Convergence Guarantees for Neural Network-Based Hamilton–Jacobi Reachability

William Hofgard
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
whofgard@stanford.edu

Abstract

We provide a novel uniform convergence guarantee for DeepReach, a deep learning-based method for solving Hamilton–Jacobi–Isaacs (HJI) equations arising in reachability analysis. Specifically, we show that the DeepReach algorithm, as introduced by Bansal et al. in their eponymous paper from 2020, is *stable* in the sense that if the loss functional for the algorithm converges to zero, then the resulting neural network approximation converges uniformly to the classical solution of the HJI equation, assuming that a classical solution exists. We also provide numerical tests of the algorithm, replicating the experiments provided in the original DeepReach paper and examining the impact that our technical modifications of the algorithm have on its empirical performance.

1 Introduction

In optimal control theory, Hamilton–Jacobi (HJ) reachability analysis describes states from which trajectories will eventually reach a specified set under an optimal control policy [1]. The set of states from which trajectories will eventually enter the specified set is typically referred to as the *backward reachable tube* (BRT). Computing BRTs accurately is of particular interest when designing autonomous systems that must avoid collisions or otherwise unsafe configurations (e.g., autonomous vehicles, aircraft, and many other pertinent, real-world examples) [2].

HJ reachability analysis requires solving an associated Hamilton–Jacobi–Isaacs (HJI) equation, a nonlinear, first-order variational partial differential equation (PDE) [1]. The HJI equation characterizes the value function for the reachability problem, and the sublevel sets of the value function determine the BRT. In low dimensions, the HJI equation can be solved relatively easily using a grid-based PDE solver. In high dimensions, recent work confronts the so-called "curse of dimensionality" by instead using parametrized neural networks to solve high-dimensional PDEs [3, 4]. For instance, Bansal et al. developed DeepReach, a deep learning-based solver for the HJI equation that determines approximate BRTs for high-dimensional optimal control problems [2]. However, autonomous systems require robust safety guarantees, and it is often difficult to quantify the approximation error made by neural networks when solving high-dimensional PDEs that lack analytical solutions [5].

We approach this problem from the perspective of PDE theory, aiming to provide a direct uniform convergence guarantee for DeepReach. Building upon the work of [6, 7, 8], we show that, under broad technical assumptions, if the neural network loss for DeepReach approaches zero, then the resulting value function will converge *uniformly* to the true value function, which solves the HJI equation. We also include numerical experiments, extending the results in [5] to our technical setting.

D3S3: Data-driven and Differentiable Simulations, Surrogates, and Solvers @ NeurIPS 2024.

2 Related Work

This paper draws upon work done in two areas: deep learning methods for HJ reachability and more general convergence guarantees for deep learning-based PDE solvers. In the latter category, relevant work can be found in [3, 6, 7, 8], which consider solving first-order, nonlinear PDEs using parametrized neural networks. The theory of viscosity solutions, applied in [6] and discussed in more detail in [1] and [9], is directly applicable to HJI equations. This line of work is closely related to physics-informed neural networks (PINNs) [10]. In terms of deep learning methods for HJ reachability, we primarily build upon the work done by Bansal et al. in DeepReach [2]. DeepReach's formulation is analogous to the deep Galerkin method (DGM) first presented in [3]. Consequently, much of the uniform convergence analysis for high-dimensional HJB equations from [3, 6, 7] applies naturally to DeepReach, with several important technical modifications outlined in Appendix A.

Since the release of DeepReach in 2020, additional analysis of the method's approximation error has been carried out. For instance, in [5], the authors propose a technique labeled *scenario optimization*, in which states near the boundary of the approximate BRT are sampled randomly in order to expand the approximate BRT. In [11], which builds upon the scenario optimization approach of [5], the same authors instead utilize conformal prediction to establish confidence bounds. Finally, [12] analyzes the impact that different activation functions have, while [13] restructures the training loss to ensure that the HJI equation's terminal condition is always satisfied, often obtaining improved results.

3 Mathematical Background

We consider the general formulation of HJ reachability presented in [1, 2]. In particular, the dynamics of an autonomous agent in an environment are modeled by a state $x \in \mathbb{R}^n$, a control $u \in \mathcal{U} \subseteq \mathbb{R}^m$, and a disturbance $d \in \mathcal{D} \subseteq \mathbb{R}^m$, with $\dot{x} = f(x, u, d)$. Denote by $\xi_{x,t}^{u,d}(\tau)$ the state at time τ, under the control $u(\cdot)$ and disturbance $d(\cdot)$, starting at time t and state x. Denoting the safe set of states by $\mathcal{L}$, define the BRT on the time interval $[t, T]$, under *worst-case* disturbances $d(\cdot)$, by

$$\mathcal{V}_{\text{BRT}}(t) := \{x \in \mathbb{R}^n : \forall u(\cdot), \exists d(\cdot), \exists \tau \in [t, T] \text{ such that } \xi_{x,t}^{u,d}(\tau) \in \mathcal{L}\}. \tag{1}$$

From the above formulation of BRTs, we can present the corresponding HJI equations, again based on the discussion from [1]. Given a target set $\mathcal{L}$, define a function $\ell : \mathbb{R}^n \to \mathbb{R}$ such that $\mathcal{L} = \{x \in \mathbb{R}^n : \ell(x) \leq 0\}$. Define a cost functional, starting at initial state x and initial time t, and the corresponding value function by

$$J(x, t, u(\cdot), d(\cdot)) = \min{}_{\tau \in [t,T]} \ell(\xi_{x,t}^{u,d}(\tau)), \quad V(t, x) := \inf{}_{d(\cdot)} \sup{}_{u(\cdot)} \{J(x, t, u(\cdot), d(\cdot))\}. \tag{2}$$

In the case that $\mathcal{L}$ is an unsafe set that the agent wishes to avoid for all time t, the agent first selects an optimal control that maximizes ℓ, and the disturbance (viewed as an adversarial player in a zero-sum, two-player game) selects the input that minimizes ℓ. Via a standard dynamic programming argument, presented in [1], one obtains the HJI variational equation for the reachability problem, given by

$$\begin{aligned} &\min\{\partial_t V(t, x) + H(t, x), \ell(x) - V(t, x)\} = 0, \\ &V(T, x) = \ell(x). \end{aligned} \tag{3}$$

Above, the Hamiltonian $H : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ is

$$H(t, x) := \sup{}_u \inf{}_d \langle \nabla_x V(t, x), f(x, u, d) \rangle. \tag{4}$$

The BRT itself is computed as a sublevel set of the solution V to the HJI equation, with

$$\mathcal{V}_{\text{BRT}}(t) := \{x : V(t, x) \leq 0\}, \quad u^\star(t, x) := \arg\max{}_{u(\cdot)} \min{}_{d(\cdot)} \langle \nabla_x V(t, x), f(x, u, d) \rangle.$$

