Extended Abstract

Disclaimer: This project is taken from my research project with my mentee Ali Fuat Sahin from EPFL. The research project officially started this quarter from March 5th and has been submitted to CoRL. This course project serves as an extension with substantial differences from the paper main text and the case studies are my independent work, though they are included in the CoRL manuscript appendix. Due to this close relationship, the method section is extra lengthy to provide detailed context necessary for the readers to follow.

Motivation From autonomous vehicles to aerial robots, robotic systems are increasingly deployed in safety-critical settings. In these domains, failures can arise not only from nominal planning errors, but also from disturbances, model mismatch, and adversarial interactions. This requires control policies and safety certificates that are robust to worst-case disturbances. Hamilton-Jacobi (HJ) reachability provides a principled framework for this purpose: by solving a dynamic game between the controller and an adversarial disturbance, reachability analysis computes the set of states from which safety can be guaranteed Mitchell et al. (2005), or a target can be reached while avoiding failure Margellos and Lygeros (2011).

However, applying reachability analysis to high-dimensional nonlinear systems remains challenging: classical grid-based solvers suffer from the curse of dimensionality, continuous-time neural solvers require accurate spatial value gradients, and reinforcement-learning- (RL)-based approaches often suffer from weak boundary anchoring and non-stationary adversarial policy optimization. In this project, we aim to propose a discrete-time actor-critic reachability framework that leverages the relative advantages of both continuous-time and RL-based methods to solve long-horizon two-player zero-sum games.

Method To address these challenges, we propose a discrete-time, windowed reachability learning framework for computing backward reachable tubes (BRTs) or backward reach-avoid tubes (BRATs). The method propagates safety values backward from the terminal boundary through a temporal curriculum, decomposing the long-horizon problem into sequential local learning stages. Within each temporal window, approximate optimal policies are constructed from the current value estimate through gradient-free bang-bang probing and refined through supervised imitation learning. The value function is then updated using Bellman-Isaacs supervision from both teacher-based one-step targets and student-policy rollout targets. To reduce long-horizon bootstrap drift, each temporal segment is further refined through rollout-anchored boundary correction before serving as the boundary condition for the preceding window. Collectively, this formulation enables scalable reachability analysis and stable long-horizon value propagation for high-dimensional systems. Finally, we leverage a two-phase synthesis-and-control strategy from Thorup et al. (2026) to utilize the learned finite-horizon policies for infinite (or long-horizon) games.

Implementation The code is implemented in python with the aid of Claude. All main functionalities for the proposed method are all manually written before Claude takes care of the argument type checking and refactorization. One of the baseline methods, **?**, is implemented by AI due to the steep learning curve of the toolbox but with careful manual inspection and tuning.

Results The proposed method significantly outperform the continuous time reachability solver Feng et al. (2025), RL-based reachability solver Hsu et al. (2021), and MPPI controllers Williams et al. (2016). Qualitatively, the heatmaps of our learned value function also align with human intuition, confirming good a learning quality.

Discussion and Conclusion The continuous-time solvers suffers from two main drawbacks: 1) the need of explicit gradient learning requires the method to utilize sinusoidal neural network, which is not scalable and unstable during training; 2) using a monolithic network leads to representational overload when learning long-horizon reachability solutions. On the other hand, RL-based reachability directly learns the converged value function, control, and disturbance policy, all of which depends on the others, resulting in an extremely weak training signal due to the moving target issue. Our framework inherits the temporal curriculum and boundary condition anchoring from the continuous-time reachability solvers to mitigate the moving target issue while leveraging the actor-critic framework from RL to unlock more stable neural representations and more stable training. Together, the method demonstrate superior performance in solving long-horizon two-player zero-sum games.

An Actor-Critic Neural Reachability Solver for High-Dimensional Zero-Sum Games

Zeyuan Feng
Department of Aero Astro
Stanford University
zeyuanf@stanford.edu

Abstract

Hamilton-Jacobi (HJ) reachability provides a principled framework for synthesizing safety certificates and robust controllers for safety-critical robotic systems. However, applying reachability analysis to high-dimensional nonlinear systems remains challenging: classical grid-based solvers suffer from the curse of dimensionality, continuous-time neural solvers require accurate spatial value gradients, and reinforcement-learning-based approaches often suffer from weak boundary anchoring and non-stationary adversarial policy optimization. We propose a discrete-time neural reachability framework for control-disturbance-affine systems that learns backward reachable tubes (BRTs) and backward reach-avoid tubes (BRATs) through Bellman-Isaacs value propagation. Our key idea is to combine equation-driven self-supervision with structured policy learning: rather than computing explicit PDE-gradients, we exploit the bang-bang structure of optimal safety interventions to construct approximate teacher actions from gradient-free value probes, converting adversarial actor learning into supervised policy learning. To stabilize long-horizon value propagation, we leverage the learned actor to train the value function backward from the terminal boundary using a windowed temporal curriculum, where each window is used as the boundary condition for the next window. Furthermore, we propose a two-phase offline synthesis strategy and a sliding-window online deployment scheme to approximate infinite-horizon reach-avoid behaviors. We validate our method on a challenging high-dimensional pursuit-evasion task, demonstrating that it learns accurate value functions while significantly improving training stability over existing solvers.

1 Introduction

From autonomous vehicles to aerial robots, robotic systems are increasingly deployed in safety-critical settings. In these domains, failures can arise not only from nominal planning errors, but also from disturbances, model mismatch, and adversarial interactions. This requires control policies and safety certificates that are robust to worst-case disturbances. Hamilton-Jacobi (HJ) reachability provides a principled framework for this purpose: by solving a dynamic game between the controller and an adversarial disturbance, reachability analysis computes the set of states from which safety can be guaranteed Mitchell et al. (2005), or a target can be reached while avoiding failure Margellos and Lygeros (2011).

