium similar to the HJ reachability analysis, the dynamic nature of their adversaries results in an evolving MDP. This variability leads to discrepancies in their value functions, deviating from the consistent behavior exhibited by the true $\mathcal{RA}_\infty^{11}(R^{11}, A^{11})$.

**Comparable Performances.** As depicted in Fig.3, HJARL achieves comparable performances to the RAP method and outperforms the RARL method. We evaluated the performance of the trained policy networks in a batch of games. With the fixed initial defender position, we traverse the initial attacker positions across the entire map uniformly. If the defender successfully captures the attacker, the corresponding initial attacker's position is marked in peach; and if the attacker reaches the destination, its initial position is marked in blue. The capture performances indicate that both HJARL and RAP yield capture outcomes similar to those predicted by HJ reachability. In contrast, RARL fails to capture the attacker in certain areas where the defender should prevail. In addition, the capture performances match the heatmaps generated by the critic networks in HJARL, further strengthening the consistency of HJARL with the optimal HJ reachability policy.

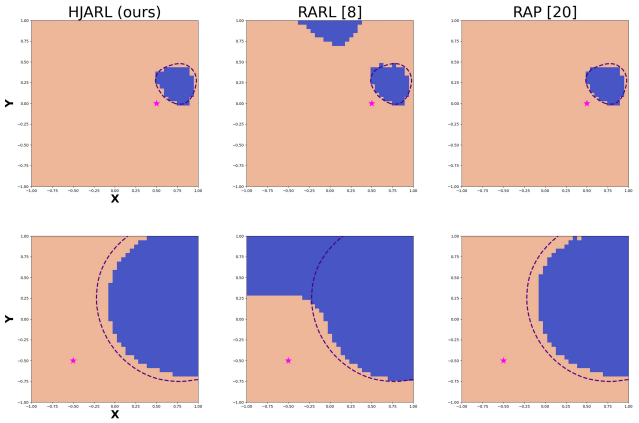


Figure 3: Trained policy game performances and the zero-level $\mathcal{RA}_\infty^{11}(R^{11}, A^{11})$ (purple dash lines) with SIG dynamics. Initial attacker positions are uniformly generated across the map at intervals of 0.05 grid units, with the defender's initial position fixed. The first and the second rows show the game results of the defender's initial positions at $[0.5, 0.0]$ and $[-0.5, -0.5]$ respectively (magenta stars).

## Quadrotor Stabilization

The quadrotor is a representative example of a high-dimensional nonlinear system. Sabatino models a quadrotor as a 12D dynamical system (Sabatino 2015). However, due to the curse of dimensionality, we only focus on the main task-specific six of these dimensions, which we use to define a 6D nominal dynamical system for HJ reachability analysis. The nominal system is defined as follows:

$$\begin{aligned}
\dot{\phi} &= p + r\left(\cos\phi \tfrac{\sin\theta}{\cos\theta}\right) + q\left(\sin\phi \tfrac{\sin\theta}{\cos\theta}\right) \\
\dot{\theta} &= q\left(\cos\phi\right) - r\left(\sin\phi\right) \\
\dot{\psi} &= r\tfrac{\cos\phi}{\cos\theta} + q\tfrac{\sin\phi}{\cos\theta} \\
\dot{p} &= \tfrac{I_y - I_z}{I_x} rq + \tfrac{u_x + d_x}{I_x} \\
\dot{q} &= \tfrac{I_z - I_x}{I_y} pr + \tfrac{u_y + d_y}{I_y} \\
\dot{r} &= \tfrac{I_x - I_y}{I_z} pq + \tfrac{u_z + d_z}{I_z}
\end{aligned} \tag{11}$$

where $\phi, \theta, \psi$ are Euler angles (roll, pitch, and yaw angles respectively) in the earth frame as shown in (Sabatino 2015), $p, q, r$ are angular velocities (roll rate, pitch rate, and yaw rate respectively) in the body frame, $u = [u_x, u_y, u_z]$ are control input torques generated by the differences among motors' speeds, $d = [d_x, d_y, d_z]$ are disturbance torques generated by disturbances like the wind.

External factors such as wind and the discrepancy between the true dynamics and the nominal system are modeled as disturbances. We assume that these disturbances can be applied to the quadrotor as actions. At the initial stage of our algorithm, we increase the upper bound of the disturbances to twice that of the control input's upper bound and compute the value function $V^i_{HJ}$ at uniform intervals of 0.1, resulting in $N = 21$ different value functions. In the second stage, we collect a total of $10^7$ steps to train the robust policy. The reward function comprises four components: a penalty proportional to the clipped action, scaled by $10^{-4}$; a penalty based on angular velocities, scaled by $10^{-3}$; a penalty of 100 for a quadrotor crash; and a penalty proportional to the distance between the current position and the destination. All baseline methods are trained with the same $10^7$ steps.

