Implicit DeepReach Without Implicit Differentiation

Xingjian Li XINGJIAN.LI@AUSTIN.UTEXAS.EDU
Oden Institute, The University of Texas at Austin

Kelvin Kan KELVIN.KAN@MATH.UCLA.EDU
Department of Mathematics, UCLA

Deepanshu Verma DVERMA@CLEMSON.EDU
School of Mathematical and Statistical Sciences, Clemson University

Krishna Kumar KRISHNAK@UTEXAS.EDU
Oden Institute, The University of Texas at Austin

Hayden Schaeffer HAYDEN@MATH.UCLA.EDU
Department of Mathematics, UCLA

Samy Wu Fung SWUFUNG@MINES.EDU
Department of Applied Mathematics and Statistics, Colorado School of Mines

Abstract

We consider the problem of approximating backward reachable and reach-avoid tubes for high-dimensional nonlinear systems whose Hamiltonians require numerical optimization over constrained controls and disturbances. Existing neural Hamilton–Jacobi (HJ) reachability methods alleviate the dimensionality bottleneck of grid-based solvers, but evaluating Hamiltonians without closed-form expressions introduces an additional computational challenge. We show that, under suitable assumptions, the gradients required for value-function training can be computed without differentiating through the inner optimization problem. Specifically, we establish an envelope-type gradient identity for the saddle point Hamiltonian, under which the optimal control and disturbance enter backpropagation as constants without gradient tracking. This enables end-to-end neural HJ reachability with numerically evaluated Hamiltonians while avoiding both solver unrolling and implicit differentiation. We demonstrate the accuracy and computational efficiency of the proposed approach on nonlinear reachability and reach-avoid problems of different scales.

Keywords: Hamilton–Jacobi Reachability, Implicit Hamiltonian, Saddle Point Envelope Theorem

1. Introduction

Hamilton–Jacobi (HJ) reachability provides a principled framework for reasoning about safety and robustness in dynamical systems. Given a target or unsafe set, reachability analysis characterizes the states from which an adversarial disturbance can force the system into the target despite the action of a controller (Mitchell et al., 2005a; Isaacs, 1965). The resulting backward reachable tube (BRT) provides both a safety certificate and an associated feedback controller, making HJ reachability a powerful tool for safety-critical control.

A primary challenge in HJ reachability is the curse of dimensionality. Classical methods solve the associated Hamilton-Jacobi-Isaacs (HJI) equation or variational inequality on a state-time grid (Mitchell et al., 2005b; Kundu and Kunisch, 2026), these methods are accurate but restricted to relatively low-dimensional systems. Neural approaches such as DeepReach (Bansal and Tomlin, 2021) alleviate this bottleneck by representing the value function with a neural network and enforcing the HJI equation at sampled state-time points. Subsequent work has further improved the scalability

© X. Li, K. Kan, D. Verma, K. Kumar, H. Schaeffer & S.W. Fung.

and accuracy of this paradigm through additional supervision and guidance (Sharpless et al., 2025; Teoh et al., 2025).

In this work, we focus on a separate bottleneck that arises when the Hamiltonian of the system does not admit a closed form solution and is instead defined *implicitly* by a saddle point problem. While optimal control and disturbance inputs can admit closed-form expressions in some standard reachability problems, more general nonlinear dynamics, coupled control–disturbance interactions, and complex admissible input sets generally require solving a numerical optimization problem to evaluate the Hamiltonian (Betts, 2010; Gelphman et al., 2026a,b). Recent approaches address such Hamiltonians through learned surrogates, data-driven approximations, or conservative bounds (Chilakamarri et al., 2025; Choi et al., 2025; Lin et al., 2026). These methods make the Hamiltonian more tractable before or outside the value-function computation, but they usually require additional computation or data collection and can introduce approximation error.

An alternative is to retain the numerical Hamiltonian optimization directly within neural HJI training, as explored in related implicit deep learning approaches for optimal control (Gelphman et al., 2026b). However, embedding the optimization into the training loop introduces an additional differentiation bottleneck. Each value-function update requires repeatedly solving the pointwise control-disturbance game, and conventional differentiation approaches propagate gradients through its solution. Unrolling the inner solver can incur substantial memory and computational overhead, while exact implicit differentiation typically requires solving an additional linear system for each Hamiltonian evaluation. In this work, we show that neither approach is necessary. By establishing an envelope-type gradient identity for the saddle point Hamiltonian, we obtain the gradients required for neural HJI training without differentiating through the inner optimization. In particular, our main contributions are: **(i) End-to-end HJ reachability with implicit Hamiltonians.** We develop a neural HJ reachability framework that directly incorporates numerically solved control-disturbance games into value-function training, enabling the treatment of nonlinear systems without tractable closed-form Hamiltonians. **(ii) Theoretically justified gradients without optimizer sensitivities.** Using a saddle point envelope argument, we establish exact gradient formulas under light assumptions, eliminating the need for solver unrolling or implicit differentiation through the inner optimization. **(iii) Numerical validation.** We demonstrate the accuracy and scalability of our approach on nonlinear reachability and reach-avoid problems in dimensions ranging from 2 to 50, showing close agreement with reference solutions and reliable closed-loop performance.

2. Background

2.1. Hamilton-Jacobi Reachability

Consider the controlled dynamical system over a finite time horizon

$$\dot{x}(s) = f\big(x(s), u(s), d(s)\big), \quad s \in [t, T], \tag{1}$$

where $x(s) \in \mathbb{R}^n$ is the state, $u(s) \in \mathcal{U}$ is the control, and $d(s) \in \mathcal{D}$ is an adversarial disturbance with t denoting an arbitrary initial time and T is a fixed terminal time. Let the target set be represented by

$$\mathcal{L} = \{x \in \mathbb{R}^n : \ell(x) \leq 0\}, \tag{2}$$

for a sufficiently regular target function ℓ. The target set can represent either a set of goal states or unsafe states, and $\ell(x)$ is typically chosen as a signed distance function.

The **Backward Reachable Tube (BRT)** (Mitchell et al., 2005a; Bansal and Tomlin, 2021) of a given problem consists of states from which the disturbance can force the system into $\mathcal{L}$ at some time over $[t, T]$ despite the action of the controller. This is characterized via the value function

$$V(x,t) = \inf_{d(\cdot)} \sup_{u(\cdot)} \min_{s\in[t,T]} \ell\big(x(s)\big), \tag{3}$$

and the BRT is given by the zero sublevel set of the value function for any given t, that is,

$$\mathcal{V}(t) = \{x \in \mathbb{R}^n : V(x,t) \leq 0\}. \tag{4}$$

Define the HJI Hamiltonian

$$H(x,p) := \max_{u\in\mathcal{U}} \min_{d\in\mathcal{D}} p^\top f(x,u,d). \tag{5}$$

where p is the adjoint variable to the dynamics. Under standard regularity assumptions and the Isaacs condition, the value function satisfies the HJI variational inequality (VI), which reads

$$\min\left\{\partial_t V(x,t) + H\big(x, \nabla_x V(x,t)\big),\ \ell(x) - V(x,t)\right\} = 0, \tag{6}$$

with terminal condition $V(x,T) = \ell(x)$ for all $x \in \mathbb{R}^n$. Crucially, the VI formulation accounts for the possibility of reaching the target set at any time within the horizon, rather than only at the terminal time, distinguishing it from the standard HJI PDE.

Similarly, we can define the **Backward Reach-Avoid Tube (BRAT)** (Fisac et al., 2015), which characterizes states from which the controller can reach the target set $\mathcal{L}$ despite adversarial disturbances, while avoiding an unsafe set $\mathcal{G}$, defined as

$$\mathcal{G} = \{x \in \mathbb{R}^n : g(x) > 0\}. \tag{7}$$

Here g is also typically chosen as a signed distance function describing obstacles or unsafe states of the system, such as joint positions that violate a robot's safety requirements. We note for goal-reaching problems, the roles of the control and disturbance in the Hamiltonian are reversed. Nonetheless, the BRAT can be similarly defined as the zero sublevel set of the corresponding value function $V(x,t)$, which satisfies a slightly different variational inequality

$$\max\left\{\min\left\{\partial_t V(x,t) + H(x, \nabla_x V(x,t)),\ \ell(x) - V(x,t)\right\},\ g(x) - V(x,t)\right\} = 0, \tag{8}$$

with terminal condition $V(x,T) = \max\{\ell(x), g(x)\}$. Here the outer maximum enforces the state constraint by excluding trajectories that enter the unsafe set during the horizon. The BRAT $\mathcal{V}(t)$, can be recovered similarly following (4).

2.2. Neural HJ Reachability

To alleviate the curse of dimensionality of grid-based HJ solvers, neural approaches such as DeepReach (Bansal and Tomlin, 2021) approximate the value function using a neural network $V_\theta(x,t)$ parameterized by θ. Following the physics-informed neural network (PINN) paradigm (Raissi et al., 2019; Wang et al., 2023), the network is trained by minimizing the HJI variational inequality residual and enforcing the terminal condition at sampled state-time points. Specifically, the training objective takes the form

$$L(\theta) = L_{\text{VI}}(\theta) + \lambda L_T(\theta), \tag{9}$$

where

$$L_{\mathrm{VI}}(\theta) = \mathbb{E}_{(x,t)\sim\rho}\left[\left(\min\left\{\partial_t V_\theta(x,t) + H(x, \nabla_x V_\theta(x,t)),\ \ell(x) - V_\theta(x,t)\right\}\right)^2\right], \tag{10}$$

$$L_T(\theta) = \mathbb{E}_{x\sim\rho_T}\left[\left(V_\theta(x,T) - V(x,T)\right)^2\right]. \tag{11}$$

Here, ρ and ρ_T denote the sampling distributions over interior state-time points and terminal states, respectively, and $\lambda > 0$ balances the two loss terms. The formulation above corresponds to the BRT problem; the BRAT loss follows analogously using the reach-avoid variational inequality.