Despite its strong theoretical foundations, HJ reachability remains difficult to apply to high-dimensional nonlinear robotic systems. Classical grid-based methods Mitchell (2004); Bui et al. (2022) solve the Hamilton-Jacobi-Isaacs (HJI) partial differential equation with strong numerical guarantees, but their computational cost grows exponentially with the state dimension Bansal et al. (2017a). To overcome this curse of dimensionality, recent works have explored learning-based approximations of HJ reachability. One prominent direction uses Physics-Informed Neural Networks

(PINNs) to solve the continuous-time HJI PDE by minimizing its residual Bansal and Tomlin (2021); Feng et al. (2025); Singh et al. (2025); Chilakamarri et al. (2025); Sharpless et al. (2024). The key appeal of this approach is that learning is self-supervised: the PDE itself provides a dense training signal, allowing the value function to be optimized without requiring ground-truth reachable sets. However, evaluating the HJI residual requires accurate spatial value gradients, $\nabla_x V$, throughout the state space. In long-horizon, high-dimensional, or nonsmooth reachability problems, these gradients can be difficult to represent and optimize reliably. While architectures such as SIRENs can improve derivative fidelity Bansal and Tomlin (2021), they often introduce optimization instability and become increasingly difficult to scale in high-capacity settings Raissi et al. (2019); Sitzmann et al. (2020).

A second line of work formulates reachability as a discrete-time zero-sum reinforcement learning (RL) problem Fisac et al. (2019); Hsu et al. (2021, 2023); Li et al. (2025). This perspective is attractive because it naturally handles learned dynamics, complex simulators, and high-dimensional systems without requiring direct access to continuous-time PDE derivatives. However, RL-based reachability methods suffer from a fundamental moving-target problem: value estimates, control policies, and adversarial disturbance policies are all updated simultaneously through mutually dependent supervision signals. As the controller adapts to the current disturbance policy, the disturbance simultaneously adapts to the evolving controller, while both remain coupled to a changing value function estimate. Without a stable boundary-conditioned anchor, these recursive updates can accumulate bootstrap error over long horizons, leading to unstable optimization and inaccurate safety boundary propagation.

To address these challenges, we propose a discrete-time, windowed reachability learning framework for computing backward reachable tubes (BRTs) or backward reach-avoid tubes (BRATs). The method propagates safety values backward from the terminal boundary through a temporal curriculum, decomposing the long-horizon problem into sequential local learning stages. Within each temporal window, approximate optimal policies are constructed from the current value estimate through gradient-free bang-bang probing and refined through supervised imitation learning. The value function is then updated using Bellman-Isaacs supervision from both teacher-based one-step targets and student-policy rollout targets. To reduce long-horizon bootstrap drift, each temporal segment is further refined through rollout-anchored boundary correction before serving as the boundary condition for the preceding window. Collectively, this formulation enables scalable reachability analysis and stable long-horizon value propagation for high-dimensional systems.

Our approach can be viewed as combining the most useful aspects of PINN- and RL-based formulations while avoiding their primary failure modes. Akin to PINN-based methods, we rely on self-supervision from the governing optimality equation: instead of minimizing a continuous-time HJI PDE residual, we minimize the discrete-time Bellman-Isaacs error induced by the dynamic programming recursion. This preserves the physics-informed learning signal from PINN-style methods, but avoids their dependence on accurate spatial value gradients, which are difficult to approximate in high-dimensional, non-smooth problems. At the same time, our formulation inherits the flexibility of discrete-time RL-style methods, which can be applied to complex sampled (black-box) dynamics. However, rather than learning control and disturbance policies through coupled adversarial actor-critic updates, we exploit the bang-bang structure of the optimal policies to convert actor learning into a supervised learning problem. These supervised policies, together with the terminal boundary condition, provide stable anchors for Bellman target construction, making value propagation substantially more stable than unconstrained minimax policy-value optimization.

Crucially, the enhanced stability of this value propagation enables us to tackle highly adversarial environments that frequently cause traditional solvers to fail. While the framework is naturally applicable to robust optimal control problems where the disturbance represents standard external perturbations or model mismatch, we explicitly pivot to solving two-player zero-sum games where the disturbance represents targeted interactions from adversarial agents. By specifically formulating the problem with asymmetric control authority, we introduce severe numerical stiffness that tests the limits of reachability solvers. We demonstrate that our method successfully learns accurate BRT and BRAT value functions in these challenging, high-dimensional adversarial systems, vastly improving training stability over existing learning-based solvers.

2 Problem Formulation

Consider a discrete-time nonlinear system with control-disturbance affine dynamics:

$$x_{k+1} = f(x_k, u_k, d_k) = f_0(x_k) + f_u(x_k)u_k + f_d(x_k)d_k, \tag{1}$$

where $x_k \in \mathcal{X}$, $u_k \in \mathcal{U} = \{u \in \mathbb{R}^{n_u} \mid \underline{u} \leq u \leq \overline{u}\}$, and $d_k \in \mathcal{D} = \{d \in \mathbb{R}^{n_d} \mid \underline{d} \leq d \leq \overline{d}\}$ denote the state, control, and disturbance at time step k, respectively. The disturbance d_k is treated adversarially to capture worst-case safety margins, representing both exogenous environmental disturbances, such as wind gusts or adversarial agents, and internal epistemic uncertainties, such as unmodeled friction. This control-disturbance affine setting remains expressive enough to model a broad class of robotic systems, including autonomous vehicles, drones, and manipulators.

Let the failure set be $\mathcal{L} = \{x \mid \ell(x) \leq 0\}$ and the target set be $\mathcal{G} = \{x \mid g(x) \leq 0\}$. We aim to approximate the solutions of a infinite-horizon reach-avoid problems, characterized by a backward reach-avoid tube (BRAT) along with the optimal control and disturbance policies

$$\mathcal{B}_L^\infty = \left\{ x : \forall d(\cdot), \exists u(\cdot), \exists \kappa \in [0, \infty], x_{x,0}^{u,d}(\kappa) \in \mathcal{G}, \forall s \in [0, \kappa], x_{x,0}^{u,d}(s) \notin \mathcal{L} \right\}, \tag{2}$$

where $x_{x,k}^{u,d}$ denotes the trajectory induced by policies $u(\cdot)$ and $d(\cdot)$ starting from state x at time step k.