To solve the Equation (3) numerically, we utilize a parametrized neural network $V_\theta : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$, with parameters $\theta \in \mathbb{R}^P$. At this point, the architecture of the neural network remains unspecified, although the original implementation of DeepReach and most subsequent related work utilize fully-connected, feedforward neural networks with sinusoidal activations between layers [2]. In [2], a standard loss functional based on the L^1 or L^2-norm is used. Below, we make a slight modification, instead using *sup-norm loss* for both our theoretical guarantees and numerical experiments:

$$L(\theta) := \|V_\theta(T, x) - \ell(x)\|_\infty + \lambda \| \min\{\partial_t V_\theta(t, x) + H_\theta(t, x), \ell(x) - V_\theta(t, x)\}\|_\infty, \tag{5}$$

where, in practice, $(t, x) \in \mathbb{R} \times \mathbb{R}^n$ are sampled points, and H_θ is the Hamiltonian associated with the approximate value function V_θ, defined as in Equation (4). The DeepReach training procedure then mimics stochastic gradient descent (SGD). The neural network is first trained with $\lambda = 0$ to learn the terminal condition of the PDE, at $t = T$. After a pre-determined number of epochs, t is then linearly decreased from $t = T$ to $t = 0$, sampling K points $\{x_k\}_{k=1}^K$ at each step and performing an SGD step on the neural network parameters θ to learn the solution to Equation (3).

Remark 3.1. In the original DeepReach algorithm, the above loss is defined with respect to the ℓ^1-norm or ℓ^2-norm. However, recent work surrounding neural network solutions to second-order HJB equations indicates that such a choice of loss may not result in convergence to the true value function [8]. Instead, using a sup-norm loss metric brings both theoretical benefits, as seen in the convergence proof in Appendix A, and practical benefits, as seen in Section 5 below. As noted above, the sup-norms involved in Equation (5) are approximated by sampling in practice: [6] and [7] discuss convergence guarantees for this sampling-based approach, which still tend to exhibit some dependence on the dimensionality of the underlying problem. For our theoretical guarantees, we assume that the sup-norm loss in Equation (5) can be approximated to arbitrary accuracy.

4 Theoretical Guarantees

We provide two results concerning the convergence of DeepReach for solving the HJI equation in Equation (3). All technical results are presented in full, including detailed proofs, in Appendix A. First, we obtain an *existence* result as a relatively straightforward consequence of the universal approximation power of neural networks. Specifically, neural networks with smooth, bounded, non-constant activations (as is the case in the sinusoidal implementation of DeepReach) can approximate functions and their derivatives *arbitrarily* well, as in [14, Theorem 3]. Under the assumption that the value function V that solves Equation (3) is locally-Lipschitz with locally-Lipschitz gradients, denoted by $V \in \mathcal{C}^{1,1}_{\text{loc}}([0, T] \times \mathbb{R}^n)$, we prove the following:

Theorem 4.1. *For every $\varepsilon > 0$, there exists a constant $C > 0$ that depends only on the dynamics $\dot{x} = f(x, u, d)$ of the underlying reachability problem such that for some $\theta \in \mathbb{R}^P$, the DeepReach loss in Equation (5) satisfies $L(\theta) \leq C\varepsilon$.*

We also prove the following *uniform convergence* result, which is far more practical in nature and does not rely upon a non-constructive universal approximation theorem. All technical assumptions are presented in detail in Appendix A.

Theorem 4.2. *Under Assumptions (A) and (B), if a sequence of parameters $\{\theta^{(k)}\}_{k\in\mathbb{N}}$ is such that $L(\theta^{(k)}) \to 0$ as $k \to \infty$, then for any compact set $K \subset \mathbb{R}^n$,*

$$\lim_{k\to\infty} \sup{}_{(t,x)\in[0,T]\times K} |V_{\theta^{(k)}}(t, x) - V(t, x)| = 0.$$

The proof of Theorem 4.2 relies on tools from the theory of viscosity solutions. Specifically, by establishing that a sequence of neural network approximators are viscosity solutions to a sequence of perturbed HJI equations and applying an appropriate comparison principle to two suitably-defined upper and lower limits of the neural network approximators, one may obtain the uniform convergence guarantee in Theorem 4.2. Finally, we note that Theorem 4.2 is agnostic towards the choice of optimization algorithm for minimizing the DeepReach loss. In particular, stochastic gradient descent (or its common variants) need not be used in training for the guarantee to hold.

5 Numerical Experiments

In this section, we provide numerical experiments that demonstrate the performance of DeepReach for the collision avoidance problem from Appendix B. This problem, first presented in [15], is a standard two-vehicle, three-dimensional reachability problem. All code for reproducing results and figures can be found this notebook. More information on the sampling and training procedures used in both the original implementation of DeepReach and the modification presented in this paper can be found in Appendix C. We utilize the same neural network architecture and training procedure as in the open-source, PyTorch-based implementation provided by [2] under the MIT License. Below, we empirically investigate the impact that fine-tuning with sup-norm loss can have on the accuracy of the approximate BRT, lending credence to the theoretical convergence guarantee from Section 4.

The first two figures below correspond to slices of the three-dimensional collision avoidance problem, evaluated at time $t = 0.7$ and on the box $(x_1, x_2) \in [-1, 1]^2$. In particular, the coordinates (x_1, x_2, θ) are the *relative* coordinates between two agents. The true value function and BRT are computed analytically as in [15]. For $\theta = \pi$, which represents the two agents facing towards each other in relative coordinates, the pre-trained DeepReach approximation deteriorates in quality; see Figure 1. However, after fine-tuning the model with sup-norm loss as defined in Equation (5), the DeepReach BRT more closely approximates the true BRT for $\theta = \pi$, as seen in Figure 2, measured in terms of the maximum distance between the boundary of the approximate BRT and the true BRT. After only 1K epochs (approximately 5 minutes on an L4 Tensor Core GPU) of training, this fine-tuning procedure exhibits noticeable improvements over models pre-trained for 100K epochs (approximately 16 hours on an L4 GPU) with ℓ^1-loss, both in terms of the sampled loss metric *and* the true maximum approximation error, as seen in Figure 3 and Figure 4 respectively. Note that Figure 3 and Figure 4 are for demonstrative purposes only, as the training loss depends stochastically on the sampled training points. Nonetheless, the trends in both figures are reproducible across trials.

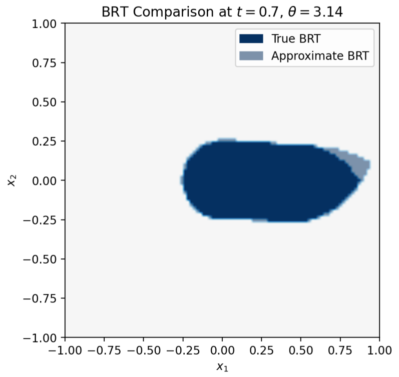

Figure 1: BRT comparison, using pre-trained model from [2].