**Performance Analysis.** The results in TABLE 1 show that HJARL delivers strong performance across all evaluation conditions, comparable to other robust adversarial RL methods. We evaluated the performance of the algorithms in three environments characterized by different types of disturbances applied directly to the first two dimensions of the control actions: random HJ disturbance, random disturbance, and constant disturbance. Random HJ disturbance is generated by selecting a $V^i_{HJ}$ randomly from the value function buffer $\mathcal{V}$; random disturbances are sampled randomly at every step; constant disturbances remain fixed throughout the entire episode. For each seed, we conducted 10 episodes (with a maximum of 1000 steps per episode), totaling 30 episodes per environment. To ensure a robust evaluation, we average the number of steps between three different seeds. The high standard deviation is attributed to the fragility of the quadrotor stabilization process, as it is prone to crashing if the control inputs are not properly designed. HJARL achieves comparable performance in these settings compared to other learning baselines; RAP and pure PPO perform well in random and constant disturbance environments; while RARL performs poorly. The inferior performance of RARL may be due to the adversary overfitting to the training distribution or getting stuck in local minima.

## Real-World Experiments

We test HJARL in a real-world one vs. one reach-avoid game with two TurtleBot3 Burger robots. The dynamics of the TurtleBot3 Burger is a 3D DubinCar model (He et al. 2023). The game is carried out in a square arena $2m \times 2m$ with a square destination shown in Fig. 4. We implement two different control policies for the attacker: HJ control and manual control. As for the HJ control, the full 6D HJ value function with all-time slices requires approximately 120 gigabytes and thus is impractical to implement. Instead, we use a 3D HJ value function from a one vs. zero reach game for the attacker to generate control inputs to drive the attacker toward the destination. Regarding the defender policy, we train the defender's control over $10^7$ steps using HJARL where the attacker uses the HJ control.

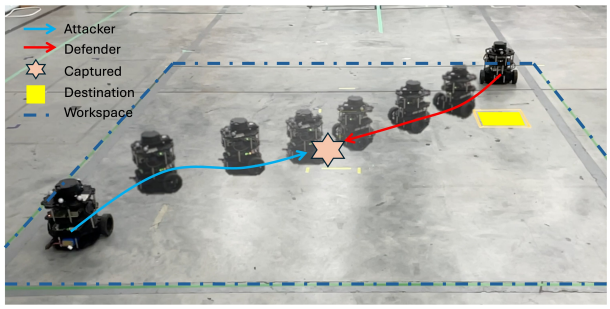


Figure 4: The real-world one vs. one reach-avoid game with two TurtleBot3 Burger robots.

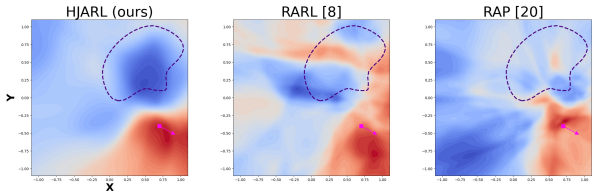


Figure 5: Trained critic networks heatmaps and the zero-level HJ BRT (purple dash lines) with DubinCar model. The initial defender is at $[0.7, -0.4, -0.5]$ with the arrow pointing in its direction (magenta square and arrow).

As illustrated in Fig.5, HJARL still demonstrates the consistency with the HJ BRT, while baseline methods do not. Though the learned critic network does not perfectly overlap with the ground truth value function, it still provides more insights than the black-box learned value networks. As listed in TABLE 2, HJARL achieves the best performance in both attacker control policies. We conducted seven games with the defender and the attacker positioned at different locations on the map. In these scenarios, the relative positions fell outside the HJ BRT, indicating that the defender could capture the attacker with the optimal control policy. HJARL demonstrates superior performance with a HJ-controlled attacker, achieving a capture rate of $85.7\%$ (6/7), outperforming RARL and RAP, which achieved $57.1\%$ and $71.4\%$, respectively. Similarly, HJARL maintains a leading position with a manually-controlled attacker, reaching a $57.1\%$ capture rate, compared to $28.6\%$ for RARL and $42.9\%$ for RAP. These results underscore the robustness and efficiency of HJARL in both control settings.