Since learning infinite-horizon (or extremely long-horizon) BRATs is computationally prohibitive for temporal-curriculum-based frameworks, we adopt a two-stage training strategy to approximate the infinite-horizon behaviors with finite-horizon solutions. Specifically, we first solve an avoid-only safety problem, characterized by a backward reachable tube (BRT) $\mathcal{B}_S(k)$, to safeguard the control player at all times. We then solve the finite-horizon BRAT under these safety constraints, which consists of states from which the target can be reached while avoiding safety violations at all times. The BRT and BRAT are mathematically defined as:

$$\begin{aligned} \mathcal{B}_S(k) &= \left\{ x : \forall u(\cdot), \exists d(\cdot), \exists \kappa \in [k, K], x_{x,k}^{u,d}(\kappa) \in \mathcal{L} \right\}, \\ \mathcal{B}_L(k) &= \left\{ x : \forall d(\cdot), \exists u(\cdot), \exists \kappa \in [k, K], x_{x,k}^{u,d}(\kappa) \in \mathcal{G}, \forall s \in [k, \kappa], x_{x,k}^{u,d}(s) \notin \mathcal{L} \right\}, \end{aligned} \tag{3}$$

Concretely, we synthesize the relevant tubes by learning their corresponding value functions. In avoid-only tasks, we learn the BRT value function V_S^* and its associated safety-preserving policy, while in reach-avoid tasks, we learn the BRAT value function V_L^* and the corresponding reach-avoid policy. These value functions are given as Bansal et al. (2017b):

$$\begin{aligned} V_S^*(x, k) &= \max_{\mathbf{u}} \min_{\mathbf{d}} \left[\min_{\kappa \in [k, K]} \ell(x_\kappa) \right], \\ V_L^*(x, k) &= \min_{\mathbf{u}} \max_{\mathbf{d}} \left[\min_{\kappa \in [k, K]} \left(\max \left(g(x_\kappa), \max_{s \in [k, \kappa]} -\ell(x_s) \right) \right) \right]. \end{aligned} \tag{4}$$

Under this convention, $\mathcal{B}_S(k) = \{x : V_S^*(x_t, k) \leq 0\}$ and $\mathcal{B}_L(k) = \{x \mid V_L^*(x, k) \leq 0\}$. The optimal value function satisfies the following discrete-time Bellman-Isaacs recursions Hsu et al. (2023):

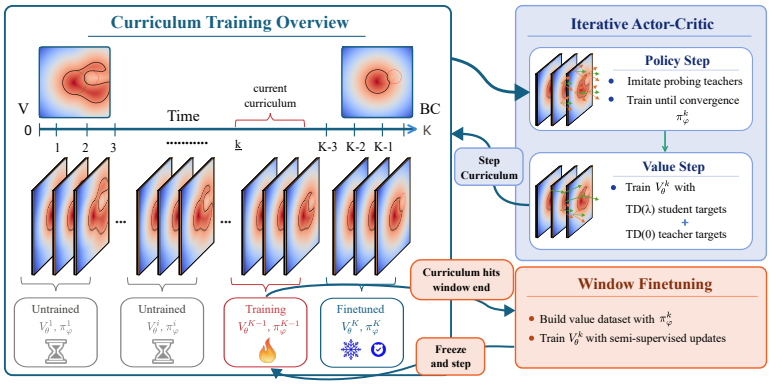
$$\begin{aligned} V_S^*(x, k) &= \min \left(\ell(x), \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} V_S(f(x, u, d), k + 1) \right), \\ V_L(x, k) &= \max \left(-\ell(x), \min \left(g(x), \min_{u \in \mathcal{U}} \max_{d \in \mathcal{D}} V_L(f(x, u, d), k + 1) \right) \right), \end{aligned} \tag{5}$$

with the boundary condition (BC) $b(x)$: $V_S^*(x, K) = \ell(x)$ and $V_L^*(x, K) = \max\big(g(x), -\ell(x)\big)$.

Our objective is to approximate these Bellman-Isaacs solutions with neural value functions and their corresponding optimal policies. Specifically, we aim to learn finite-horizon neural value functions $V_\theta(x, k)$ along with shared-network control and disturbance policies, $\pi_\phi^u(x, k)$ and $\pi_\phi^d(x, k)$, for both the BRT and BRAT. Together, they approximate the solution of infinite-horizon two-player zero-sum games.

3 Method

As stated in the problem formulation, our method first solves the safety problem (BRT) till the value function converged to infinite-time solution, then leverage the BRT to synthesize a safety-constrained



Algorithm 1 Overview

Require: K, g, ℓ, W
  $\underline{k} \leftarrow K$, initialize NNs
  while $\underline{k} \geq 0$ **do**
    $\pi \leftarrow \text{Supervise}(\hat{u}^{\underline{k}:K}_{\text{teacher}})$
    $V \leftarrow \text{Update}(y^{\underline{k}:K}_{\text{mixed}})$
    if $(K - \underline{k}) \bmod W = 0$ **then**
      FinetunePhase
      Freeze current window
      Unfreeze next window
    end if
    $\underline{k} \leftarrow \underline{k} - 1$
  end while

Figure 1: Offline Synthesis Overview: backward temporal curriculum with actor-critic learning.

BRAT. Finally, we leverage a online-deployment solution from Thorup et al. (2026) to deploy the strategy for "infinite horizon" (extremely long-horizon) simulations. Since the BRT and BRAT synthesis are similar, we break down the method section into offline synthesis phase and a brief online deployment subsection.

3.1 Offline Synthesis Overview

The offline synthesis phase learns the neural value function associated with BRT or BRAT. Our key idea is to train backward from the terminal boundary using a windowed temporal curriculum: each window learns a local value function and corresponding control and disturbance policies through approximate teacher supervision and Bellman-error minimization, and then undergoes a finetuning phase for reducing any learning errors before being frozen as the boundary condition for the preceding window.

Temporal Partitioning via Windowed Network Architectures. Approximating long-horizon Bellman-Isaacs solutions with a single monolithic network can lead to representational overload, particularly when the value function exhibits sharp temporal variation or nonsmooth reachable-set boundaries. To reduce this burden, we partition the discrete horizon K into sequential temporal windows, each represented by a specialized network. Let $\mathcal{K} = \{0, 1, \ldots, K\}$. We define a neural value function $V_\theta : \mathcal{X} \times \mathcal{K} \to \mathbb{R}$ together with a shared policy network $\pi_\phi^{u,d} : \mathcal{X} \times \mathcal{K} \to \mathcal{U} \times \mathcal{D}$ that parameterizes both the control and disturbance policies. The value function is boundary-aware by construction at the terminal time $k = K$ and uses independent network weights within each temporal window:

$$V_\theta(x, k) = \begin{cases} b(x), & \text{if } k = K, \\ V_\theta^w(x, k), & \text{if } K - wW \leq k < K - (w-1)W, \end{cases} \tag{6}$$

where $W \in \mathbb{Z}^+$ denotes the window length, $w \in \mathbb{Z}^+$ is the window index counted backward from the terminal time, and $b(x)$ is the terminal boundary condition.