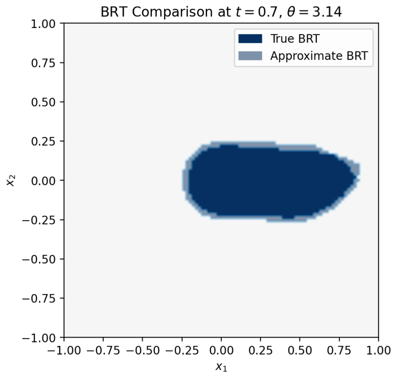

Figure 2: BRT comparison, using fine-tuned model with sup-norm loss.

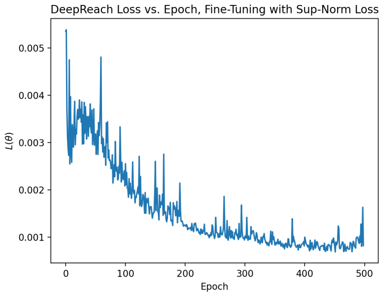

Figure 3: Fine-tuning with sup-norm loss results in significant decreases in DeepReach loss over just 500 epochs.

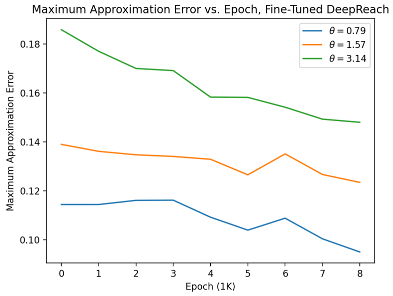

Figure 4: Maximum approximation error continues to decrease over 8K epochs, especially for $\theta = \pi$.

6 Conclusions and Future Work

We provide two novel convergence guarantees for a variant of the DeepReach algorithm, first introduced in [2]. In particular, by using a sup-norm loss metric, we show that the DeepReach loss can be made arbitrarily small by neural network approximators (Theorem 4.1) before showing that *any* sequence of neural networks that takes the DeepReach loss to zero must converge uniformly to the true solution to the HJI equation in question (Theorem 4.2). We then demonstrate this convergence result empirically, showing that by using the sup-norm loss metric, pre-trained DeepReach models can be fine-tuned to produce more accurate BRTs in a collision avoidance setting. Future work may investigate whether Theorem 4.2 can be extended to HJI equations that do *not* admit classical solutions. In terms of implementation, improving the performance of DeepReach via *adversarial training*, a common technique for training neural networks with sup-norm loss, is possible [8]. Experimenting with more expressive neural network architectures may also yield better approximations.

References

[1] Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton–Jacobi Reachability: A Brief Overview and Recent Advances. In 2017 IEEE 56th Annual Conference on Decision and Control (CDC), pages 2242–2253. IEEE, 2017.

[2] Somil Bansal and Claire Tomlin. DeepReach: A Deep Learning Approach to High-Dimensional Reachability. In IEEE International Conference on Robotics and Automation (ICRA), 2021.

[3] Justin Sirignano and Konstantinos Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. Journal of Computational Physics, 375:1339–1364, 2018.

[4] Weinan E, Jiequn Han, and Arnulf Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Commun. Math. Stat., 5(4):349–380, 2017.

[5] Albert Lin and Somil Bansal. Generating Formal Safety Assurances for High-Dimensional Reachability, 2023.

[6] William Hofgard, Jingruo Sun, and Asaf Cohen. Convergence of the Deep Galerkin Method for Mean Field Control Problems, 2024.

[7] Asaf Cohen, Mathieu Laurière, and Ethan Zell. Deep Backward and Galerkin Methods for the Finite State Master Equation, 2024.

[8] Chuwei Wang, Shanda Li, Di He, and Liwei Wang. Is L^2 Physics Informed Loss Always Suitable for Training Physics Informed Neural Network? In Advances in Neural Information Processing Systems, volume 35, pages 8278–8290. Curran Associates, Inc., 2022.

[9] L. C. Evans and P. E. Souganidis. Differential Games and Representation Formulas for Solutions of Hamilton-Jacobi-Isaacs Equations. Indiana University Mathematics Journal, 33(5):773–797, 1984.

[10] M. Raissi, P. Perdikaris, and G.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, February 2019.

[11] Albert Lin and Somil Bansal. Verification of Neural Reachable Tubes via Scenario Optimization and Conformal Prediction, 2024.

[12] Qian Wang and Tianhao Wu. Enhancing the Performance of DeepReach on High-Dimensional Systems through Optimizing Activation Functions, 2023.

[13] Aditya Singh, Zeyuan Feng, and Somil Bansal. Imposing Exact Safety Specifications in Neural Reachable Tubes, 2024.

[14] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. Neural Networks, 4(2):251–257, 1991.

[15] Ian Mitchell. A Robust Controlled Backward Reach Tube with (Almost) Analytic Solution for Two Dubins Cars. In Goran Frehse and Matthias Althoff, editors, ARCH20. 7th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH20), volume 74 of EPiC Series in Computing, pages 242–258. EasyChair, 2020.

[16] Michael G. Crandall, Hitoshi Ishii, and Pierre-Louis Lions. User's guide to viscosity solutions of second order partial differential equations. Bull. Amer. Math. Soc. (N.S.), 27(1):1–67, 1992.

A Technical Proofs

In this section, we provide proofs of Theorem 4.1 and Theorem 4.2 respectively. First, some additional notation is necessary. Consider an instantiation of Equation (3) on $\mathbb{R} \times \mathbb{R}^n$. Elements of the relevant class of fully-connected neural networks, as considered in [2, 13] take the form of a network with L layers, maximum width n, and a common activation function σ:

$$V_\theta(t, x) := \sigma(W_L \dots \sigma(W_1 x + \alpha t + c_1) \dots + c_L), \tag{6}$$

where the activation function σ is applied elementwise. Above, W_i are weight matrices, c_i are bias vectors, and α is a scalar weight. In turn, the parameters of each neural network are of the form $\theta = (W_1, \dots, W_L, c_1, \dots, c_L, \alpha) \in \mathbb{R}^P$ (upon flattening all weight matrices into vectors), where P depends on the maximum width of the network, the depth L of the network, and the dimension n of the target HJI equation. In turn, we take $\mathfrak{C}_{n+1}^{(P)}(\sigma)$ to be the class of neural networks with parameters θ of dimension at most P (but any number of layers L), from which we define $\mathfrak{C}_{n+1}(\sigma)$ as the class of neural networks with parameters θ of any dimension (i.e., allowing networks with unbounded width).