## Conclusion

In this work, we introduce HJARL, a robust RL training framework with interpretable disturbance generation via HJ reachability analysis. Our approach leverages HJ value functions to create an interpretable disturbance generation for a

| | Random HJ | Random | Constant |
|---|---|---|---|
| HJARL (ours) | $\mathbf{702 \pm 454}$ | $901 \pm 295$ | $\mathbf{677 \pm 457}$ |
| PPO (Schulman et al. 2017) | $287 \pm 430$ | $\mathbf{968 \pm 171}$ | $638 \pm 474$ |
| RARL (Pinto et al. 2017) | $253 \pm 408$ | $738 \pm 433$ | $313 \pm 449$ |
| RAP (Vinitsky et al. 2020) | $257 \pm 411$ | $802 \pm 394$ | $675 \pm 458$ |

Table 1: Performances of Quadrotor on Episode Length

| | HJ attacker | Manual attacker |
|---|---|---|
| HJARL (ours) | $\mathbf{6}/7$ (**85.7**%) | $\mathbf{4}/7$ (**57.1**%) |
| RARL (Pinto et al. 2017) | $4/7$ (57.1%) | $2/7$ (28.6%) |
| RAP (Vinitsky et al. 2020) | $5/7$ (71.4%) | $3/7$ (42.9%) |

Table 2: Performances of real-world experiments

robust policy training pipeline, where we evaluated across two simulation environments and one real-world experiment. We show that HJARL achieves robust performances comparable to state-of-the-art methods while retaining an interpretable adversary in terms of disturbance generation and physical explanation. Despite the inherent challenges of scaling model-based methods to high-dimensional systems and ensuring robust guarantees, recent advances in learning-based techniques that approximate high-dimensional HJ reachability value functions could be employed in future work.

## Acknowledgments

This work was supported by the Canada CIFAR AI Chairs and NSERC Discovery Grants Programs.We thank Satvik Garg, Wenxiang He, and Hanjie Liu for their help with real-world experiments.

## References

Aliyu, M. 2011. *Nonlinear $\mathcal{H}_\infty$-control, Hamiltonian systems and Hamilton-Jacobi equations*. CRC.

Bansal, S.; and Tomlin, C. J. 2021. Deepreach: A deep learning approach to high-dimensional reachability. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 1817–1824. IEEE.

Başar, T.; and Bernhard, P. 2008. *H-infinity optimal control and related minimax design problems: a dynamic game approach*. Springer Science & Business Media.

Başar, T.; and Olsder, G. J. 1998. *Dynamic noncooperative game theory*. SIAM.

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

Bui, M.; Giovanis, G.; Chen, M.; and Shriraman, A. 2022. Optimizeddp: An efficient, user-friendly library for optimal control and dynamic programming. *arXiv preprint arXiv:2204.05520*.

Chen, M.; Herbert, S.; and Tomlin, C. J. 2016. Exact and efficient Hamilton-Jacobi-based guaranteed safety analysis via system decomposition. *arXiv preprint arXiv:1609.05248*.

Chen, M.; and Tomlin, C. J. 2018. Hamilton–Jacobi reachability: Some recent theoretical advances and applications in unmanned airspace management. *Annual Review of Control, Robotics, and Autonomous Systems*, 1: 333–358.

Chen, M.; Zhou, Z.; and Tomlin, C. J. 2014. Multiplayer reach-avoid games via low dimensional solutions and maximum matching. In *2014 American control conference*, 1444–1449. IEEE.

Cheng, X.; Shi, K.; Agarwal, A.; and Pathak, D. 2024. Extreme parkour with legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 11443–11450. IEEE.

Crandall, M. G.; and Lions, P.-L. 1983. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 277(1): 1–42.

Doyle, J.; Glover, K.; Khargonekar, P.; and Francis, B. 1988. State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. In *1988 American Control Conference*, 1691–1696. IEEE.

Fisac, J. F.; Chen, M.; Tomlin, C. J.; and Sastry, S. S. 2015. Reach-avoid problems with time-varying dynamics, targets and constraints. In *Proceedings of the 18th international conference on hybrid systems: computation and control*, 11–20.

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

Francis, B. A. 1987. Lecture notes in control and information sciences.

Gong, Z.; and Herbert, S. 2024. Robust Control Lyapunov-Value Functions for Nonlinear Disturbed Systems. *arXiv preprint arXiv:2403.03455*.

He, C.; Gong, Z.; Chen, M.; and Herbert, S. 2023. Efficient and Guaranteed Hamilton-Jacobi Reachability via Self-Contained Subsystem Decomposition and Admissible Control Sets. *IEEE Control Systems Letters*.

Hsu, K.-C.; Nguyen, D. P.; and Fisac, J. F. 2023. Isaacs: Iterative soft adversarial actor-critic for safety. In *Learning for Dynamics and Control Conference*, 90–103. PMLR.