The shared policy parameterization is decomposed into temporal windows: $\pi_\phi^{(\cdot)}(x, k) = \pi_\phi^{(\cdot),w}(x, k)$ for $K - wW \leq k < K - (w-1)W$, where $(\cdot) \in \{u, d\}$ denotes the control and disturbance policies. This windowed representation decomposes the long-horizon dynamic programming problem into shorter-horizon subproblems. Each network focuses its representational capacity on a localized temporal segment, while later-time windows, once trained and frozen, serve as fixed boundary conditions for earlier windows.

As established in the problem formulation, our method first solves the avoid-only safety problem...

Backward Curriculum. As illustrated in Alg. 1 and Fig. 1, training proceeds backward along the discrete time axis, starting from the terminal boundary condition. At each curriculum step, the algorithm alternates between policy training and value training, followed by a window-finetuning phase whenever a temporal window is completed. Each phase is trained to near convergence before the curriculum advances. Crucially, the policy update is performed before the value update to ensure that the updated policies are available to construct accurate temporal-difference (TD) targets

for value learning. Whenever the elapsed curriculum horizon $(K - \underline{k})$ reaches a multiple of the window duration W, the algorithm triggers a finetuning phase to calibrate the value network against rollout-derived targets, effectively aligning the policies and the value function. After fine-tuning, the parameters of the completed window, including both value and policy networks, are frozen and used as the boundary condition for the preceding window. We now detail each of these phases.

3.1.1 Policy Training via Bang-Bang Probing

The policy training phase learns control and disturbance policies that approximate the optimal actions in the Bellman-Isaacs recursion in Eqn. 4. To mitigate the "moving target" problem prevalent in joint control-disturbance optimization (a minimax update), we construct approximate teacher actions from the current value estimate and use them to supervise the student policy networks:

$$\mathcal{L}_\pi = \frac{1}{N}\sum_{i=1}^{N} w_i \left(\left\| \pi_\phi^u(x_i, k_i) - \hat{u}^*(x_i, k_i) \right\|_2^2 + \left\| \pi_\phi^d(x_i, k_i) - \hat{d}^*(x_i, k_i) \right\|_2^2 \right), \tag{7}$$

where $x_i \in \mathcal{X}$ are sampled states, $k_i \in \mathcal{K}_{\text{curr}} = \{\underline{k}, \underline{k}+1, \ldots, K-(w-1)W-1\}$ are time indices, $(\hat{u}^*, \hat{d}^*)$ are the estimated teacher actions, and w_i weights the confidence of each teacher label.

Estimating Teacher Policies. Our teacher construction is inspired by the bang-bang structure of optimal policies in continuous-time Hamilton-Jacobi reachability Bansal et al. (2017b); Mitchell et al. (2007). For continuous-time control-disturbance-affine dynamics $\dot{x} = f_c(x, u, d) = f_{c,0}(x) + f_{c,u}(x)u + f_{c,d}(x)d$, the continuous-time optimal actions are strictly bang-bang:

$$\hat{u}_j^* = \begin{cases} \overline{u}_j, & C_{u,j} > 0, \\ \underline{u}_j, & C_{u,j} \le 0, \end{cases} \qquad \hat{d}_j^* = \begin{cases} \underline{d}_j, & C_{d,j} > 0, \\ \overline{d}_j, & C_{d,j} \le 0, \end{cases} \tag{8}$$

where $C_{u,j}$ and $C_{d,j}$ are the j-th components of the gradient projections $f_{c,u}(x)^\top \nabla_x V$ and $f_{c,d}(x)^\top \nabla_x V$, respectively. For the opposite minimization-maximization convention (e.g., in BRAT), the endpoint assignments are reversed accordingly.

Unlike PINN-based reachability solvers, our method does not require analytic spatial gradients $\nabla_x V_\theta$ for policy extraction. Instead, we infer the relevant gradient-projection signs through discrete value comparisons of forward-simulated next states. For each control dimension j, we evaluate $\Delta V_j^u := V_\theta(x^{\overline{u}_j}, k+1) - V_\theta(x^{\underline{u}_j}, k+1) \approx \Delta t\, \nabla_x V_\theta(x, k+1)^\top f_{c,u}(x)\big(\overline{u}_j - \underline{u}_j\big) = \Delta t\, C_{u,j}\big(\overline{u}_j - \underline{u}_j\big)$, where $x^{\overline{u}_j}$ and $x^{\underline{u}_j}$ denote the next discrete-time states obtained by setting only the j-th control dimension to its upper or lower bound, while holding the remaining action dimensions fixed at a baseline value. Since $\overline{u}_j - \underline{u}_j > 0$, the sign of ΔV_j^u approximates the sign of $C_{u,j}$ for sufficiently small discretization step Δt. An analogous probing procedure yields $\operatorname{sign}(C_{d,j}) \approx \operatorname{sign}(\Delta V_j^d)$ for each disturbance dimension.

This procedure requires only $2(n_u + n_d)$ forward value evaluations per sampled state and provides an efficient, gradient-free approximation of the teacher actions $(\hat{u}^*, \hat{d}^*)$. To reduce the influence of ambiguous labels, such as regions where the value is locally flat or multiple actions yield nearly identical next-step values (e.g., $\nabla V_x \to \mathbf{0}$), we downweight uncertain teachers through the confidence weight w_i, computed as the inverse variance of these directional probe scores.

3.1.2 Value Training via Mixed Temporal-Difference Targets

A fundamental challenge in this decoupled actor-critic scheme is compounding suboptimality: if a suboptimal student policy is used to train the value network, the resulting value estimate may become biased, which in turn yields corrupted teacher actions in subsequent policy updates. To mitigate this error propagation, we train the value function using a mixture of one-step TD targets generated from teacher actions and multi-step TD targets generated from student-policy rollouts. The teacher targets keep the value update anchored to the Bellman-Isaacs recursion to preserve optimality, while the student targets improve consistency with the learned policies over longer rollouts.