A key requirement of these approaches is the repeated evaluation of the Hamiltonian $H(x, \nabla_x V_\theta)$ during training. Existing neural HJ methods typically consider settings in which the saddle point problem admits a closed-form solution. However, for more complex dynamics, determining the optimal control and disturbance may require numerical optimization. Addressing the resulting differentiability and computational cost bottlenecks remains an open challenge for neural HJ methods.

3. Neural HJ Reachability with Implicit Hamiltonians

As discussed in Section 2, existing neural HJ reachability methods typically rely on Hamiltonians that admit closed-form evaluations. For general nonlinear dynamics with constrained controls and disturbances, the Hamiltonian is defined implicitly through a saddle point optimization problem. Under a standard projected gradient ascent-descent (PGDA) formulation, the inner optimization can be recast as the fixed-point problem over $(u^\star, d^\star)$ with each iteration defined as

$$u^{k+1} = \Pi_{\mathcal{U}}\left(u^k + \eta_u \nabla_u \left[p_\theta^\top f(x, u^k, d^k)\right]\right),$$

$$d^{k+1} = \Pi_{\mathcal{D}}\left(d^k - \eta_d \nabla_d \left[p_\theta^\top f(x, u^k, d^k)\right]\right).$$

Here, $\Pi_{\mathcal{U}}$ and $\Pi_{\mathcal{D}}$ denote projections onto the admissible sets, and η_u, $\eta_d > 0$ are the step sizes. Under suitable convex–concave assumptions PGDA can provably converge, more general saddle point problems may require alternative algorithms (Korpelevich, 1976; Letcher et al., 2019; Mazumdar et al., 2025). Regardless of the solver, differentiating through the inner optimization introduces an additional computational bottleneck in neural HJI training. Two conventional approaches exist: **(i) Automatic Differentiation (AD):** Unrolling the inner solver and backpropagating through its iterations, which can incur substantial computational and memory overhead especially as problem size and number of inner iterations increase. **(ii) Implicit Differentiation:** Using the implicit function theorem (IFT) (Lorraine et al., 2020) it is possible to compute the sensitivities of the saddle point directly under

$$\begin{bmatrix}\partial_\theta u^\star \\ \partial_\theta d^\star\end{bmatrix} = -\begin{bmatrix}\nabla^2_{uu}(p_\theta^\top f) & \nabla^2_{ud}(p_\theta^\top f) \\ \nabla^2_{du}(p_\theta^\top f) & \nabla^2_{dd}(p_\theta^\top f)\end{bmatrix}^{-1}\begin{bmatrix}\partial_\theta \nabla_u (p_\theta^\top f) \\ \partial_\theta \nabla_d (p_\theta^\top f)\end{bmatrix}\Bigg|_{(u^\star, d^\star)}. \tag{12}$$

Algorithm 1: Gradient-free neural HJ reachability with implicit Hamiltonians.

Input: Dynamics f, target ℓ, value network V_θ, learning rate α
Output: Trained value function V_θ
while *not converged* **do**
  Sample $\{(x_i, t_i)\}_{i=1}^N \sim \rho$
  $p_i \leftarrow \nabla_x V_\theta(x_i, t_i)$
  $(V_i, V_{t,i}) \leftarrow (V_\theta(x_i, t_i), \partial_t V_\theta(x_i, t_i))$
  for $i = 1, \ldots, N$ **do**
    $(u_i^\star, d_i^\star) \leftarrow$ SaddleSolve$(x_i, \mathrm{stopgrad}(p_i))$
    $H_i \leftarrow p_i^\top f(x_i, \mathrm{stopgrad}(u_i^\star, d_i^\star))$
    $R_i \leftarrow \min\{V_{t,i} + H_i, \ell(x_i) - V_i\}$
  end
  $L \leftarrow \frac{1}{N}\sum_{i=1}^N R_i^2$
  $\theta \leftarrow \theta - \alpha \nabla_\theta L$
end

Here we assume an interior saddle point for simplicity. While efficient, boundary projections can lead to singular or ill-conditioned Jacobians and subsequent numerical instabilities, especially in large-scale problems, as demonstrated in (Li et al., 2026b). Both approaches introduce additional differentiation costs on top of the repeated saddle point solves required during training, rendering them impractical at scale.

In this work, we show that differentiating through the inner optimization *is unnecessary*. Leveraging a saddle point envelope type argument, we establish exact gradients for neural HJI training without differentiating through the inner solver, which has also been observed in the optimal transport setting Amos (2023); Park et al. (2026). We state the formal theory in Section 3.1 and present the resulting end-to-end training algorithm in Section 3.2.

3.1. Theoretical Analysis

We now establish an envelope type gradient identity that enables neural HJ reachability training without differentiating through the inner saddle point optimization.

Notation. We write $z = (u, d) \in \mathcal{U} \times \mathcal{D}$, and use z and (u, d) interchangeably. Let $p_\theta := \nabla_x V_\theta(x, t)$ and $\varphi(p, z) := p^\top f(x, z)$, so that $H(x, p) = \max_{u\in\mathcal{U}} \min_{d\in\mathcal{D}} \varphi(p, u, d)$ by (5).

Assumptions. For an arbitrary parameter θ_0, we assume: **(A1)** $\mathcal{U}$ and $\mathcal{D}$ are nonempty, compact and convex; **(A2)** $f(x, \cdot)$ is C^1; **(A3)** the mapping $\theta \mapsto p_\theta$ is differentiable at θ_0; **(A4)** there exists $\delta_0 = \delta(p_{\theta_0}) > 0$ such that, for every $p \in \mathcal{B}_0 := \{p \in \mathbb{R}^n : \|p - p_{\theta_0}\|_2 < \delta_0\}$, the Isaacs condition $\max_{u\in\mathcal{U}} \min_{d\in\mathcal{D}} \varphi(p, u, d) = \min_{d\in\mathcal{D}} \max_{u\in\mathcal{U}} \varphi(p, u, d)$ holds; **(A5)** the function $\varphi(p_{\theta_0}, \cdot, \cdot)$ has a unique saddle point on $\mathcal{U} \times \mathcal{D}$, which we denote by $z^\star(p_{\theta_0})$.

Theorem 1 *Under* (A1)–(A5)*, the mapping $\theta \mapsto H(x, p_\theta)$ is differentiable at θ_0, with*

$$\begin{aligned}\partial_\theta H(x, p_\theta)|_{\theta_0} &= f\big(x, z^\star(p_{\theta_0})\big)^\top (\partial_\theta p_\theta)|_{\theta_0} + \underbrace{\big[\nabla_z \big(p_{\theta_0}^\top f(x, z)\big)\big|_{z=z^\star(p_{\theta_0})}\big]^\top (\partial_\theta z^\star(p_\theta))\,|_{\theta_0}}_{0 \text{ contribution}} \\ &= f(x, z^\star(p_{\theta_0}))^\top (\partial_\theta p_\theta)|_{\theta_0}.\end{aligned} \tag{13}$$

In other words, the extra term that would result from chain rule does not appear. Thus, the saddle point can be treated as a constant during backpropagation, with no gradients propagated through $z^\star(p_\theta)$ or the solver that computes it.

We defer the proof of Theorem 1 to Appendix Section E, numerical validation of Theorem 1 is also provided in Appendix Section C.

Remark. We highlight several important properties of Theorem 1. **(i)** Assumptions (A1)-(A5) are compatible with standard HJ reachability settings involving compact convex input constraints, sufficiently regular dynamics, and locally well-defined saddle points. In particular, the result does not require strong convexity-concavity or second-order nonsingularity conditions crucial for IFT. **(ii)** Theorem 1 applies pointwise at any θ_0 satisfying these assumptions and therefore extends throughout training wherever they continue to hold. We validate this numerically and measure the effect of inexact inner solves in Appendix Section C, enabling our proposed Algorithm 1. **(iii)** The result applies to arbitrary compact and convex control and disturbance sets, beyond simple box constraints.

Moreover, the gradient identity is independent of the numerical saddle point solver. **(iv)** Finally, the result does not require differentiability of the optimal control and disturbance mappings with respect to the network parameters. Consequently, the gradient identity remains valid for saddle points in the interior or on the boundary of the admissible sets.

3.2. Gradient-Free Training for Neural HJ Reachability with Implicit Hamiltonians

Following the envelope identity established in Section 3.1, we develop an end-to-end neural HJ training framework that incorporates numerically evaluated Hamiltonians without differentiating through the inner saddle-point solver. We describe its main implementation components below.

Exact terminal condition enforcement through network parameterization. Following (Singh et al., 2025), we enforce the terminal condition $V_\theta(x,T) = \ell(x)$ by parameterizing the value function as

$$V_\theta(x,t) = e^{-(T-t)}\ell(x) + \left(1 - e^{-(T-t)}\right) N_\theta(x,t), \tag{14}$$

where N_θ denotes the neural network output. Unlike (Singh et al., 2025), we opt for an exponential decay to reduce the influence of the nonsmooth terminal function $\ell(x)$ away from T, which empirically we observe improves training. This parameterization satisfies the terminal condition exactly, eliminating the need for the terminal loss (11).

Saddle point solver. We primarily employ PGDA to solve the inner saddle point problem, using adaptive step sizes based on the Barzilai–Borwein (BB) method (Barzilai and Borwein, 1988) to improve practical convergence. The step sizes are updated as

$$\begin{aligned}
\eta_u^{k+1} &= \mathrm{clip}\left(\frac{\|u^{k+1}-u^k\|_2^2}{(u^k-u^{k+1})^\top(\nabla_u\varphi^{k+1}-\nabla_u\varphi^k)}, \eta_{\min}, \eta_{\max}\right), \\
\eta_d^{k+1} &= \mathrm{clip}\left(\frac{\|d^{k+1}-d^k\|_2^2}{(d^{k+1}-d^k)^\top(\nabla_d\varphi^{k+1}-\nabla_d\varphi^k)}, \eta_{\min}, \eta_{\max}\right).
\end{aligned} \tag{15}$$

Here, $\eta_{\min}$ and $\eta_{\max}$ are predefined bounds for numerical stability, and step sizes are updated independently for each sample. This provides a computationally inexpensive adaptive step-size selection, particularly useful when the saddle point objective is relatively flat. We evaluate its effectiveness against a fixed step size solver in Section C. We note that while PGDA convergence depends on the structure of the underlying game, the proposed training framework is independent of the saddle point solver and can accommodate any alternative numerical methods.