Additionally, given any compact subset $K \subset \mathbb{R}^n$, define the standard norm on $\mathcal{C}^1(K)$, the space of continuously-differentiable functions $f : K \to \mathbb{R}$, given by

$$\|f\|_{\mathcal{C}^1(K)} := \sup_{x \in K} |f(x)| + \sup_{x \in K} \|\nabla f(x)\|_2$$

Finally, define an operator by

$$\mathcal{L}[V](t, x) := \min\{\partial_t V(t, x) + H(t, x), \ell(x) - V(t, x)\},$$

observing that the first line of Equation (3) becomes $\mathcal{L}[V](t, x) = 0$. Throughout this section, we also impose the following mild technical assumptions.

Assumption A. *Given an underlying reachability problem, with dynamics $\dot{x} = f(x, u, d)$, we assume that:*

(1) There exist compact sets $\mathcal{U} \subseteq \mathbb{R}^m$ and $\mathcal{D} \subseteq \mathbb{R}^m$ such that all control inputs u and disturbances d satisfy $u \in \mathcal{U}$ and $d \in \mathcal{D}$.

(2) There exists a compact set $K \subseteq \mathbb{R}^n$ such that for any initial state $x(0) \in K$ and sequence of optimal control inputs and disturbances $u \in \mathcal{U}$, $d \in \mathcal{D}$, the state trajectory satisfies $x(t) \in K$ for all $t \in [0, T]$.

(3) There exists a constant $C_f > 0$ such that for any $x \in K$, $\|f(x, u, d)\|_2 \leq C_f$ for all $(u, d) \in \mathcal{U} \times \mathcal{D}$.

For instance, these standard assumptions all hold for the collision avoidance example introduced in Section 3. By scaling all states to lie in the d-dimensional box $[-1, 1]^d$ in Section 5 above, we ensure that the compactness assumption above is always met. If dynamics $f(x, u, d)$ are continuous, then the fact that all trajectories lie in some compact set K immediately implies the boundedness assumption above. Effectively, we require bounded state trajectories, bounded control inputs and disturbances, and bounded dynamics.

The next assumption, placed on Equation (3), is discussed in detail in [1] and [9]. Under a wide variety of circumstances, Equation (3) will indeed admit unique solutions.

Assumption B. *There exists a unique, classical solution $V \in \mathcal{C}^{1,1}(\Omega_K)$ to Equation (3). Furthermore, V is also the unique viscosity solution to Equation (3), as defined in Definition A.3 below.*

At this point, we note that certain HJI equations *only* admit viscosity solutions, the appropriate definition of a weak solution for first-order, nonlinear PDEs that resemble the HJI equation. The framework for establishing neural network approximation and convergence guarantees for equations that admit viscosity solutions [6], however, does not yet extend to the case of equations that do not admit unique classical solutions. In many cases, however, including (again) the collision avoidance example in Section 3, this is not a concern.

Now, we prove Theorem 4.1, relying on the approximation guarantees provided by [14, Theorem 3]. To make the proof more palatable, we construct several technical lemmas.

Lemma A.1. *Let $V \in \mathcal{C}^{1,1}(\Omega_K)$ solve Equation (3) on $[0,T] \times K$, for some compact set $K \subset \mathbb{R}^n$. Then, for any $\varepsilon > 0$ and bounded, non-constant activation $\sigma : \mathbb{R} \to \mathbb{R}$, there exists $V_\theta \in \mathfrak{C}_{n+1}(\sigma)$ such that*

$$\begin{aligned}\varepsilon > \sup_{(t,x)\in\Omega_K} |V(t,x) - V_\theta(t,x)| + \sup_{(t,x)\in\Omega_K} |\partial_t V(t,x) - \partial_t V_\theta(t,x)| \\ + \sup_{(t,x)\in\Omega_K} \|\nabla_x V(t,x) - \nabla_x V_\theta(t,x)\|_2.\end{aligned} \tag{7}$$

Proof. This is a direct consequence of the universal approximation theorem for neural networks, stated in Proposition [14, Theorem 3]. In particular, the result therein states that if $\sigma \in \mathcal{C}^m(\mathbb{R})$ is a non-constant and bounded activation, then $\mathfrak{C}_{n+1}$ is uniformly m-dense on compact sets in $\mathcal{C}^m(\mathbb{R}^{n+1})$. In particular, for all $h \in \mathcal{C}^m(\mathbb{R}^{n+1})$, all compact subsets $K \subset \mathbb{R}^{n+1}$, and any $\varepsilon > 0$, there exists $\psi \in \mathfrak{C}_{n+1}$ such that $\|h - \psi\|_{\mathcal{C}^m(K)} < \varepsilon$. Taking $m = 1$, $h = V$, and $\psi = V_\theta$ suffices to prove the lemma. □

Equipped with the preceding lemma, the following estimates will allow us to prove Theorem 4.1.

Lemma A.2. *Assume that $\varepsilon > 0$ and V_θ are as in Lemma A.1 and*

$$H_\theta(t,x) := \sup_u \inf_d \langle \nabla_x V_\theta(t,x), f(x,u,d)\rangle.$$

Then, there exists $C_f > 0$ depending only on the dynamics f such that

$$|H_\theta(t,x) - H(t,x)| < C_f \varepsilon$$

for any $(t,x) \in \Omega_K$. Furthermore, $|V_\theta(T,x) - \ell(x)| < \varepsilon$ for all $x \in K$.

Proof. The second estimate follows easily from Lemma A.1 and the fact that $V(T,x) = \ell(x)$. Indeed, the fact that $\sup_{(t,x)\in\Omega_K} |V(t,x) - V_\theta(t,x)| < \varepsilon$ implies that

$$|\ell(x) - V_\theta(T,x)| = |V(T,x) - V_\theta(T,x)| < \varepsilon$$

for all $x \in K$. On the other hand, Lemma A.1 also guarantees that for any $(t,x) \in \Omega_K$,

$$\sup_{(t,x)\in\Omega_K} \|\nabla_x V(t,x) - \nabla_x V_\theta(t,x)\|_2 < \varepsilon.$$

As a result, for *fixed* $(t,x) \in \Omega_K$, taking

$$p(t,x) := \frac{1}{\varepsilon}(\nabla_x V(t,x) - \nabla_x V_\theta(t,x)) \in \mathbb{R}^n,$$

it follows that

$$\nabla_x V_\theta(t,x) = \nabla_x V(t,x) + \varepsilon p(t,x),$$

where $\|p(t,x)\|_2 < 1$. In turn, we can write

$$\begin{aligned}|H_\theta(t,x) - H(t,x)| &= \left|\sup_u \inf_d \langle \nabla_x V_\theta(t,x), f(x,u,d)\rangle - \sup_u \inf_d \langle \nabla_x V(t,x), f(x,u,d)\rangle\right| \\ &= \left|\sup_u \inf_d \langle \nabla_x V(t,x) + \varepsilon p(t,x), f(x,u,d)\rangle - \sup_u \inf_d \langle \nabla_x V(t,x), f(x,u,d)\rangle\right| \\ &= \varepsilon \left|\sup_u \inf_d \langle p(t,x), f(x,u,d)\rangle\right| \\ &\le \varepsilon \sup_u \left|\inf_d \langle p(t,x), f(x,u,d)\rangle\right| \\ &\le \varepsilon \sup_u \sup_d |\langle p(t,x), f(x,u,d)\rangle| \\ &\le \varepsilon \sup_u \sup_d \|p(t,x)\|_2 \|f(x,u,d)\|_2 \\ &< \varepsilon \sup_u \sup_d \|f(x,u,d)\|_2 \\ &\le C_f \varepsilon,\end{aligned}$$