For a sampled state-time pair (x_0, k_0), we first compute a one-step teacher target. Let $x_1 = f(x_0, \hat{u}^*, \hat{d}^*)$ be the next state under the teacher actions. The teacher TD target is then $y_{\text{teacher}} =$

$\mathcal{J}\big(x_0, V_\theta(x_1, k_0+1)\big)$, where $\mathcal{J}(x,V) = \min\big(\ell(x), V\big)$ for BRT computations and $\mathcal{J}(x,V) = \max\left(-\ell(x), \min\left(g(x), V\right)\right)$ for BRAT.

We also construct a multi-step student target by rolling out the learned policies (π^u_ϕ, π^d_ϕ) for M steps, where M is sampled from a geometrically decaying distribution. Let $x_{0:M}$ denote the resulting trajectory and let $V_\theta(x_M, k_0+M)$ be the bootstrapped terminal value. The student target is $y_{\text{student}} = \mathcal{J}\big(x_{0:M}, V_\theta(x_M, k_0+M)\big)$. For BRT, this reduces to the minimum safety margin along the rollout, clipped by the bootstrapped value at the terminal rollout state. Similarly, for BRAT, it evaluates the minimum reach-avoid cost along the rollout.

The value network are trained to minimize the MSE residual of V_θ against the evenly mixed targets (i.e., 50% student samples). To improve boundary-condition compliance, we oversample time indices near the current boundary, $k_0 = K-(w-1)W-1$, when drawing $k_0 \in \mathcal{K}_{\text{curr}}$. Importantly, both student and teacher policies are filtered by the converged BRT solution to ensure infinite-horizon safety when generating TD targets. Specifically, we employ a least-restrictive filter Borquez et al. (2024):

$$u_{\text{filtered}} = \begin{cases} u_{\text{nom}}, & \text{if } V_S(x,0) > 0 \\ \pi^u_{\phi,\text{BRT}}(x), & \text{otherwise} \end{cases} \tag{9}$$

where $\pi^u_{\phi,\text{BRT}}(x)$ is the optimal safety policy synthesized during the avoid-only phase. This switching logic prevents myopic behaviors of reaching closer at a cost of slight collisions and ensures that the BRAT value training remains strictly within the safe domain.

3.1.3 Finetune Phase

Although mixed TD targets reduce short-horizon suboptimality, the backward windowed curriculum remains susceptible to bootstrap drift across temporal boundaries. If a completed window is frozen with biased value estimates, those errors become the boundary condition for the preceding window and can propagate backward through the horizon. To reduce this drift, we perform a boundary-correction fine-tuning phase at the end of each temporal window before freezing its parameters.

This phase calibrates the value network against rollout-derived targets under the *learned* student policies. Specifically, we optimize a semi-supervised objective that combines local Bellman consistency with supervised anchor targets obtained from discrete-time dynamics rollouts:

$$\mathcal{L}_{\text{FT}} = \mathcal{L}^{\text{student}}_{\text{TD}} + \lambda_{\text{sup}}\mathcal{L}_{\text{sup}} = \tfrac{1}{B}\left(\sum_{i=1}^{B}\left(V_\theta(x_i,k_i) - y_{\text{student},i}\right)^2 + \lambda_{\text{sup}}\sum_{m=1}^{B} w_m\left(V_\theta(x_m,k_m) - G_m\right)^2\right). \tag{10}$$

Here, $\mathcal{L}^{\text{student}}_{\text{TD}}$ uses only student-policy TD targets, omitting teacher targets so that the update aligns the value network with the policies being evaluated. The second term anchors the value network to empirical rollout targets G_m, sampled from a static dataset. To construct the G_m dataset, we forward-simulate from (x_m, k_m) using the learned student policies until reaching the upper boundary of the active window, $k_{\text{bound}} = K-(w-1)W$, where $w \geq 1$ is the current window index. The anchor target is computed from the same stage-wise reachability cost along this rollout and bootstrapped with the frozen value function at k_{bound}:

$$\begin{aligned} G^{\text{BRT}} &= \min\left[\min_{\kappa\in[k_m,k_{\text{bound}}]}\ell(x_\kappa),\ V_\theta\left(x_{k_{\text{bound}}}, k_{\text{bound}}\right)\right]. \\ G^{\text{BRAT}} &= \min\left[\min_{\kappa\in[k_m,k_{\text{bound}}]}\left(\max\left(g(x_\kappa), \max_{s\in[k_m,\kappa]} -\ell(x_s)\right)\right),\ V_\theta\left(x_{k_{\text{bound}}}, k_{\text{bound}}\right)\right]. \end{aligned} \tag{11}$$

The first term evaluates whether the target is reached safely within the active window, while the second term bootstraps from the frozen boundary value. To discourage false-positive safety predictions, we apply asymmetric weights $w_m = 1 + \lambda_{\text{fp}}\mathbb{1}\{V_\theta(x_m,k_m) > 0 \text{ and } G_m < 0\}$, where λ_{fp} controls the penalty for states predicted to be safe but evaluated as unsafe by rollout. The fine-tuning phase is run with a reduced learning rate until convergence. The active window is then frozen and used as the boundary condition for the preceding window.

3.2 Online Deployment

To execute the synthesized strategies online for infinite-horizon tasks, we adapt the temporal-window deployment strategy proposed by Thorup et al. (2026). Rather than relying on a single time index, we evaluate the policies over a sliding window of duration K_{OL} to better coupe with learning errors.

At runtime, if the target is deemed currently unreachable from the state (i.e., the BRAT value function $V_L(x, 0) > 0$), we move the temporal window to the maximal learning horizon, spanning $[0, K_{\text{OL}}]$. Conversely, if the target is reachable, we center the sliding window around the shortest time-to-reach. Let K_{mid} denote the largest time index (corresponding to the smallest remaining time) such that the goal is reachable:

$$K_{\text{mid}} = \arg\max_k \{k \mid V_L(x, k) \leq 0\}. \tag{12}$$

The active temporal window is then dynamically defined as $\left[K_{\text{mid}} - \frac{1}{2}K_{\text{OL}}, K_{\text{mid}} + \frac{1}{2}K_{\text{OL}}\right]$.

At each discrete execution step, we query the optimal reach-avoid policy $\pi^u_{\phi,\text{BRAT}}(x, k)$ across all time indices within this active window. We then aggregate these proposed bang-bang actions and select the most-voted one. This majority-voted action serves as our nominal task-driven control, u_{nom}, which is subsequently passed through the least-restrictive safety filter (9) to ensure safety during deployment.