Combining the exact terminal-condition parameterization with gradient-free Hamiltonian evaluation yields the end-to-end training procedure summarized in Algorithm 1. After training, the BRT is recovered from the zero sublevel set of V_θ, and the corresponding feedback policy can be obtained using the same saddle point solver. For conciseness, Algorithm 1 is written in BRT formulation; the BRAT extension follows analogously.

4. Experiments

We consider three numerical experiments, covering both BRT and BRAT problems, with state dimensions from 2 to 50. In all cases, the Hamiltonians are implicitly defined and require iterative solvers to compute the saddle points, and optimal saddle points occur both in the interior and on the

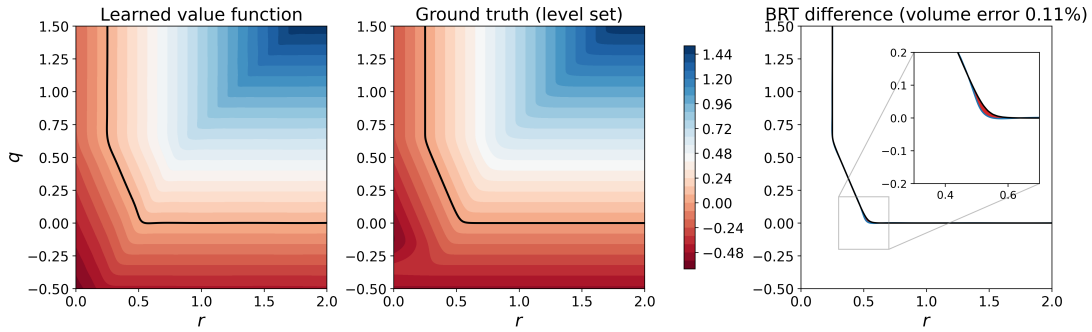


Figure 1: Learned and reference value functions for the energy-constrained pursuit-evasion problem. **Left:** Learned V_θ; **Middle:** grid-based reference; **Right:** BRT volume differences.

boundaries of the admissible sets, providing sufficient numerical evidence supporting our approach. All experiments were conducted on a single NVIDIA RTX 5090 GPU. Code will be made available upon publication of the work. We refer to Appendix Section B and Section D for additional details.

4.1. Energy Constrained Pursuit–Evasion Example

We first consider a two-player zero-sum differential game with state $x = (r, q) \in \mathcal{X} = [0, 2] \times [-0.5, 1.5]$, where r denotes the relative distance between the players and q their relative energy resource, with $q > 0$ indicating an advantage for the evader. The evader and pursuer select controls $u, d \in [0, 2]$, respectively, under the dynamics

$$\begin{cases} \dot{r} = (1 + 0.2q)\tanh(u) - 1.2(1 - 0.1q)\tanh(d), \\ \dot{q} = -0.2u^2 + 0.2d^2. \end{cases} \tag{16}$$

The target set is defined by $\mathcal{L} = \{(r, q) : r \leq 0.25\} \cup \{(r, q) : q \leq 0\}$, with terminal function $\ell(r, q) = \min\{r - 0.25, q\}$, the target set is reached when either the relative distance falls below the given threshold or the evader exhausts its resource advantage.

Given the finite time horizon $T = 1.5$, the BRT can be defined as in Section 2, the Hamiltonian is separable in u and d, allowing their optimal values to be computed independently from their respective stationarity conditions

$$\begin{cases} p_r(1 + 0.2q)\,\mathrm{sech}^2(u) - 0.4p_q u & = 0, \\ -1.2p_r(1 - 0.1q)\,\mathrm{sech}^2(d) + 0.4p_q d & = 0, \end{cases} \tag{17}$$

subject to the input constraints, with $p = [p_r, p_q]^\top$ and $p = \nabla_x V(x, t)$. The Hamiltonian is provably convex-concave over the input domain, and the separable structure enables comparison against a grid-based level-set solver while retaining nontrivial numerical optimization of the optimal inputs.

We refer to Appendix Section D for details on model parameterization and training details. As shown in Figure 1, the learned value function closely matches the grid-based reference solution, achieving a training loss on the order of 10^{-6} and a BRT volume error of only 0.11%.

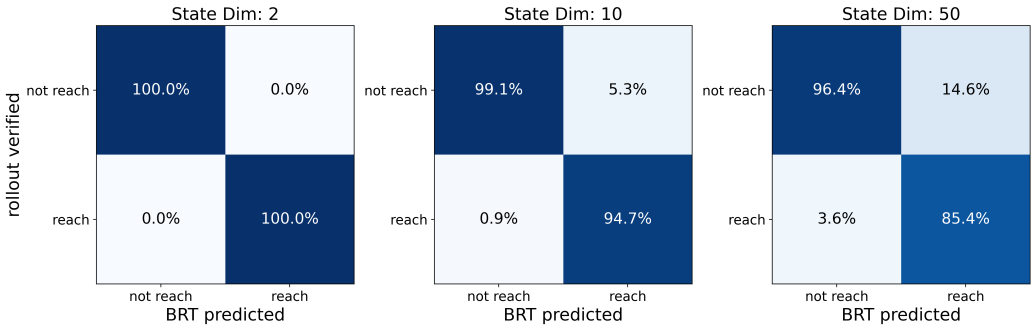


Figure 2: Closed-loop consistency of learned BRT predictions across state dimensions, quantifying mismatches between BRT predictions and closed-loop rollouts. Confusion matrices are column-normalized by predicted class.

4.2. Extending to High Dimensions

We extend the 2D energy-constrained pursuit-evasion example to N independent evader-pursuer pairs, with joint state $x = (r_1, q_1, \ldots, r_N, q_N) \in \mathbb{R}^{2N}$ and dynamics

$$\begin{cases} \dot{r}_i = (1 + k_i^1 q_i)\tanh(u_i) - 1.2(1 - k_i^2 q_i)\tanh(d_i), \\ \dot{q}_i = -\alpha_i u_i^2 + \beta_i d_i^2, \end{cases} \qquad i = 1, \ldots, N. \tag{18}$$

The target set is defined by $\mathcal{L} = \{x : \ell(x) \leq 0\}$, where $\ell(x) = \min_{i=1,\ldots,N}\{r_i - r_i^{\text{reach}}, q_i - q_i^{\text{reach}}\}$ so that the target is reached whenever any pair violates its separation or energy threshold.. We refer to Appendix Section B for the complete problem formulation.

We consider 10D and 50D extensions to evaluate scalability of our proposed method. As shown in Figure 5, training converges in both cases, with increasing residual losses at higher dimensions given the added complexity. To assess the learned BRTs without grid-based reference solutions, we perform a closed-loop consistency test on 2048 randomly sampled initial states for each problem, comparing BRT predictions with rollouts under the learned saddle-point policy, similar to that of (Bansal and Tomlin, 2021). The column-normalized confusion matrices in Figure 2 show that agreement remains high, particularly for states predicted to be unreachable (99.1% in 10D and 96.4% in 50D). Agreement for predicted-reachable states decreases to 94.7% and 85.4%, respectively, indicating greater classification disagreement as dimensionality increases.

4.3. Two-Vehicle Collision Avoidance in 6D

We consider a two-vehicle collision avoidance problem based on Bansal and Tomlin (2021), with joint state $x = (x_1, \ldots, x_6) \in \mathbb{R}^4 \times \mathbb{S}^1 \times \mathbb{S}^1$. Unlike the standard constant speed setting, we introduce steering-dependent speed for the evader, resulting in the modified dynamics

$$\begin{cases} \dot{x}_1 = 0.75\cos x_3, & \dot{x}_4 = 0.75e^{-0.4(u/3)^2}\cos x_6, \\ \dot{x}_2 = 0.75\sin x_3, & \dot{x}_5 = 0.75e^{-0.4(u/3)^2}\sin x_6, \\ \dot{x}_3 = d, & \dot{x}_6 = u, \end{cases}$$

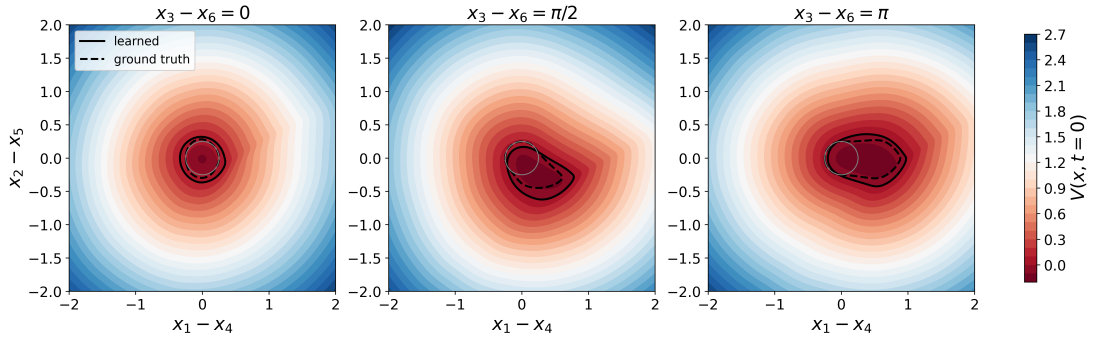


Figure 3: Learned and reference BRTs across relative-heading slices, showing close agreement with a grid-based reference solution.

where $u, d \in [-3, 3]$ denote the evader's and pursuer's turn-rate controls, respectively. This modification introduces nonlinear control dependence into the Hamiltonian, requiring numerical optimization of the evader's optimal control. We follow the capture target formulation of Bansal and Tomlin (2021), with the complete problem definition given in Appendix Section B.