The chain of inequalities above utilizes several standard tricks to interchange the absolute value with the supremum over control inputs u (resp. disturbances d) before applying the Cauchy–Schwarz inequality and the fact that $\|p(t,x)\|_2 < 1$ for all $(t,x) \in \Omega_K$. Finally, the last inequality utilizes the assumption that the dynamics are bounded. □

Because the bounds obtained in Lemma A.2 are uniform in $(t,x) \in \Omega_K$, we can in fact write that

$$\|H_\theta - H\|_\infty < C_f \varepsilon, \quad \|V_\theta(T,\cdot) - \ell\|_\infty < \varepsilon,$$

where $\|\cdot\|_\infty$ denotes the supremum norm on Ω_K as in the definition of the DeepReach loss in Equation (5). With both of the above lemmas, we can now prove Theorem 4.1.

Proof of Theorem 4.1. By Lemma A.1, we obtain the existence of $V_\theta \in \mathfrak{C}_{n+1}(\sigma)$ such that

$$\begin{aligned}\varepsilon > \sup_{(t,x)\in\Omega_K} |V(t,x) - V_\theta(t,x)| + \sup_{(t,x)\in\Omega_K} |\partial_t V(t,x) - \partial_t V_\theta(t,x)| \\ + \sup_{(t,x)\in\Omega_K} \|\nabla_x V(t,x) - \nabla_x V_\theta(t,x)\|_2.\end{aligned}$$

It remains to show that V_θ satisfies the conditions in the statement of Theorem 4.1. Namely, we must show that $L(\theta) \le C\varepsilon$ for some $C > 0$ that only depends on the dynamics of the underlying reachability problem. Recall that

$$\begin{aligned}L(\theta) &= h_1(\theta) + \lambda h_2(\theta) \\ &= \|V_\theta(T,x) - \ell(x)\|_\infty + \lambda \| \min\{\partial_t V_\theta(t,x) + H_\theta(t,x), \ell(x) - V_\theta(t,x)\}\|_\infty,\end{aligned}$$

where $\lambda > 0$ is a tunable parameter. Now, Lemma A.2 allows us to establish two useful lower bounds. First, because V satisfies the HJI equation in Equation (3), we have that

$$\partial_t V(t,x) + H(t,x) \ge \min\{\partial_t V(t,x) + H(t,x), \ell(x) - V(t,x)\} = 0$$

for all $(t,x) \in \Omega_K$. Thus, it follows that

$$\partial_t V_\theta(t,x) + H_\theta(t,x) \ge \partial_t V_\theta(t,x) + H_\theta(t,x) - \partial_t V(t,x) + H(t,x) > -\varepsilon - C_f\varepsilon,$$

applying two of the bounds from Lemma A.2. Taking $C := 2\max\{C_f, 1\}$, it follows that

$$\partial_t V_\theta(t,x) + H_\theta(t,x) > -C\varepsilon \tag{8}$$

for all $(t,x) \in \Omega_K$. Similarly, observe that because

$$\ell(x) - V(t,x) \ge \{\partial_t V(t,x) + H(t,x), \ell(x) - V(t,x)\} = 0$$

for all (t,x), we have that

$$\ell(x) - V_\theta(t,x) \ge \ell(x) - V_\theta(t,x) - \ell(x) - V(t,x) = V_\theta(t,x) - V(t,x) > -\varepsilon \ge -C\varepsilon \tag{9}$$

again applying the corresponding bound from Lemma A.2. Combining Equation (8) and Equation (9), it follows that

$$\min\{\partial_t V_\theta(t,x) + H_\theta(t,x), \ell(x) - V_\theta(t,x)\} > -C\varepsilon$$

for all $(t,x) \in \Omega_K$.

Now, because V solves Equation (3), for all $(t,x) \in \Omega_K$, we must have that either $\partial_t V(t,x) + H(t,x) = 0$ or $\ell(x) - V(t,x) = 0$ (or, both conditions could possibly hold, but this case is not relevant below). In the former case, we have that

$$\begin{aligned}-C\varepsilon < \min\{\partial_t V_\theta(t,x) + H_\theta(t,x), \ell(x) - V_\theta(t,x)\} &\le \partial_t V_\theta(t,x) + H_\theta(t,x) \\ &= \partial_t V_\theta(t,x) + H_\theta(t,x) - \partial_t(t,x) - H(t,x) \\ &\le C\varepsilon,\end{aligned}$$

again applying the result of Lemma A.2 in the last line. On the other hand, if $(t,x) \in \Omega_K$ is such that $\ell(x) - V(t,x) = 0$, then we see that

$$\begin{aligned}-C\varepsilon < \min\{\partial_t V_\theta(t,x) + H_\theta(t,x), \ell(x) - V_\theta(t,x)\} &\le \ell(x) - V_\theta(t,x) \\ &= \ell(x) - V_\theta(t,x) - (\ell(x) - V(t,x)) \\ &= V(t,x) - V_\theta(t,x) \\ &< \varepsilon \\ &\le C\varepsilon,\end{aligned}$$

applying the result of Lemma A.2 yet again. Because the above two cases are exhaustive, it follows that

$$|\min\{\partial_t V_\theta(t,x) + H_\theta(t,x), \ell(x) - V_\theta(t,x)\}| < C\varepsilon$$

for all $(t,x) \in \Omega_K$. In other words,

$$h_2(\theta) = \|\min\{\partial_t V_\theta(t,x) + H_\theta(t,x), \ell(x) - V_\theta(t,x)\}\|_\infty < C\varepsilon.$$

Now, the fact that $\ell(x) = V(T,x)$ for all $x \in K$ implies that

$$h_1(\theta) = \|V_\theta(T,x) - \ell(x)\|_\infty = \|V_\theta(T,x) - V(t,x)\|_\infty < \varepsilon \leq C\varepsilon,$$

by Lemma A.2. Thus, it follows that

$$L(\theta) = h_1(\theta) + \lambda h_2(\theta) < (C + \lambda C)\varepsilon.$$

By taking $C' := C + \lambda C$, which only depends on the parameter $\lambda > 0$ and the dynamics f of the underlying reachability problem, we conclude that for any $\varepsilon > 0$, there exists some set of parameters θ such that $L(\theta) < C'\varepsilon$. □

The above result ensures that the DeepReach loss is an appropriate metric for solving Equation (3) numerically: good approximations of the true solution to Equation (3) correspond to small loss values in Equation (5). However, it is not particularly practical. To provide a practical guarantee, we turn to Theorem 4.2. Rather than simply establishing the *existence* of a network that makes the DeepReach loss arbitrarily small, Theorem 4.2 shows that by performing training under which the DeepReach loss converges to zero (which is possible by the above result), the corresponding neural networks must converge uniformly to the solution to Equation (3). In some sense, this result justifies the use of the DeepReach algorithm, as it guarantees that training via gradient descent (or some other method) to minimize the DeepReach loss will result in estimators that uniformly converge to the true value function for a given reachability problem.