4 Experimental Results

We evaluate our framework on a challenging 10D long-horizon reach-avoid pursue-evade problem. featuring one evader (modeled as a 4D Dubins car) and two cooperative pursuers (modeled as 3D Dubins cars). The game takes place in a bounded 10×10 field containing a central circular obstacle with a radius of 1.

The 10D state vector concatenates the states of all three agents: $x = [x_e, y_e, \theta_e, v_e, x_{p_1}, y_{p_1}, \theta_{p_1}, x_{p_2}, y_{p_2}, \theta_{p_2}]^\top$. The spatial coordinates for all agents are bounded by the field size $x, y \in [-5.0, 5.0]$, headings by $\theta \in [-\pi, \pi]$, and the evader's velocity by $v_e \in [0.0, 2.0]$. The two pursuers act as the joint control player (minimizer) aiming to capture the evader, with control inputs $u = [\omega_{p_1}, \omega_{p_2}]^\top$ bounded by $\omega_{p_i} \in [-1.0, 1.0]$. The evader acts as the disturbance player aiming to escape, with inputs $d = [\omega_e, a_e]^\top$ bounded by $\omega_e \in [-1.2, 1.2]$ and $a_e \in [-2.0, 2.0]$. Both pursuers move at a constant linear velocity $v_{\text{const}} = 1.0$.

Table 1: Quantitative catching rates for the 10D pursuit-evasion game. We evaluate the performance of our method against the MPPI baseline by pairing different combinations of pursuer and evader policies.

		Evader Policy			
		MPPI	**MADR**	**RARL**	**Ours**
Pursuer Policy	**MPPI**	40%	48%	64%	14%
	MADR	12%	20%	42%	8%
	RARL	14%	26%	40%	8%
	Ours	**88%**	**100%**	**100%**	**42%**

The continuous-time dynamics of the multi-agent system are given by

$$\begin{aligned} \dot{x}_e &= v_e \cos(\theta_e), & \dot{y}_e &= v_e \sin(\theta_e), & \dot{\theta}_e &= \omega_e, & \dot{v}_e &= a_e \\ \dot{x}_{p_i} &= v_{\text{const}} \cos(\theta_{p_i}), & \dot{y}_{p_i} &= v_{\text{const}} \sin(\theta_{p_i}), & \dot{\theta}_{p_i} &= \omega_{p_i}, & i &\in \{1, 2\} \end{aligned}$$

and we apply Euler integration to obtain the discrete-time dynamics.

The target set is successfully reached if the evader is caught (comes within a capture radius of $r_{\text{catch}} = 0.75$ of any pursuer) or if the evader crashes into an obstacle/wall:

$$g(x) = \min\left(\min_{i \in \{1,2\}} \left(\|p_e - p_{p_i}\| - r_{\text{catch}}\right), \text{SDF}(p_e)\right),$$

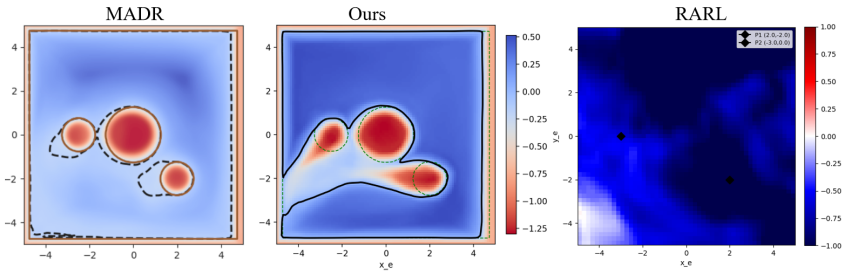


Figure 2: Value function heatmap: x_e-y_e slices at $[\theta_e, v_e, x_{p_1}, y_{p_1}, \theta_{p_1}, x_{p_2}, y_{p_2}, \theta_{p_2}] = [0, 1, 2, -2, -\pi, -2.5, 0, -\pi/2]$.

where SDF$(\cdot)$ denotes the signed distance function to the boundary and the central circular obstacle. The failure set is triggered if the pursuers fail the mission by colliding with the obstacles or each other:

$$\ell(x) = \min\left(\min_{i\in\{1,2\}} \text{SDF}(p_{p_i}), \|p_{p_1} - p_{p_2}\| - 2r_{\text{robot}}\right).$$

We compare against three state-of-the-art baselines: **MADR** Teoh et al. (2025) (an MPC-guided continuous-time PINN-based solver), **RARL** Hsu et al. (2021) (an RL-based solver for BRAT using Double Deep Q-Network learning), and **MPPI** Williams et al. (2016). We trained MADR for 250K epochs with carefully tuned hyperparameters and trained RARL with 30K warmup steps followed by 4M gradient updates. The problem horizons for both MADR and our method are set to 8s. MADR training took 7 hours, RARL training finished in 6 hours, and our method finished training in 7 hours on an RTX 4090 GPU. On the other hand, the MPPI controller does not require offline training but demands significantly more online computation. The MPPI pursuers operate independently, each individually minimizing their cumulative distance to the evader while incorporating a SDF penalty term to ensure obstacle avoidance. Conversely, the MPPI evader operates by simply maximizing its distance from the pursuers while avoiding obstacles.

The performance is benchmarked in a cross-play evaluation setting. We simulate 50 trajectories, each with a 20s duration, due to limited computes. Table 1 presents the quantitative catching rates across all combinations of pursuer and evader policies. Our framework demonstrates dominant performance in both the adversarial evading and cooperative pursuing roles. When our joint pursuer policy is deployed, it captures the MPPI evader in 88% of the rollouts and achieves a 100% catch rate against both the MADR and RARL evaders. Conversely, when our method is deployed in the evader role, it exhibits highly robust evasive capabilities. Against the baseline MPPI, MADR, and RARL pursuers, our evader successfully escapes in the vast majority of trials, restricting them to exceptionally low catch rates of 14%, 8%, and 8%, respectively. Notably, only our own learned pursuers pose a significant threat to our evader, achieving a 42% catch rate in self-play. Furthermore, it is worth highlighting that even the MPPI baseline—despite relying on a naive optimal control cost function without formal game-theoretic awareness—outperformed the learned MADR and RARL pursuers, catching its own evader 40% of the time compared to their 12% and 14%. The poor performance of the learned baselines can be directly attributed to the underlying problem structure. As previously established, formulating the problem with asymmetric control authority yields non-trivial optimal strategies but introduces significant numerical stiffness during training. To further illustrate this challenge, Appendix A presents a related low-dimensional case study demonstrating that even traditional grid-based numerical solvers can fail under these asymmetric conditions due to computational artifacts. We attribute the success of our framework to two key design choices. First, compared to the continuous-time PINN formulation of MADR, our discrete-time temporal window approach and gradient-free neural representation allow us to utilize standard activation functions, which significantly improves neural network capacity and training stability. Second, compared to the RL-driven approach, our method leveraged boundary conditions and the finetune rollout dataset for supervision, providing a substantially stronger training signal.