Training yielded a MSE loss of approximately 2×10^{-3}. We note although the model is trained directly in the full 6D state space, a reference solution can be computed in 3D using relative coordinates, exploiting the translation and rotation symmetries of the dynamics. As shown in Figure 3, the learned BRT closely matches a grid-based reference solution across multiple two-dimensional slices, with slightly conservative predictions. Importantly, we obtain the desired results without enforcing said symmetries during neural HJ training.

We further evaluate the learned value function through closed-loop trajectory rollouts in Figure 6 in the Appendix, using the same numerical Hamiltonian optimization as during training. The trajectories demonstrate both successful evasion and capture, with the inner solver achieving residual errors below 10^{-4} across the tested scenarios. The steering-dependent speed also changes the reachable set compared with the constant speed setting: by reducing its speed during rapid turning, the evader can escape from certain configurations that would otherwise lead to capture. The trajectories also illustrate how initially reachable states may leave the BRT over time.

4.4. Quadrotor Reach-Avoid Control under Adversarial Disturbances

We consider a 6D planar quadrotor reach-avoid problem under adversarial wind disturbances. The quadrotor is controlled through two bounded rotor thrusts and must reach a prescribed target region while avoiding ground collision and excessive pitch angles. The wind disturbance enters through nonlinear velocity-dependent drag, requiring numerical optimization to determine the worst case disturbance. We refer to Appendix Section B for the complete dynamics and problem formulation.

To address training difficulties arising from the nested BRAT variational inequality, we adopt a curriculum learning strategy common in PINNs literature (Bansal and Tomlin, 2021; Krishnapriyan et al., 2021; Wang et al., 2022), where we gradually extend the sampled time horizon from T to 0 during training. We evaluate the learned value function through a closed-loop consistency test on 2048 randomly sampled initial states at $t = 0$. As shown in Table 1, the rollout outcomes agree with the predicted reach-avoid classifications in over 90% of cases for both predicted classes, with

Table 1: Closed-loop consistency of BRAT predictions using learned value function. "Boundary" denotes $|V_\theta(x,0)| < 0.05$. Both crashed and timeout cases have $V_\theta(x,0) > 0$.

		Reached	Crashed	Timeout	Total
$V_\theta(x,0) \leq 0$	**All**	**837 (90.8%)**	**71 (7.7%)**	**14 (1.5%)**	**922**
	Boundary	46	44	7	97
	Confident	791	27	7	825
$V_\theta(x,0) > 0$	**All**	**62 (5.5%)**	**911 (80.9%)**	**153 (13.6%)**	**1126**
	Boundary	24	53	4	81
	Confident	38	858	149	1045

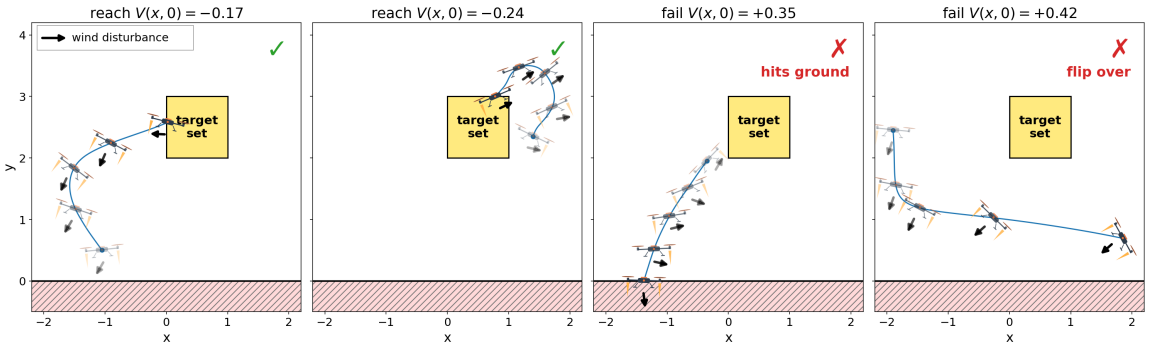


Figure 4: Rollouts for quadrotor control under disturbances using the learned V_θ. Successful and failed trajectories are shown, with outcomes consistent with the BRAT predictions.

most discrepancies occurring near the zero level set. We further visualize representative trajectories in Figure 4, where feedback controls and worst-case disturbances derived from the learned value function produce outcomes consistent with the predicted BRAT.

5. Conclusion

We presented an end-to-end neural HJ reachability framework for nonlinear systems with implicitly defined Hamiltonians. Using a perturbation argument based on saddle-point inequalities, we prove that gradients can be computed without differentiating through the inner optimization, avoiding the overhead and numerical instability under solver unrolling or implicit differentiation. Experiments on both reachability and reach-avoid problems demonstrate the effectiveness and scalability of our approach, extending neural HJ methods to a much broader and more practical class of problems.

References

Brandon Amos. On amortizing convex conjugates for optimal transport. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TQ5WUwS_4ai.

Somil Bansal and Claire J. Tomlin. DeepReach: A deep learning approach to high-dimensional reachability. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1817–1824. IEEE, 2021. doi: 10.1109/ICRA48506.2021.9561949.

Robert G Bartle and Donald R Sherbert. *Introduction to real analysis*, volume 2. Wiley New York, 2000.

Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.

John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM, 2010.

Vamsi Krishna Chilakamarri, Zeyuan Feng, and Somil Bansal. Reachability analysis for black-box dynamical systems. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3552–3558. IEEE, 2025. doi: 10.1109/ICRA55743.2025.11127528.

Jason J. Choi, Christopher A. Strong, Koushil Sreenath, Namhoon Cho, and Claire J. Tomlin. Data-driven hamiltonian for direct construction of safe set from trajectory data. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pages 2951–2958. IEEE, 2025. doi: 10.1109/CDC57313.2025.11312763.

Zeyuan Feng, Le Qiu, and Somil Bansal. Bridging Model Predictive Control and Deep Learning for Scalable Reachability Analysis. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.114.

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

Samy Wu Fung, Howard Heaton, Qiuwei Li, Daniel McKenzie, Stanley Osher, and Wotao Yin. JFB: Jacobian-free backpropagation for implicit networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6648–6656, 2022.

Eric Gelphman, Deepanshu Verma, Nicole Tianjiao Yang, Stanley Osher, and Samy Wu Fung. End-to-end training of high-dimensional optimal control with implicit hamiltonians via jacobian-free backpropagation. In *2026 American Control Conference (ACC)*. IEEE, 2026a.

Eric Gelphman, Deepanshu Verma, Nicole Tianjiao Yang, Stanley Osher, and Samy Wu Fung. On the convergence of jacobian-free backpropagation for optimal control problems with implicit hamiltonians. *arXiv preprint arXiv:2602.00921*, 2026b.

Gautier Hattenberger, Murat Bronz, and Jean-Philippe Condomines. Evaluation of drag coefficient for a quadrotor model. *International Journal of Micro Air Vehicles*, 15:17568293221148378, 2023.

Howard Heaton and Samy Wu Fung. Explainable ai via learning to optimize. *Scientific Reports*, 13 (1):10103, 2023.

Howard Heaton, Samy Wu Fung, Aviv Gibali, and Wotao Yin. Feasibility-based fixed point networks. *Fixed Point Theory and Algorithms for Sciences and Engineering*, 2021(1):21, 2021.

William Hofgard. Convergence guarantees for neural network-based hamilton–jacobi reachability. In *NeurIPS 2024 Workshop on Data-driven and Differentiable Simulations, Surrogates, and Solvers*, 2024. URL https://openreview.net/forum?id=SH1isYejz9.

Rufus Isaacs. *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. John Wiley & Sons, New York, 1965.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Galina M Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.

Sudeep Kundu and Karl Kunisch. Policy iteration for hamilton-jacobi-isaacs equations with control constraints and comparison with hamilton-jacobi-bellman equations. *arXiv preprint arXiv:2609.29368*, 2026.

Minhyeok Kwon and Yongsoon Eun. Quadrotor dynamics in a wind field: Equilibria analysis and energy dissipation. *International Journal of Control, Automation and Systems*, 22(11):3275–3284, 2024.

Alistair Letcher, David Balduzzi, Sébastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. Differentiable game mechanics. *Journal of Machine Learning Research*, 20 (84):1–40, 2019.

Xingjian Li, Kelvin Kan, Deepanshu Verma, Krishna Kumar, Stanley Osher, and Ján Drgoňa. Zero-shot transferable solution method for parametric optimal control problems. In *2026 American Control Conference (ACC)*, pages 2003–2010. IEEE, 2026a.

Xingjian Li, Kelvin Kan, Deepanshu Verma, Krishna Kumar, Stanley Osher, and Samy Wu Fung. End-to-end learning of safe optimal feedback control in high dimensions with control barrier function layers. *arXiv preprint arXiv:2607.20674*, 2026b.

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

Albert Lin, Alessandro Pinto, and Somil Bansal. Robust verification of controllers under state uncertainty via hamilton–jacobi reachability analysis. In *Proceedings of the 8th Annual Learning for Dynamics and Control Conference*, volume 331 of *Proceedings of Machine Learning Research*, pages 1432–1446. PMLR, 2026.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International conference on artificial intelligence and statistics*, pages 1540–1552. PMLR, 2020.

Robert Mahony, Vijay Kumar, and Peter Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE robotics & automation magazine*, 19(3):20–32, 2012.

Eric Mazumdar, S Shankar Sastry, and Michael I Jordan. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. *ACM/IMS Journal of Data Science*, 2(2):1–26, 2025.

Daniel McKenzie, Howard Heaton, and Samy Wu Fung. Differentiating through integer linear programs with quadratic regularization and davis-yin splitting. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856. URL https://openreview.net/forum?id=H8IaxrANWl.