Before presenting the proof of Theorem 4.2, several tools from PDE theory, and specifically, the theory of viscosity solutions to PDEs, are necessary. Note that, as discussed in [1] and [9], the HJI equation in Equation (3) admits a viscosity solution V. We utilize the following standard definition of viscosity solutions, introduced in [9]:

Definition A.3. *A function $v \in \mathcal{C}((0,T) \times Int(K))$ is:*

(i) a viscosity subsolution of Equation (3) if for any test function $\varphi \in \mathcal{C}^1((0,T) \times Int(K))$, $\mathcal{L}[\varphi] \leq 0$ for every local maximum $(t_0, x_0) \in (0,T) \times Int(K)$ of $v - \varphi$ on $(0,T) \times Int(K)$.

(ii) a viscosity supersolution of Equation (3) if for any test function $\varphi \in \mathcal{C}^1((0,T) \times Int(K))$, $\mathcal{L}[\varphi] \geq 0$ for every local minmum $(t_0, x_0) \in (0,T) \times Int(K)$ of $v - \varphi$ on $(0,T) \times Int(K)$.

(iii) a viscosity solution of Equation (3) if v is both a viscosity subsolution and viscosity supersolution.

More background on viscosity solutions, their motivation, and their many useful properties can be found in [9] and [16]. Without going into unnecessary technical detail, we remark that any classical solution to Equation (3) is also a viscosity solution, a fact that we leverage below [16]. Additionally, viscosity solutions to Equation (3) satisfy a standard comparison principle, in the sense that if u is a viscosity supersolution and v a viscosity subsolution, then $v \leq u$ on $[0,T) \times K$ [9, 16].

We also require the notion of a *proper* nonlinear PDE, in the context of viscosity solutions. This concept, also from [16] is crucial for establishing uniform convergence guarantees for equations that admit viscosity solutions.

Definition A.4. *Suppose that a nonlinear PDE is of the form $F(x, u, Du, D^2u) = 0$, where $F : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathcal{S}^n \to \mathbb{R}$, with $\mathcal{S}^n$ denoting the set of symmetric matrices. Then, F is proper if, holding all other inputs fixed,*

$$F(x, r, p, X) \leq F(x, s, p, Y)$$

for all $r \leq s$ and $X \preceq Y$, with the latter denoting the standard order on PSD matrices.

From the above definition a straightforward reformulation of Equation (3) shows that the HJI equation in question is necessarily proper.

Lemma A.5. *There exists a map* $F : \Omega_K \times \mathbb{R} \times \mathbb{R}^{n+1} \times \mathcal{S}^{n+1} \to \mathbb{R}$ *such that* V *solves Equation* (3) *if and only if*

$$F(t, x, V(t, x), \nabla V(t, x), \nabla^2 V(t, x)) = 0,$$

and the resulting equation is proper in the sense of Definition A.4.

Proof. Any solution to Equation (3) satisfies

$$\min \{\partial_t V(t, x) + H(t, x), \ell(x) - V(t, x)\} = 0$$

on Ω_K. Because $\min\{a, b\} = -\max\{-a, -b\}$, the above equality holds if and only if

$$-\max \{-\partial_t V(t, x) - H(t, x), V(t, x) - \ell(x)\} = 0. \tag{10}$$

Thus, we take

$$F(t, x, r, p, X) := \max \left\{ -p_t - \sup_u \inf_d \langle p_x, f(x, u, d)\rangle, r - \ell(x) \right\}, \tag{11}$$

where $p = (p_t, p_x) \in \mathbb{R} \times \mathbb{R}^n$, taking into account both the time and spatial derivatives of solutions to Equation (3). Then, Equation (10) holds if and only if

$$F(t, x, V(t, x), \nabla V(t, x), \nabla^2 V(t, x)) = 0.$$

Furthermore, F defines a proper equation because it does not depend on the input $X \in \mathcal{S}^{n+1}$, and if $r \leq s$, then $r - \ell(x) \leq s - \ell(x)$ for any $x \in \Omega_K$. In turn,

$$\begin{aligned} F(t, x, r, p, X) &= \max \left\{ -p_t - \sup_u \inf_d \langle p_x, f(x, u, d)\rangle, r - \ell(x) \right\} \\ &\leq \max \left\{ -p_t - \sup_u \inf_d \langle p_x, f(x, u, d)\rangle, s - \ell(x) \right\} \\ &= F(t, x, s, p, X), \end{aligned}$$

holding all other inputs fixed. Thus, F is proper. □

Before proving Theorem 4.2, consider the following *perturbed* version of Equation (3), indexed by $k \in \mathbb{N}$.

$$\begin{aligned} &\min \{\partial_t U(t, x) + H(t, x), \ell(x) - U(t, x)\} = \varepsilon^{(k)}(t, x), \\ &U(T, x) = \ell(x) + \delta^{(k)}(x). \end{aligned} \tag{12}$$

From Theorem 4.1, it immediately follows that there exists a sequence of neural networks $\{V_{\theta^{(k)}}\}_{k\in\mathbb{N}}$ that solve perturbed HJI equations of the above form. Taking

$$\varepsilon^{(k)}(t, x) := \min \{\partial_t V_{\theta^{(k)}}(t, x) + H_{\theta^{(k)}}(t, x), \ell(x) - U(t, x)\}, \quad \delta^{(k)}(x) := V_{\theta^{(k)}}(T, x) - \ell(x),$$

Theorem 4.1 establishes that the sequence $\{V_{\theta^{(k)}}\}_{k\in\mathbb{N}}$ satisfies Equation (12), and we further have that

$$L(\theta^{(k)}) = h_1(\theta^{(k)}) + \lambda h_2(\theta^{(k)}) = \|\varepsilon^{(k)}\|_\infty + \lambda \|\delta^{(k)}\|_\infty \to 0$$

as $k \to \infty$. Conversely, it also holds that if $L(\theta^{(k)}) \to 0$, then the corresponding neural networks satisfy perturbed equations akin to Equation (12). Furthermore, by the construction of the DeepReach loss, it follows that $\|\varepsilon^{(k)}\|_\infty \to 0$ and $\|\delta^{(k)}\|_\infty \to 0$ as $k \to \infty$. From Theorem (4.1), we know that such a sequence of parameters exists.