Hu, H.; Bui, M.; and Chen, M. 2023. Multi-agent reach-avoid games: Two attackers versus one defender and mixed

integer programming. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, 7227–7233. IEEE.

Huang, H.; Ding, J.; Zhang, W.; and Tomlin, C. J. 2011. A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag. In *2011 IEEE International Conference on Robotics and Automation*, 1451–1456. IEEE.

Huang, P.; Xu, M.; Fang, F.; and Zhao, D. 2022. Robust Reinforcement Learning as a Stackelberg Game via Adaptively-Regularized Adversarial Training. In *the 31st International Joint Conference on Artificial Intelligence (IJCAI)*. Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence.

Huang, P.; Zhang, X.; Cao, Z.; Liu, S.; Xu, M.; Ding, W.; Francis, J.; Chen, B.; and Zhao, D. 2023. What went wrong? closing the sim-to-real gap via differentiable causal discovery. In *Conference on Robot Learning*, 734–760. PMLR.

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

Molchanov, A.; Chen, T.; Hönig, W.; Preiss, J. A.; Ayanian, N.; and Sukhatme, G. S. 2019. Sim-to-(Multi)-Real: Transfer of Low-Level Robust Control Policies to Multiple Quadrotors. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 59–66.

Morimoto, J.; and Doya, K. 2005. Robust reinforcement learning. *Neural computation*, 17(2): 335–359.

Oikarinen, T.; Zhang, W.; Megretski, A.; Daniel, L.; and Weng, T.-W. 2021. Robust deep reinforcement learning through adversarial loss. *Advances in Neural Information Processing Systems*, 34: 26156–26167.

Pan, X.; Seita, D.; Gao, Y.; and Canny, J. 2019. Risk averse robust adversarial reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, 8522–8528. IEEE.

Perolat, J.; Scherrer, B.; Piot, B.; and Pietquin, O. 2015. Approximate dynamic programming for two-player zero-sum Markov games. In *International Conference on Machine Learning*, 1321–1329. PMLR.

Pinto, L.; Davidson, J.; Sukthankar, R.; and Gupta, A. 2017. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, 2817–2826. PMLR.

Puterman, M. L. 1990. Markov decision processes. *Handbooks in operations research and management science*, 2: 331–434.

Reddi, A.; Tölle, M.; Peters, J.; Chalvatzaki, G.; and D'Eramo, C. 2023. Robust Adversarial Reinforcement Learning via Bounded Rationality Curricula. *arXiv preprint arXiv:2311.01642*.

Sabatino, F. 2015. Quadrotor control: modeling, nonlinear control design, and simulation.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

van der Schaft, A. 1992. $\mathcal{L}_2$-gain analysis of nonlinear systems and nonlinear state feedback $\mathcal{H}_\infty$ control. *IEEE transactions on automatic control*, 37(6): 770–784.

van der Schaft, A. J. 1991. On a state space approach to nonlinear $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control. *Systems & control letters*, 16(1): 1–8.

Vinitsky, E.; Du, Y.; Parvate, K.; Jang, K.; Abbeel, P.; and Bayen, A. 2020. Robust reinforcement learning using adversarial populations. *arXiv preprint arXiv:2008.01825*.

Wang, H.; Borquez, J.; and Bansal, S. 2024. Providing Safety Assurances for Systems with Unknown Dynamics. *arXiv preprint arXiv:2403.05771*.

Wang, Y.; Xu, M.; Shi, G.; and Zhao, D. 2023. Guardians as you fall: Active mode transition for safe falling. *arXiv preprint arXiv:2310.04828*.

Xu, M.; Liu, Z.; Huang, P.; Ding, W.; Cen, Z.; Li, B.; and Zhao, D. 2022. Trustworthy reinforcement learning against intrinsic vulnerabilities: Robustness, safety, and generalizability. *arXiv preprint arXiv:2209.08025*.

Yang, C.-M.; and Bhounsule, P. A. 2024. Robust Control using Control Lyapunov Function and Hamilton-Jacobi Reachability. *arXiv preprint arXiv:2404.05625*.

Zames, G. 1981. Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on automatic control*, 26(2): 301–320.

Zhang, C.; Xiao, W.; He, T.; and Shi, G. 2024. WoCoCo: Learning Whole-Body Humanoid Control with Sequential Contacts. *arXiv preprint arXiv:2406.06005*.

Zhang, K.; Hu, B.; and Basar, T. 2020. On the stability and convergence of robust adversarial reinforcement learning: A case study on linear quadratic systems. *Advances in Neural Information Processing Systems*, 33: 22056–22068.