Daniel McKenzie, Howard Heaton, Qiuwei Li, Samy Wu Fung, Stanley Osher, and Wotao Yin. Three-operator splitting for learning to predict equilibria in convex games. *SIAM Journal on Mathematics of Data Science*, 6(3):627–648, 2024b. doi: 10.1137/22M1544531.

Ian M. Mitchell, Alexandre M. Bayen, and Claire J. Tomlin. A time-dependent hamilton–jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005a. doi: 10.1109/TAC.2005.851439.

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

Yesom Park, Eric Gelphman, Stanley Osher, and Samy Wu Fung. Implicit neural optimal transport via fixed-point optimization. *arXiv preprint arXiv:2605.10792*, 2026.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

Hayden Schaeffer. Neural network approximation and structural complexity of semilinear pdes with lower bounds. *arXiv preprint arXiv:2610.40384*, 2026.

David SeWell, Xingjian Li, Stepan Tretiakov, Krishna Kumar, and David Fridovich-Keil. Neural operators for multi-task control and adaptation. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=jciOb0z5Wm.

William Sharpless, Zeyuan Feng, Somil Bansal, and Sylvia Herbert. Linear supervision for non-linear, high-dimensional neural control and differential games. In *Proceedings of the 7th Annual Learning for Dynamics and Control Conference*, volume 283 of *Proceedings of Machine Learning Research*, pages 365–377. PMLR, 2025.

Aditya Singh, Zeyuan Feng, and Somil Bansal. Exact imposition of safety boundary conditions in neural reachable tubes. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5489–5495. IEEE, 2025.

Ryan Teoh, Sander Tonkens, William Sharpless, Aijia Yang, Zeyuan Feng, Somil Bansal, and Sylvia Herbert. MADR: MPC-guided adversarial DeepReach. *arXiv preprint arXiv:2510.18845*, 2025.

Lieven Vandenberghe and Stephen Boyd. *Convex optimization*, volume 1. Cambridge university press Cambridge, 2004.

Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks. *arXiv preprint arXiv:2203.07404*, 2022.

Sifan Wang, Shyam Sankaran, Hanwen Wang, and Paris Perdikaris. An expert's guide to training physics-informed neural networks. *arXiv preprint arXiv:2308.08468*, 2023.

Appendix A. Related Work

We provide a review of the literature on neural HJ reachability and subsequent works that extend these methods to more complex and challenging problems. Additionally, we provide an overview of recent advances in methods for handling implicit Hamiltonians.

High-dimensional neural HJ reachability. DeepReach (Bansal and Tomlin, 2021) introduced a mesh-free neural approach for solving HJ reachability problems in dimensions beyond the range of conventional grid-based solvers. Rather than discretizing the state space, DeepReach directly parameterizes the value function and minimizes the HJI variational inequality residual. Subsequent approaches have improved the scalability and training of neural HJ solvers. Sharpless et al. (Sharpless et al., 2025) augment the HJI loss with supervision generated from tractable linearized differential games and demonstrate neural value-function learning for systems with tens of state dimensions. Similarly, Feng et al. (Feng et al., 2025) and MADR (Teoh et al., 2025) leverage MPC-based supervision to improve neural HJ reachability learning. These methods train the HJ value function end-to-end, but assume that the Hamiltonian required by the HJI residual can be evaluated efficiently. On the theoretical side, Hofgard (Hofgard, 2024) establishes conditional convergence guarantees for DeepReach, while Lin and Bansal (Lin and Bansal, 2023) provide formal safety assurances through error bounds and reachable-tube corrections. Related theoretical work also studies neural approximation of HJ-type PDEs (Schaeffer, 2026).

HJ reachability without closed-form Hamiltonians. Several recent works address HJ reachability problems in which the Hamiltonian is difficult to evaluate or unavailable in closed form. (Chilakamarri et al., 2025) learn a neural Hamiltonian surrogate from black-box dynamics samples for subsequent use in HJ solvers. (Choi et al., 2025) construct a conservative data-driven Hamiltonian directly from trajectory data, while (Lin et al., 2026) extend HJ-based verification to complex perception-based controllers under state uncertainty. These approaches make a difficult Hamiltonian tractable through approximation or bounding before it is used by the value function solver. In contrast, we retain the numerical optimization defining the Hamiltonian as an explicit component of neural HJI training and differentiate the value-function objective through its solution.

Implicit Hamiltonians in Control. Implicit optimization layers have been studied broadly in machine learning and differentiable optimization, with Jacobian-Free Backpropagation (JFB), which provides an inexpensive alternative to exact implicit differentiation (Fung et al., 2022; Heaton et al., 2021; Heaton and Fung, 2023; McKenzie et al., 2024a,b). In control, (Li et al., 2026b) combine operator splitting with JFB to enable scalable end-to-end training through embedded CBF-QP safety layers, which demonstrates that Jacobian-free differentiation can substantially reduce the computational cost of repeatedly differentiating through optimization problems along control trajectories. More closely related to our setting, (Gelphman et al., 2026a) develop an end-to-end method for high-dimensional optimal control in which the Hamiltonian minimization is solved implicitly and differentiated using JFB. Subsequent work establishes convergence guarantees for the resulting approximate-gradient training procedure (Gelphman et al., 2026b). Imitation-based function-encoder policies instead sidestep the Hamiltonian, transferring zero-shot across tasks (Li et al., 2026a). As shown in Section 3.1, the particular structure of the HJ reachability problem allows for exact gradient evaluation of the training objective without requiring approximate differentiation techniques, providing a more direct and natural alternative to existing approaches for handling implicit Hamiltonians.

Appendix B. Additional Experimental Details

This appendix provides supplementary problem formulations, Hamiltonian derivations, and implementation details for the numerical experiments presented in Section 4.

B.1. High-Dimensional Energy-Constrained Pursuit–Evasion

We extend the two-player pursuit–evasion problem in Section 4 to N independent evader-pursuer pairs, with joint state $x = (r_1, q_1, \ldots, r_N, q_N) \in \mathcal{X} \subset \mathbb{R}^{2N}$. For each pair i, r_i and q_i denote the relative distance and energy resource, respectively, with controls $u_i \in [0, \bar{u}_i]$ and $d_i \in [0, \bar{d}_i]$. The dynamics are

$$\begin{cases} \dot{r}_i = (1 + k_i^1 q_i)\tanh(u_i) - 1.2(1 - k_i^2 q_i)\tanh(d_i), \\ \dot{q}_i = -\alpha_i u_i^2 + \beta_i d_i^2, \end{cases} \qquad i = 1, \ldots, N, \tag{19}$$

where $k_i^1, k_i^2, \alpha_i, \beta_i > 0$ may vary across pairs to model heterogeneous maneuverability and energy consumption rate. The target set consists of states in which at least one pair violates its separation or energy threshold, and is represented by the terminal function

$$\ell(x) = \min_{i=1,\ldots,N} \left\{ r_i - r_i^{\text{reach}}, q_i - q_i^{\text{reach}} \right\}. \tag{20}$$

The corresponding target set is

$$\mathcal{L} = \bigcup_{i=1}^{N} \left(\{ r_i \le r_i^{\text{reach}} \} \cup \{ q_i \le q_i^{\text{reach}} \} \right).$$

Since the individual evader–pursuer pairs are dynamically decoupled, the Hamiltonian separates as $H(x,p) = \sum_{i=1}^{N} H_i(x_i, p_i)$, where $x_i = (r_i, q_i)$ and $p_i = [p_{r_i}, p_{q_i}]^\top$. Thus, the optimal control and disturbance can be computed independently for each pair using the same numerical optimization procedure as in the 2D case. We also note that under suitable k_i^1 and k_I^2 we can preserve the convex-concave structure of the Hamiltonian, satisfying the necessary assumptions for the theory.

As the problem dimension increases, the reachable set occupies a larger portion of the state space, leading to an imbalance between reachable and unreachable states under uniform sampling. To address this imbalance and stabilize training, we employ importance sampling to ensure that decreasing training loss leads to improved closed-loop performance.

B.2. Two-Vehicle Collision Avoidance

We consider a finite-horizon collision avoidance problem between two planar vehicles. The joint state is $x = (x_1, \ldots, x_6) \in \mathbb{R}^4 \times \mathbb{S}^1 \times \mathbb{S}^1$, where (x_1, x_2, x_3) and (x_4, x_5, x_6) denote the positions and headings of the pursuer and evader, respectively. We modify the standard constant-speed dynamics (Bansal and Tomlin, 2021) by introducing steering dependent speed for the evader:

$$\begin{cases} \dot{x}_1 = 0.75\cos x_3, & \dot{x}_4 = 0.75 e^{-0.4(u/3)^2}\cos x_6, \\ \dot{x}_2 = 0.75\sin x_3, & \dot{x}_5 = 0.75 e^{-0.4(u/3)^2}\sin x_6, \\ \dot{x}_3 = d, & \dot{x}_6 = u, \end{cases} \tag{21}$$

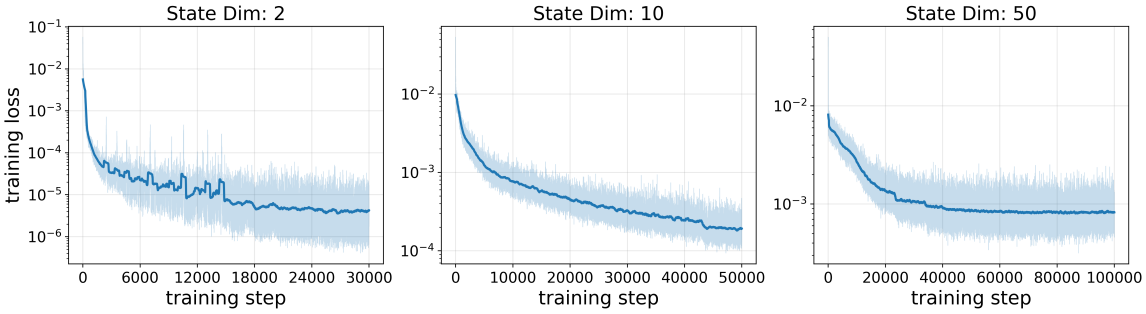


Figure 5: Training loss curves for the energy-constrained pursuit-evasion problems in 2D, 10D, and 50D. The losses decrease and stabilize across all three settings, demonstrating convergence of our proposed neural HJ training procedure. Higher-dimensional problems exhibit larger final residual losses.