Now, the sequence of above equations could instead be written as

$$F(t, x, U(t, x), \nabla U(t, x), \nabla^2 U(t, x)) = \max \{-\partial_t U(t, x) - H(t, x), \ell(x) - U(t, x)\} = -\varepsilon^{(k)}(t, x),$$

where F is as in Lemma A.5. In turn, defining

$$F^{(k)}(t, x, U(t, x), \nabla U(t, x), \nabla^2 U(t, x)) := F(t, x, U(t, x), \nabla U(t, x), \nabla^2 U(t, x)) + \varepsilon^{(k)}(t, x), \tag{13}$$

it immediately follows from Lemma A.5 that the sequence in Equation (12) is proper, given by

$$F^{(k)}(t, x, U(t, x), \nabla U(t, x), \nabla^2 U(t, x)) = 0, \tag{14}$$

and that $F^{(k)} \to F$ uniformly as $k \to \infty$.

The novelty of Theorem 4.2, however, lies in establishing that for *any* such sequence of neural networks $\{V_{\theta^{(k)}}\}_{k \in \mathbb{N}}$ obtained via training, $V_{\theta^{(k)}} \to V$ uniformly on Ω_K. In essence, rather than simply establishing the existence of such a sequence, the following proof guarantees that standard training procedures such as gradient descent, coupled with standard techniques to avoid local minima (e.g., momentum, randomized restarts or perturbations, etc.), on the DeepReach loss will result in approximations that uniformly converge to the true solution to Equation (3) as training progresses. We require one more technical lemma, based on the convergence properties of viscosity solutions from [16].

Lemma A.6. *Suppose that u_k is a sequence of viscosity subsolutions to a sequence of proper equations, defined by $F^{(k)}(x, u, Du, D^2u) = 0$, for $x \in K \subset \mathbb{R}^n$ compact. Then,*

$$\overline{u}(x) := \lim_{j \to \infty} \sup \left\{ u_k(y) : k \geq j,\ y \in K,\ \|y - x\|_2 \leq \frac{1}{j} \right\}$$

is a viscosity subsolution to the equation $\overline{F}(x, u, Du, D^2u) = 0$, where

$$\overline{F}(x, u, Du, D^2u) := \limsup_{k \to \infty} F^{(k)}(x, u, Du, D^2u).$$

Similarly, if u_k is instead a sequence of viscosity supersolutions to $F_k(x, u, Du, D^2u) = 0$, then

$$\underline{u}(x) := \lim_{j \to \infty} \inf \left\{ u_k(y) : k \geq j,\ y \in K,\ \|y - x\|_2 \leq \frac{1}{j} \right\}$$

is a viscosity supersolution to the equation $\underline{F}(x, u, Du, D^2u) = 0$, where $\underline{F}$ is defined by taking the limit infimum of $F^{(k)}$ above.

Proof. This is the result of Lemma 6.1 and the corresponding Remarks 6.2, 6.3, and 6.4 in [16]. □

Equipped with the above technical lemma, we are finally prepared to prove Theorem 4.2.

Proof of Theorem 4.2. Consider any sequence of parameters $\{\theta^{(k)}\}_{k \in \mathbb{N}}$ such that $L(\theta^{(k)}) \to 0$ as $k \to \infty$. As discussed above, such a sequence of parameters defines a family $\{V_{\theta^{(k)}}\}_{k \in \mathbb{N}}$ that satisfies the sequence of perturbed PDEs in Equation (12), with error terms $\varepsilon^{(k)} : [0, T] \times K \to \mathbb{R}$ and $\delta^{(k)} : K \to \mathbb{R}$ satisfying

$$\|\varepsilon^{(k)}\|_\infty \to 0, \quad \|\delta^{(k)}\|_\infty \to 0$$

as $k \to \infty$. By Lemma A.5, these perturbed PDEs are equivalently described by proper formulations, defined in Equation (14). Now, to invoke Lemma A.6, we define

$$\overline{V}(t, x) := \lim_{j \to \infty} \sup \left\{ V_{\theta^{(k)}}(s, y) : k \geq j,\ (s, y) \in \Omega_K,\ \|(t, x) - (s, y)\|_2 \leq \frac{1}{j} \right\}$$

and

$$\underline{V}(t, x) := \lim_{j \to \infty} \inf \left\{ V_{\theta^{(k)}}(s, y) : k \geq j,\ (s, y) \in \Omega_K,\ \|(t, x) - (s, y)\|_2 \leq \frac{1}{j} \right\}.$$

Because $F^{(k)} \to F$ uniformly on $\Omega_K \times \mathbb{R} \times \mathbb{R}^{n+1} \times \mathcal{S}^{n+1}$ as $k \to \infty$, it follows that

$$\liminf_{k \to \infty} F^{(k)} = F = \limsup_{k \to \infty} F^{(k)}.$$

Thus, by Lemma A.6, $\overline{V}(t, x)$ is a viscosity subsolution to the proper equation defined by F in Equation (11). Similarly, $\underline{V}(t, x)$ is a viscosity supersolution to the proper equation defined in Equation (11). Note that, by construction, $\overline{V}$ and $\underline{V}$ satisfy the pointwise inequality

$$\underline{V}(t, x) \leq \overline{V}(t, x)$$

for all $(t, x) \in \Omega_K$. However, because $\overline{V}$ is a viscosity supersolution and $\underline{V}$ a viscosity subsolution, by the comparison principle for viscosity solutions in [16, Thoerem 3.3], we have that

$$\overline{V}(t, x) \leq \underline{V}(t, x)$$

for all $(t, x) \in \Omega_K$. Consequently, it follows that $\overline{V}(t, x) = \underline{V}(t, x)$, and the resulting function is both a viscosity subsolution and viscosity supersolution. Hence, by Definition A.3, $\overline{V}(t, x) = \underline{V}(t, x)$ is a viscosity solution to Equation (11). By Assumption (B), however, Equation (11) admits a unique viscosity solution, so $\overline{V}(t, x) = \underline{V}(t, x) = V(t, x)$. Then, by [16, Remark 6.4], the construction of $\overline{V}$ and $\underline{V}$ respectively ensures that

$$\lim_{k \to \infty} V_{\theta^{(k)}}(t, x) = V(t, x)$$

on any compact subset of $[0, T) \times K$. If not, then there would exists $\nu > 0$ and a sequence $\{n_k\}_{k \in \mathbb{N}}$ and $\{(t_k, x_k)\}_{k \in \mathbb{N}} \in [0, T) \times K$, with $(t_k, x_k) \to (t, x) \in [0, T] \times K$, such that

$$V_{\theta^{(n_k)}}(t_k, x_k) - V(t_k, x_k) > \varepsilon \quad \text{or} \quad V_{\theta^{(n_k)}}(t_k, x_k) - V(t_k, x_k) < -\varepsilon.$$