Finally, we qualitatively evaluate the learned value function slices at the x_e-y_e plane. For this specific configuration, the evader is initialized facing rightward ($\theta_e = 0$) with a speed of $v_e = 1$. The first pursuer is positioned at $(2.0, -2.0)$ facing leftward ($\theta_{p_1} = -\pi$), while the second pursuer starts at

$(-2.5, 0)$ facing downward ($\theta_{p_2} = -\pi/2$). As shown in Fig. 2, our method synthesizes a significantly larger catch set compared to MADR, which also aligns closely with human intuition. Conversely, RARL yields a highly nonphysical value function with visible artifacts. It is important to note, however, that RL-based methods like RARL are sensitive to hyperparameter tuning and process a steep learning curve for the users. Therefore, these results may not reflect the its maximum capability.

5 Conclusion and Limitations

In this work, we introduced a discrete-time, gradient-free neural framework for high-dimensional Hamilton-Jacobi (HJ) reachability and demonstrated its effectiveness in solving long-horizon pursue-evade games. Our method partitions long-horizon problems into temporal windows, adapting a teacher-student actor-critic framework to jointly approximate optimal policies and value functions. By inheriting the backward temporal curriculum from PINNs, we preserve stable boundary anchoring while maintaining the flexibility of discrete-time RL. To approximate infinite-horizon policies, we further adopt a two-phase offline synthesis strategy and a sliding-window online deployment scheme. Together, our framework achieves significant improvements over existing methods in solving long-horizon and numerically stiff reachability problems.

While our discrete-time framework scales robustly to high-dimensional systems and vision-based observations, several avenues for refinement remain. First, our method is sensitive to the integration time-step. Proper tuning is critical: excessively large steps introduce significant discretization error, while overly small steps shrink the directional value differences and induce biased policies. For simple reachability problems with closed-form dynamics, DeepReachMPC generally requires less tuning to provide strong baseline performance. Consequently, a valuable future direction is the empirical quantification of optimal Δt bounds or the development of adaptive time-stepping during the temporal curriculum.

Second, because our bang-bang action probing assumes affine dynamics, the framework provides suboptimal solutions for non-affine black-box systems, where RL-based methods theoretically approximate optimal solutions. This bang-bang nature may also struggle with tasks requiring soft and smooth control interactions. An exciting future direction is to replace single-step bang-bang teachers with multi-step episodic rollouts to relax the affine dynamics requirement—at the cost of higher variance in the learning signal—enabling the framework to solve reachability problems that demand more nuanced control, such as humanoid balancing or intricate manipulation tasks.

6 Team Contributions

- **Zeyuan Feng:** Everything.
- **External Collaborator (Ali Fuat Sahin, Role: Mentee):** Helped determine the core design choices for the offline synthesis algorithm used in the CoRL submission through ablation studies. Because this course project and the CoRL paper started at the same time and evolved together this quarter, there is inevitably some intersection. However, I made a strict effort to keep them separate, and Ali was not involved in any of the case studies or writing for this course project.

Changes from Proposal I skipped the final safety-performance co-optimization phase since it will introduce excessive contents to this project. This pivot has been mentioned in my project milestone submission.

References

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J. Tomlin. 2017a. Hamilton-Jacobi reachability: A brief overview and recent advances. In *IEEE 56th Annual Conference on Decision and Control (CDC)*. 2242–2253. doi:10.1109/CDC.2017.8263977

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. 2017b. Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances. In *IEEE Conference on Decision and Control (CDC)*.

Somil Bansal and Claire J Tomlin. 2021. DeepReach: A deep learning approach to high-dimensional reachability. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Javier Borquez, Kaustav Chakraborty, Hao Wang, and Somil Bansal. 2024. On safety and liveness filtering using hamilton-jacobi reachability analysis. *IEEE Transactions on Robotics* (2024).

Minh Bui, George Giovanis, Mo Chen, and Arrvindh Shriraman. 2022. Optimizeddp: An efficient, user-friendly library for optimal control and dynamic programming. *arXiv preprint arXiv:2204.05520* (2022).

Vamsi Krishna Chilakamarri, Zeyuan Feng, and Somil Bansal. 2025. Reachability analysis for black-box dynamical systems. *IEEE International Conference on Robotics and Automation (ICRA)* (2025).

Zeyuan Feng, Le Qiu, and Somil Bansal. 2025. Bridging Model Predictive Control and Deep Learning for Scalable Reachability Analysis. *Robotics: Science and Systems (RSS)* (2025).

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

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

Kai-Chieh Hsu, Vicenç Rubies-Royo, Claire J Tomlin, and Jaime F Fisac. 2021. Safety and liveness guarantees through reach-avoid reinforcement learning. *arXiv preprint arXiv:2112.12288* (2021).

Jingqi Li, Donggun Lee, Jaewon Lee, Kris Shengjun Dong, Somayeh Sojoudi, and Claire Tomlin. 2025. Certifiable Deep Learning for Reachability Using a New Lipschitz Continuous Value Function. arXiv:2408.07866 [eess.SY] `https://arxiv.org/abs/2408.07866`

K. Margellos and J. Lygeros. 2011. Hamilton-Jacobi Formulation for Reach–Avoid Differential Games. *IEEE Trans. Automat. Control* 56, 8 (2011), 1849–1861.

I. Mitchell. 2004. A toolbox of level set methods. *http://www. cs. ubc. ca/mitchell/ToolboxLS/toolboxLS.pdf* (2004).

Ian M Mitchell et al. 2007. A toolbox of level set methods. *UBC Department of Computer Science Technical Report TR-2007-11* 1 (2007), 6.

Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. 2005. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on automatic control* 50, 7 (2005), 947–957.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* 378 (2019), 686–707.

William Sharpless, Zeyuan Feng, Somil Bansal, and Sylvia Herbert. 2024. Linear Supervision for Nonlinear, High-Dimensional Neural Control and Differential Games. *arXiv preprint arXiv:2412.02033* (2024).

Aditya Singh, Zeyuan Feng, and Somil Bansal. 2025. Exact Imposition of Safety Boundary Conditions in Neural Reachable Tubes. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. 5489–5495. doi:`10.1109/ICRA55743.2025.11127972`

Vincent Sitzmann, Julien NP Martel, Alexander W Bergman, David B Lindell, and Gordon Wetzstein. 2020. Implicit neural representations with periodic activation functions. *arXiv preprint arXiv:2006.09661* (2020).

Ryan Teoh, Sander Tonkens, William Sharpless, Aijia Yang, Zeyuan Feng, Somil Bansal, and Sylvia Herbert. 2025. MADR: MPC-guided Adversarial DeepReach. arXiv:2510.18845 [cs.RO] `https://arxiv.org/abs/2510.18845`

Santiago Thorup, Luca Castelletto, Zeyuan Feng, and Somil Bansal. 2026. Neural Backward Reach-Avoid Tubes with MPC Supervision for High-Dimensional Systems: An Application to Safe Spacecraft Docking. arXiv:2605.02021 [cs.RO] `https://arxiv.org/abs/2605.02021`

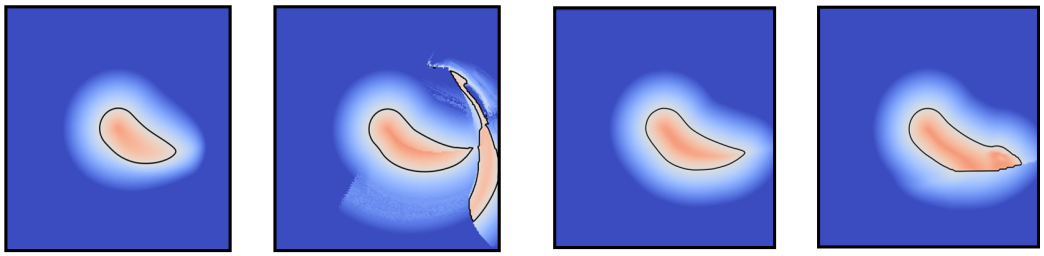
Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. 2016. Aggressive driving with model predictive path integral control. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 1433–1440. doi:10.1109/ICRA.2016.7487277

A Evaluation on a Low-Dimensional Two-Player Zero-Sum Game: 4D Human-Robot Collision Avoidance

We formulate a 4D human-robot collision avoidance scenario as a Worst-Case BRT computation. In this game, an ego vehicle (modeled as a 4D Dubins car) attempts to avoid an adversarial human agent (modeled as a 3D Dubins vehicle). The system is modeled in a relative coordinate frame with the following dynamics:

$$\begin{aligned}
\dot{x}_{\text{rel}} &= -v_{\text{ego}} + v_{\text{human}} \cos(\theta_{\text{rel}}) + \omega_{\text{ego}} y_{\text{rel}} \\
\dot{y}_{\text{rel}} &= v_{\text{human}} \sin(\theta_{\text{rel}}) - \omega_{\text{ego}} x_{\text{rel}} \\
\dot{\theta}_{\text{rel}} &= \omega_{\text{human}} - \omega_{\text{ego}} \\
\dot{v}_{\text{ego}} &= a_{\text{ego}}
\end{aligned}$$

The 4D state vector is defined as $x = [x_{\text{rel}}, y_{\text{rel}}, \theta_{\text{rel}}, v_{\text{ego}}]^\top$, where x_{rel} and y_{rel} denote the relative position of the human with respect to the ego vehicle, θ_{rel} is the relative heading, and v_{ego} is the linear velocity of the ego vehicle. The state bounds are given as $[-3.0, 3.0]^2 \times [-\pi, \pi] \times [0.1, 1.0]$. We use Euler integration to obtain the discrete-time dynamics.



(a) OptimizedDP solution (b) Our solution

Figure 3: Qualitative comparison of the 4D human-robot collision avoidance game. For both methods, we display the predicted value function (left sub-panel) alongside the reachability rollout cost evaluated from its induced policy (right sub-panel). The heatmap slices are taken at x-y plane where $\theta_{\text{rel}} = \pi/2$ and $v_{\text{ego}} = 1.0$. The grid-based OptimizedDP solution exhibits severe numerical artifacts due to the asymmetric control authority between players. In contrast, our neural framework maintains a relatively more consistent safety boundary that closely matches the empirical rollout cost.

The ego vehicle acts as the control player aiming to maximize safety, with control inputs $u = [\omega_{\text{ego}}, a_{\text{ego}}]^\top$ bounded by $\omega_{\text{ego}} \in [-1.0, 1.0]$ and $a_{\text{ego}} \in [-2.0, 2.0]$. The human acts as the disturbance player aiming to cause a collision, with a higher turn rate $d = \omega_{\text{human}} \in [-2.0, 2.0]$. The human is assumed to move at a constant linear velocity $v_{\text{human}} = 1.0$.

Table 2: Quantitative safety evaluation across 20,000 random rollouts in the 4D collision avoidance environment.

Method	**False Positives (FP)** $\downarrow$	**False Negatives (FN)** $\downarrow$
OptimizedDP	1569	**10**
Ours	**206**	35

The safety set is defined by avoiding a collision ball of radius $r_{\text{goal}} = 0.5$ around the ego vehicle: $\ell(x) = \sqrt{x_{\text{rel}}^2 + y_{\text{rel}}^2} - r_{\text{goal}}$.

In this scenario, both players possess distinct advantages—the ego vehicle has greater longitudinal flexibility through variable speed, while the human has a higher maximum turn rate. This asymmetric control authority introduces sharp value gradients that cause numerical artifacts in the grid-based OptimizedDP solver Bui et al. (2022), as illustrated by the qualitative value and cost function heatmaps in Fig. 3a. Quantitatively, the value-policy consistency is further evaluated with numbers of FP and FN among 20,000 trajectory rollouts, as shown in Table. 2. Our method yields a significantly smaller number of FP while being slightly more conservative with a higher number of FN.