where $u, d \in [-3, 3]$ denote the evader's and pursuer's turn-rate controls, respectively. The evader's speed decreases with the magnitude of its turn rate, introducing nonlinear control dependence into the Hamiltonian. For the time horizon $T = 1$, the capture target is defined by

$$\ell(x) = \sqrt{(x_1 - x_4)^2 + (x_2 - x_5)^2} - 0.25, \tag{22}$$

with $\mathcal{L} = \{x : \ell(x) \leq 0\}$. The corresponding Hamiltonian is

$$\begin{aligned} H(x,p) = {} & 0.75(p_1 \cos x_3 + p_2 \sin x_3) - 3|p_3| \\ & + \max_{u \in [-3,3]} \left[0.75 e^{-0.4(u/3)^2} (p_4 \cos x_6 + p_5 \sin x_6) + p_6 u \right], \end{aligned} \tag{23}$$

where $p = \nabla_x V(x,t)$. While the pursuer's optimal turn rate admits a closed-form solution, the evader's optimal control is determined through an inner optimization problem due to the nonlinear dependence of the Hamiltonian on its turn rate.

We further observe that the curvature of the evader's maximization objective is determined entirely by the sign of the term $(p_4 \cos x_6 + p_5 \sin x_6)$. Specifically, the objective is strongly concave for $p_4 \cos x_6 + p_5 \sin x_6 > 0$ and strongly convex for $p_4 \cos x_6 + p_5 \sin x_6 < 0$ over the admissible control set. Consequently, the global maximizer either lies on the domain boundary or can be provably recovered through methods such as projected gradient ascent. Finally, we use the computational domain

$$\mathcal{X} = [-1.5, 1.5]^2 \times [-\pi, \pi) \times [-1.5, 1.5]^2 \times [-\pi, \pi),$$

with state-time samples drawn uniformly during training.

B.3. Quadrotor Reach-Avoid Control under Adversarial Disturbances

We consider a six-dimensional planar quadrotor subject to adversarial wind disturbances, with dynamics and constraints specified below. Let $x = (x_1, x_2, x_3, x_4, x_5, x_6) \in \mathbb{R}^6$ denote the quadrotor's position, velocity, pitch angle and pitch rate, respectively. The two thrusters of the rotor can be controlled directly via $u \in U = [0, \bar{u}]^2$, the adversary chooses wind velocity $d \in D = \{d \in \mathbb{R}^2 :$

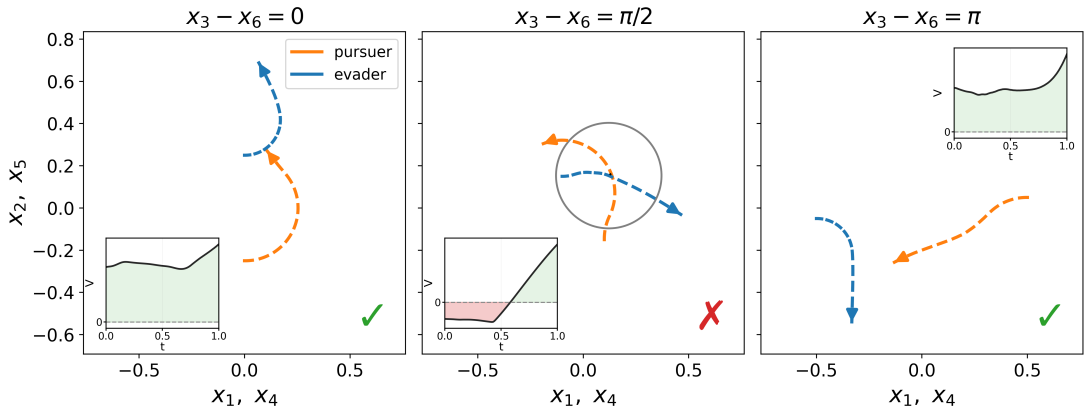


Figure 6: Closed-loop trajectories for the two-vehicle collision avoidance problem, using feedback controls obtained from the learned value function and online Hamiltonian optimization. **Left & Right:** successful evasion; **Middle:** capture. The learned value function is overlaid along each trajectory to illustrate its temporal evolution.

$\|d\|_2 \leq \bar{d}\}$. The quadrotor operates under standard dynamics as in (Mahony et al., 2012; SeWell et al., 2026), which are given by

$$\begin{cases} \dot{x}_1 &= x_3, \\ \dot{x}_2 &= x_4, \\ \dot{x}_3 &= \dfrac{-(u_1+u_2)\sin x_5 - k(x,d)(x_3-d_1)}{m}, \\ \dot{x}_4 &= \dfrac{(u_1+u_2)\cos x_5 - k(x,d)(x_4-d_2)}{m} - g_0, \\ \dot{x}_5 &= x_6, \\ \dot{x}_6 &= \dfrac{L}{I}(u_2-u_1). \end{cases}$$

Here $m = 1\,\mathrm{kg}$ is the mass, $I = 0.05\,\mathrm{kg\,m^2}$ is pitch inertia, $L = 0.2\,\mathrm{m}$ is the rotor's arm length, we use $g_0 = 9.81\,\mathrm{m/s^2}$ to denote gravity. We also set the maximum thrust to $\bar{u} = 7\,\mathrm{N}$ and the maximum wind speed to $\bar{d} = 3\,\mathrm{m/s}$. While the rotor dynamics are affine in control, the wind disturbance enters via a nonlinear drag force which we define here as

$$k(x,d) = c_1 + c_2\sqrt{(x_3-d_1)^2 + (x_4-d_2)^2}.$$

Here the force model combines linear and quadratic drag based on relative air velocity, while not the highest fidelity aerodynamics model, it is still commonly used for simulation and analysis (Hattenberger et al., 2023; Kwon and Eun, 2024). For our example we consider coefficients $c_1 = 0.5$ and $c_2 = 0.05$. The terminal time is set to $T = 2\mathrm{s}$. For simplicity, we consider a box region over which we sample and solve the resulting BRAT problem, which we define as $\Omega = [-2,2] \times [-0.5,4] \times [-3,3] \times [-3,3] \times [-1.2,1.2] \times [-4,4]$.

The reach target of the problem is a boxed region over x_1 and x_2 at $[0,1] \times [2,3]$, we define the reach and avoid set functions as

$$\begin{cases} \ell(x) & = \max\{-x_1, x_1 - 1, 2 - x_2, x_2 - 3\}, \\ g(x) & = \max\left\{-x_2,\ |x_5| - \frac{\pi}{3}\right\}, \end{cases}$$

and subsequently the target and unsafe sets as

$$\mathcal{L} = \{x : \ell(x) \leq 0\}, \quad \mathcal{G} = \{x : g(x) > 0\}$$

following the conventions established in Section 2. We note that the unsafe set encodes two failure scenarios: the first occurs when the quadrotor reaches the ground, and the second occurs when the pitch angle exceeds the prescribed safety limit, potentially leading to a loss of stability and a drone flip-over. We place no direct target restrictions on x_3, x_4, x_5, x_6, although a successful arrival at the target must nevertheless satisfy $g(x) \leq 0$. The optimal wind disturbance is determined by the constrained optimization problem

$$d^\star \in \arg\max_{\|d\|_2 \leq \bar{d}} -\frac{k(x,d)}{m}\Big[p_3(x_3 - d_1) + p_4(x_4 - d_2)\Big].$$

Under our choice of parameters, the maximization admits a unique solution for every state and adjoint, though obtaining $d^\star$ always requires an iterative solver, making this a challenging yet suitable testbed for our proposed approach.

Appendix C. Additional Numerical Results

C.1. Numerical Verification of The Theory

We provide numerical verification of our main theoretical results Section 3.1 here in this section. We mainly consider the the 2D energy constrained pursuit–evasion problem as the example to focus on. Our focus is to validate the accuracy of the training gradient $\nabla_\theta \mathcal{L}$ under our proposed approach in saddle point optimization. For simplicity we apply k steps of projected gradient ascent/descent with fixed step size of 0.1. Early stopping is disabled to ensure a fair comparison. We compare two different scenarios:

- **Gradient Free (ours):** the inner optimization is untracked and only the obtained saddle point is supplied to the loss function.
- **Automatic Differentiation (AD):** each inner optimization step is fully tracked and unrolled during backpropagation.

For value function construction we consider two different parameterizations: **(i)** a polynomial parameterization for the value function with sensitivity under control, and **(ii)** a neural network with the same size used in our numerical testing. For both cases we initialize the value functions so that the adjoints remain strictly positive,thus ensuring the min-max structure and a strictly interior saddle point. for each testing case, we sample with batch size 1024 uniformly over the domain.

We report gradient consistency results for both test cases under different solver iterations $k \in \{1, 2, 5, 10, \ldots, 500\}$ and display the results in Table 2. The gradient accuracy over the trainable

Table 2: Gradient consistency test under different value function parameterizations and solver accuracies. Here we define $\Delta\nabla_\theta\mathcal{L} := \nabla_\theta\mathcal{L}_{\text{ad}} - \nabla_\theta\mathcal{L}_{\text{ours}}$. All terms are in 2-norm unless otherwise specified. Gradient norm for the polynomial case is $\|\nabla_\theta\mathcal{L}\| = 14.0$ and for the neural network case $\|\nabla_\theta\mathcal{L}\| = 6.7$.

	Polynomial V_θ			Neural Network V_θ		
k	$\|\Delta\nabla_\theta\mathcal{L}\|$	$\|\partial_u\varphi\|_\infty$	$\|\partial_d\varphi\|_\infty$	$\|\Delta\nabla_\theta\mathcal{L}\|$	$\|\partial_u\varphi\|_\infty$	$\|\partial_d\varphi\|_\infty$
1	2.4×10^{-3}	2.7×10^{-1}	1.2×10^{-1}	2.0×10^{-4}	4.7×10^{-2}	4.2×10^{-2}
2	3.9×10^{-3}	2.1×10^{-1}	1.1×10^{-1}	3.9×10^{-4}	4.7×10^{-2}	4.2×10^{-2}
5	5.2×10^{-3}	1.5×10^{-1}	6.8×10^{-2}	9.1×10^{-4}	4.5×10^{-2}	4.0×10^{-2}
10	3.7×10^{-3}	7.8×10^{-2}	3.2×10^{-2}	1.6×10^{-3}	4.2×10^{-2}	3.8×10^{-2}
20	1.0×10^{-3}	2.2×10^{-2}	7.3×10^{-3}	2.6×10^{-3}	3.6×10^{-2}	3.3×10^{-2}
50	1.2×10^{-5}	4.4×10^{-4}	8.2×10^{-5}	3.3×10^{-3}	2.4×10^{-2}	2.3×10^{-2}
100	8.7×10^{-9}	6.7×10^{-7}	4.6×10^{-8}	2.2×10^{-3}	1.3×10^{-2}	1.4×10^{-2}
200	7.9×10^{-15}	1.6×10^{-12}	1.5×10^{-14}	5.5×10^{-4}	4.5×10^{-3}	5.4×10^{-3}
500	0	1.1×10^{-15}	1.1×10^{-15}	7.4×10^{-6}	2.1×10^{-4}	3.1×10^{-4}

weights is governed by the accuracy of the inner solve, and vanishes with it. As k increases, the Hamiltonian residuals $\|\partial_u\varphi\|_\infty$ and $\|\partial_d\varphi\|_\infty$ decrease, and $\|\Delta\nabla_\theta\mathcal{L}\|$ decreases with them. For the polynomial V_θ the solve converges to machine precision by $k \approx 200$, and the two gradients coincide exactly at $k = 500$. The neural network converges more slowly because under random initialization its adjoint p_θ is small and the Hamiltonian is only weakly curved in the controls. Nevertheless the gradient discrepancies are negligible at $k = 500$ for both cases especially regarding the $O(1)$ gradient norms. For cases where the saddle points lie on the boundaries, the results follow trivially. Specifically, when evaluated at the exact boundary, the gradient difference is exactly 0, since no fixed-point error needs to be accounted for. Overall the numerical results fully verifies our theoretical claims.

C.2. Inner Solver Comparison

The inner saddle point problem (5) is solved approximately at every training step. While our theory in Section 3.1 ensures gradient consistency at the fixed point, the quality of the truncated solve can still affect training. We hereby conduct an ablation study and compare between fixed and adaptive step size selection in the saddle point optimization under the same gradient descent rule. We test on the 10D energy constrained pursuit–evasion example, where both optimizers are given the same budget of 200 iterations. Two copies of the value network are trained under exactly the same sampling and with identical weight initialization; hence, any difference in training is due to the inner solver alone. We also keep a reference solution for the fixed point solver, which uses PGDA with a fixed step size of 0.1 and 5000 iterations. This achieves a median fixed point error of 1.6×10^{-6} at each sampling mini-batch and serves as a ground truth reference when comparing gradient accuracies. We run a total of 10000 training steps for the test.

We present the results in Table 3. Under the same iteration budget, adaptive step sizing yields a substantially more accurate inner solve than a carefully selected fixed step size. Importantly, the computational overhead is marginal, allowing us to maintain the fast training required for neural HJ reachability at large. However, we note that despite the lower solver accuracy, using plain PGDA

Table 3: Inner solver accuracy comparison between fixed and adaptive step size selection. Hamiltonian gap and gradient alignment measure relative value and gradient errors against the reference solution. Numbers averaged across 10K training steps except for the final loss.

	% of stationary samples*	Hamiltonian gap ↓	Gradient alignment ↑	Final Training loss ↓	Per training step solver time ↓
Fixed $\eta = 0.1$	22%	4.8%	0.965	8.55×10^{-4}	**0.214**s
Adaptive η	**98.9%**	**0.4%**	**0.999**	$\mathbf{8.30 \times 10^{-4}}$	0.247s

* we use tolerance at 1×10^{-3} to determine convergence.

with a fixed step size does not lead to significant errors in either the gradient updates or the overall training loss behavior. We attribute this to the stochasticity of Adam and mini-batch training in general. The key takeaway we want to highlight is that convergence of the value function can still be expected even when the inner solve admits a small error.

Appendix D. Training Setup and Details on Numerical Experiments

We provide details on the training setup used in this paper. We consider a total of five different experiments. In all examples the value function V_θ is defined per (14), with N_θ being a residual MLP with $\tanh$ activations and terminal condition enforced exactly by construction. Training only minimizes the VI residual at sampled locations. For all examples except for the quadrotor, we employ a uniform sampling scheme over the state-time space. We train all networks with Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 10^{-3}. The learning rate is halved whenever training hits a plateau, the patience for which we set differently for each example. The default solver for the inner optimization is PGDA with Barzilai–Borwein adaptive step sizing, as introduced in Section 3 with initial step size 0.1. We terminate early whenever the fixed-point residual reaches 1×10^{-4}. The maximum number of iterations for the inner optimization is set separately for each individual example. We try to balance between solver accuracy and speed during training. All code is implemented in PyTorch (Paszke et al., 2019), and all runs use random seed 0; problem-specific settings are listed in Table 4.

Appendix E. Proof of Theorem

For completeness, we restate the assumptions and main theorem from Section 3.1 and provide its detailed proof, together with the supporting lemmas.

Notations. Write $z = (u, d) \in \mathcal{U} \times \mathcal{D}$. We denote

$$p_\theta := \nabla_x V_\theta(x, t), \tag{24}$$

$$\varphi(p, u, d) = \varphi(p, z) := p^\top f(x, u, d) = p^\top f(x, z), \tag{25}$$

where we slightly abuse notations, and used z and (u, d) interchangeably, and thus

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

Table 4: Problem-specific training settings. Inner iterations are the maximum number of iterations of the inner solver.

Example	T	Training Steps	Batch Size	Network Size (Width, # of Layers)	LR patience	Inner Iters.
Energy Constrained Pursuit–Evasion (2D)	1.5	30k	256	(128, 3)	2000	100
Energy Constrained Pursuit–Evasion (10D)	1.5	50k	256	(128, 4)	3000	200
Energy Constrained Pursuit–Evasion (50D)	1.5	100k	256	(256, 4)	5000	500
Two-Vehicle Collision Avoidance	1.0	50k	1024	(128, 4)	3000	100
Planar Quadrotor (BRAT)	2.0	70k	1024	(256, 4)	3000	10*

* We use a problem-specific solver for the BRAT example based on global contraction, which converges faster and is more accurate than PGDA.

Assumptions. For an arbitrary parameter θ_0, we assume the following:

(A1) $\mathcal{U}$ and $\mathcal{D}$ are nonempty, compact and convex;

(A2) $f(x,\cdot)$ is C^1;

(A3) the mapping $\theta \mapsto p_\theta$ is differentiable at θ_0;

(A4) there exists $\delta_0 = \delta(p_{\theta_0}) > 0$ such that, for every $p \in \mathcal{B}_0 := \{p \in \mathbb{R}^n : \|p - p_{\theta_0}\|_2 < \delta_0\}$, the Isaacs condition

$$\max_{u\in\mathcal{U}} \min_{d\in\mathcal{D}} \varphi(p,u,d) = \min_{d\in\mathcal{D}} \max_{u\in\mathcal{U}} \varphi(p,u,d)$$

holds;

(A5) the function $\varphi(p_{\theta_0},\cdot,\cdot)$ has a unique saddle point on $\mathcal{U}\times\mathcal{D}$, which we denote by $z^\star(p_{\theta_0})$.

Theorem 1 *Under* (A1)–(A5)*, the mapping $\theta \mapsto H(x,p_\theta)$ is differentiable at θ_0, with*

$$\begin{aligned}\partial_\theta H(x,p_\theta)|_{\theta_0} &= f\big(x,z^\star(p_{\theta_0})\big)^\top (\partial_\theta p_\theta)|_{\theta_0} + \underbrace{\big[\nabla_z\big(p_{\theta_0}^\top f(x,z)\big)\big|_{z=z^\star(p_{\theta_0})}\big]^\top (\partial_\theta z^\star(p_\theta))\,|_{\theta_0}}_{0 \text{ contribution}} \\ &= f(x,z^\star(p_{\theta_0}))^\top (\partial_\theta p_\theta)|_{\theta_0}.\end{aligned} \tag{13}$$

In other words, the extra term that would result from chain rule does not appear. Thus, the saddle point can be treated as a constant during backpropagation, with no gradients propagated through $z^\star(p_\theta)$ or the solver that computes it.

Proof sketch: The key idea of our proof is to avoid differentiating through the saddle point solution map, which may be nondifferentiable. Instead, we directly analyze perturbations of the Hamiltonian using saddle point inequalities. Together with compactness and uniqueness, these bounds establish differentiability of the Hamiltonian and yield the desired gradient identity through the chain rule.

Proof Let $S(p)$ denote the set of saddle points at p, that is,

$$\mathcal{S}(p) := \left\{ z = (u, d) \in \mathcal{U} \times \mathcal{D} : \ \varphi(p, \tilde{u}, d) \leq \varphi(p, u, d) \leq \varphi(p, u, \tilde{d}) \quad \forall \tilde{u} \in \mathcal{U}, \ \tilde{d} \in \mathcal{D} \right\}. \quad (26)$$

Step 1: Existence of Saddle Point. By (A4), for every $p \in \mathcal{B}_0 := \{p \in \mathbb{R}^n : \|p - p_{\theta_0}\|_2 < \delta_0\}$, the Isaacs condition

$$\max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \varphi(p, u, d) = \min_{d \in \mathcal{D}} \max_{u \in \mathcal{U}} \varphi(p, u, d) =: v \quad (27)$$

holds. Here, the extrema are attained by compactness and continuity. Let $u^\star$ attain the outer maximum on the left and $d^\star$ the outer minimum on the right. On one hand,

$$\varphi(p, u^\star, d^\star) \geq \min_{d \in \mathcal{D}} \varphi(p, u^\star, d) = v,$$

where in the first step, we used the property of minimum, and in the second step, we used the definition of v in (27). On the other hand,

$$\varphi(p, u^\star, d^\star) \leq \max_{u \in \mathcal{U}} \varphi(p, u, d^\star) = v,$$

using the same argument. Combining these two inequalities, we have

$$\varphi(p, u^\star, d^\star) = v. \quad (28)$$

Next, we argue that $(u^\star, d^\star)$ is a saddle point. On one hand, we have, for every $\tilde{u} \in \mathcal{U}$,

$$\varphi(p, \tilde{u}, d^\star) \ \leq \ \max_{u \in \mathcal{U}} \varphi(p, u, d^\star) \ = \ v \ = \ \varphi(p, u^\star, d^\star), \quad (29)$$

by the definition of $d^\star$. On the other hand, similarly, for every $\tilde{d} \in \mathcal{D}$,

$$\varphi(p, u^\star, \tilde{d}) \ \geq \ \min_{d \in \mathcal{D}} \varphi(p, u^\star, d) \ = \ v \ = \ \varphi(p, u^\star, d^\star), \quad (30)$$

by the definition of $u^\star$. Together, (29) and (30) establish that $(u^\star, d^\star)$ is a saddle point, that is $(u^\star, d^\star) \in \mathcal{S}(p)$. Thus, $\mathcal{S}(p) \neq \emptyset$ for every $p \in \mathcal{B}_0$, which is the existence of saddle point. Moreover, by the definition of the Hamiltonian H, (28) becomes

$$H(x, p) = \varphi(p, u^\star, d^\star). \quad (31)$$

Step 2: Linearization Error Bound. We bound the error of linearizing H around p_{θ_0}. Given any fixed $p' \in \mathcal{B}_0$, write $(u_0, d_0) := z^\star(p_{\theta_0})$ and let $(u', d') \in \mathcal{S}(p')$ be any saddle point at p' (which exists by Step 1).

Upper Bound. Using (SV) of Lemma 2, we have

$$H(x, p') \leq \varphi(p', u', d_0),$$
$$H(x, p_{\theta_0}) \geq \varphi(p_{\theta_0}, u', d_0).$$

Subtracting these two inequalities, we have

$$H(x, p') - H(x, p_{\theta_0}) \leq (p' - p_{\theta_0})^\top f(x, u', d_0). \quad (32)$$

Lower Bound. Similarly, using (SV) of Lemma 2, we have

$$H(x,p') \geq \varphi(p',u_0,d'),$$
$$H(x,p_{\theta_0}) \leq \varphi(p_{\theta_0},u_0,d').$$

Subtracting these two inequalities, we have

$$H(x,p') - H(x,p_{\theta_0}) \geq (p'-p_{\theta_0})^\top f(x,u_0,d'). \tag{33}$$

Error Bound. Let the linearization error be

$$R(p') := H(x,p') - H(x,p_{\theta_0}) - (p'-p_{\theta_0})^\top f(x,u_0,d_0). \tag{34}$$

Plugging (32) and (33) into (34), we have

$$(p'-p_{\theta_0})^\top (f(x,u_0,d') - f(x,u_0,d_0)) \leq R(p') \leq (p'-p_{\theta_0})^\top (f(x,u',d_0) - f(x,u_0,d_0)). \tag{35}$$

Next, we establish the linearization error bound as

$$\left| H(x,p') - H(x,p_{\theta_0}) - (p'-p_{\theta_0})^\top f(x,z^\star(p_{\theta_0})) \right| = |R(p')| \tag{36}$$

by the definition (34), and $z^\star(p_{\theta_0}) = (u_0,d_0)$, and

$$|R(p')| \leq \left\|p'-p_{\theta_0}\right\|_2 \omega(p') \tag{37}$$

by applying the Cauchy–Schwarz inequality to (35), where

$$\omega(p') = \max\left\{ \left\|f(x,u_0,d') - f(x,u_0,d_0)\right\|_2, \left\|f(x,u',d_0) - f(x,u_0,d_0)\right\|_2 \right\}. \tag{38}$$

Step 3: Differentiability of H. We prove that $H(x,\cdot)$ is differentiable at p_{θ_0}. By Lemma 3, as $\mathcal{B}_0 \ni p' \to p_{\theta_0}$, then for any choice of saddle point $(u',d') \in \mathcal{S}(p')$ (which exists by Step 1), $(u',d') \to (u_0,d_0)$. By (A2), $f(x,\cdot)$ is continuous, thus $\omega(p') \to 0$ as $p' \to p_{\theta_0}$. Therefore,

$$|R(p')| / \left\|p'-p_{\theta_0}\right\|_2 \leq \omega(p') \to 0$$

as $p' \to p_{\theta_0}$. Since $R(p')$ does not depend on the choice of the saddle point (u',d'), this implies that $H(x,\cdot)$ is differentiable at p_{θ_0}, and $\nabla_p H(x,p)\big|_{p_{\theta_0}} = f(x,z^\star(p_{\theta_0}))$.

Step 4: Gradient-free. By (A3), p_θ is differentiable at θ_0, thus by Step 3 and the chain rule,

$$\partial_\theta H(x,p_\theta)|_{\theta_0} = \left(\nabla_p H(x,p)\big|_{p_{\theta_0}}\right)^\top (\partial_\theta p_\theta)|_{\theta_0} = f(x,z^\star(p_{\theta_0}))^\top (\partial_\theta p_\theta)|_{\theta_0}.$$

■

Supporting Lemmas.

Lemma 2 (Saddle Value) *(Vandenberghe and Boyd, 2004, Section 5.4.2) If $(u^\star, d^\star)$ is a saddle point at p, then $H(x,p) = \varphi(p, u^\star, d^\star)$ and*

$$\varphi(p, u, d^\star) \;\le\; H(x,p) \;\le\; \varphi(p, u^\star, d), \qquad \forall\, u \in \mathcal{U},\ d \in \mathcal{D}. \tag{SV}$$

Lemma 3 (Continuity of saddle points at p_{θ_0}) *Let $S(p)$ denote the set of saddle points at p, that is,*

$$\mathcal{S}(p) := \left\{ z = (u,d) \in \mathcal{U} \times \mathcal{D} :\ \varphi(p, \tilde{u}, d) \le \varphi(p, u, d) \le \varphi(p, u, \tilde{d}) \quad \forall \tilde{u} \in \mathcal{U},\ \tilde{d} \in \mathcal{D} \right\}. \tag{26}$$

Under (A1), (A2), and (A5), if the sequence $p_k \to p_{\theta_0}$ with $p_k \in \mathcal{B}_0$ (defined in (A4)) and $z_k \in \mathcal{S}(p_k)$, then $z_k \to z^\star(p_{\theta_0})$.

Proof Write $z_0 := z^\star(p_{\theta_0})$ and $z_k = (u_k, d_k)$ for all k. Suppose the contrary, that is, there exists $\epsilon > 0$ such that there is a subsequence $\{z_{k'}\}$ with $\|z_{k'} - z_0\|_2 \ge \epsilon$ for all k'.

By (A1), since $\mathcal{U} \times \mathcal{D}$ is compact, by the Bolzano-Weierstrass theorem (Bartle and Sherbert, 2000, Theorem 3.4.8), there exists a convergent subsequence $\{z_{k''}\}$ of $\{z_{k'}\}$ such that $z_{k''} \to \bar{z} = (\bar{u}, \bar{d}) \in \mathcal{U} \times \mathcal{D}$ and

$$\|\bar{z} - z_0\|_2 \ge \epsilon, \tag{39}$$

by continuity of $\|\cdot\|_2$. Also, remark that $p_{k''} \to p_{\theta_0}$ by assumption. Since $z_{k''} = (u_{k''}, d_{k''}) \in \mathcal{S}(p_{k''})$, we have (Vandenberghe and Boyd, 2004, Section 5.4.2)

$$\varphi(p_{k''}, \tilde{u}, d_{k''}) \le \varphi(p_{k''}, u_{k''}, d_{k''}) \le \varphi(p_{k''}, u_{k''}, \tilde{d}), \qquad \forall\, \tilde{u} \in \mathcal{U},\ \tilde{d} \in \mathcal{D}. \tag{40}$$

Consider $\varphi(p,u,d) = p^\top f(x,u,d)$. By (A2), $f(x,\cdot)$ is continuous on $\mathcal{U} \times \mathcal{D}$. Together with continuity of inner product, they imply φ is jointly continuous in (p,u,d). Thus, taking $k'' \to \infty$ gives

$$\varphi(p_{\theta_0}, \tilde{u}, \bar{d}) \le \varphi(p_{\theta_0}, \bar{u}, \bar{d}) \le \varphi(p_{\theta_0}, \bar{u}, \tilde{d}), \qquad \forall\, \tilde{u} \in \mathcal{U},\ \tilde{d} \in \mathcal{D}.$$

This implies that $\bar{z} = (\bar{u}, \bar{d})$ is a saddle point at p_{θ_0} (Vandenberghe and Boyd, 2004, Section 5.4.2). On the other hand, by (A5), $\mathcal{S}(p_{\theta_0}) = \{z_0\}$ is a singleton. Thus, we have $\bar{z} \in \mathcal{S}(p_{\theta_0}) = \{z_0\}$, but this contradicts (39) and thus contradicts our original assumption. ■