Taking $k \to \infty$ however, and using the definitions of $\overline{V}$ and $\underline{V}$ respectively, the continuity of V would imply that

$$|V(t, x) - V(t, x)| \geq \varepsilon$$

for some $(t, k) \in [0, T] \times K$, a clear contradiction. Thus, uniform convergence holds any any compact subset of $[0, T) \times K$. To extend convergence to *all* of $\Omega_K = [0, T] \times K$, recall that $\delta^{(k)}(x) = V_{\theta^{(k)}}(T, x) - \ell(x)$ converges uniformly to zero as $k \to \infty$, for all $x \in K$. As a result, it follows that

$$\lim_{k \to \infty} \sup_{(t,x) \in \Omega_K} |V_{\theta^{(k)}}(t, x) - V(t, x)| = 0$$

as claimed. □

B Baseline Collision Avoidance Application

In this appendix, we describe the baseline collision avoidance example, explored in [2, 5] and Section 5 above. Consider the case of two airplanes approaching each other midair, as in the original DeepReach paper [2]. In the worst-case scenario, one of the planes acts as a pursuer that attempts to collide with the other plane, while the other plane acts as an evader. The relative dynamics of this system, presented in [2] and [13], are given by

$$\dot{x}_1 = -v_e + v_p \cos\theta + \omega_e x_2, \quad \dot{x}_2 = v_p \sin\theta - \omega_e x_1, \quad \dot{\theta} = \omega_p - \omega_e, \tag{15}$$

where $x = (x_1, x_2)$ represents the relative positions of the two planes in a two-dimensional plane, and θ is their relative heading. The two velocities v_e and v_p are the evader's and pursuer's velocities respectively, and the angular velocities ω_e and ω_p are defined analogously. For simplicity, the velocities v_e and v_p are held constant, while the two agents select inputs ω_e and ω_p. Note that, in our notation above, ω_p, the control input for the pursuer, is actually the *disturbance* in the context of the reachability problem. We constrain $|\omega_e|, |\omega_p| \leq \omega_{\max}$ for some constant $\omega_{\max} > 0$, and consider a standard unsafe set of the form $\mathcal{L} = \{x : \|x\|_2 \leq \beta\}$.

Taking $\ell(x) = \|x\|_2 - \beta$, where $\beta > 0$ is a safety parameter, it follows that $\mathcal{L} = \{x : \ell(x) \leq 0\}$, as is desired in the setting of HJ reachability. Consequently, we can carry out the DeepReach algorithm as outlined above, training the neural network using the DeepReach loss in Equation (5) to obtain an approximation V_θ for the value function. In turn, the sublevel set

$$\mathcal{V}^{\theta}_{\text{BRT}}(t) := \{x : V_\theta(t, x) \leq 0\}$$

will approximate the true BRT for the collision avoidance problem. As noted in [15], this example is a natural baseline to consider for reachability analysis because the *true* BRT $\mathcal{V}$ is possible to describe analytically (at least, very nearly, up to some sampling error). Additionally, as shown in [2], it is possible to analytically compute the Hamiltonian for this problem. Denoting

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} := \begin{bmatrix} \partial_{x_1} V(t, x_1, x_2, \theta) \\ \partial_{x_2} V(t, x_1, x_2, \theta) \\ \partial_{\theta} V(t, x_1, x_2, \theta) \end{bmatrix},$$

we have that

$$H(t,x) = p_1(-v_e + v_p \cos x_3) + p_2(v_p \sin x_3) - \omega_{\max}|p_1 x_3 - p_2 x_1 - p_3| + \omega_{\max} p_3 \tag{16}$$

Because the Hamiltonian can be computed explicitly, implementing the DeepReach loss is straightforward in this instance. For general reachability problems, however, it is possible to efficiently solve the optimization problem in Equation (4) at each step [2]. This basic setup can be extended to higher-dimensional collision avoidance problems. For instance, with two evading agents and one pursuing agent, the problem becomes a nine-dimensional collision avoidance problem. As shown in [2], DeepReach still provides a robust approximation of the BRT in this high-dimensional setting.

C DeepReach Trained with ℓ^1-Loss

In this appendix, we provide specific details pertaining to the training and fine-tuning of DeepReach models, as well as plots reproducing the results of [2]. All plots in this appendix are produced using a DeepReach model, pre-trained exactly as in [2], with ℓ^1-loss. If ℓ^2-loss is used instead, we obtain comparable results. By pretraining with ℓ^1-loss (resp. ℓ^2-loss) before fine-tuning with sup-norm loss, we can circumvent the expensive, time-consuming procedure of training DeepReach models from scratch while still obtaining a useful comparison of the impact that training sup-norm loss can have on the accuracy of such models.

Regardless of the loss metric in use, at each step of DeepReach training, $K = 65000$ uniformly-sampled points are used to train the neural network as in Equation (5). All states are scaled to lie in the interval $[-1, 1]$, and the time interval in question is rescaled to satisfy $T = 1$, both of which improve training stability. As in [2], we utilize a simple three-layer feedforward neural network, with hidden layer size 512 and sinusoidal activation given by $\sigma(x) = \sin(x)$. The hyperparameter λ is set to $\lambda = 0$ during pre-training (500 – 1K epochs, depending on the trial) and $\lambda = 1.5 \times 10^2$ during training (the remaining 500 – 10K epcohs, depending on the trial), and the neural network loss in minimized using the Adam optimizer. The true BRT for the collision avoidance problem is computed numerically, using the Level Set Toolbox (LST) PDE solver, as in implementation provided by [2, 15]. LST also provides a numerical solution for the true value function V.

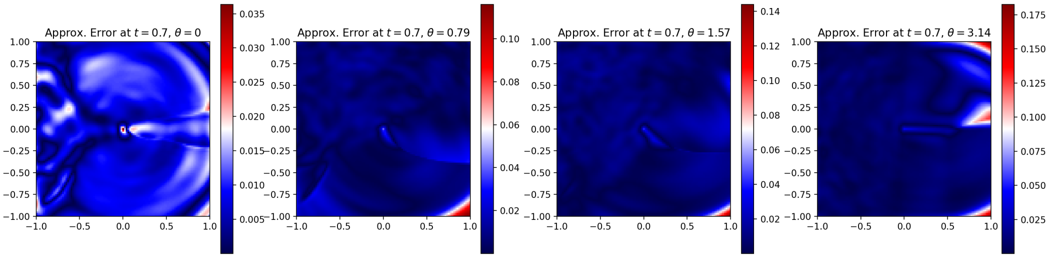

Figure 5: Absolute difference between true and approximate value functions, using pre-trained DeepReach model from [2].

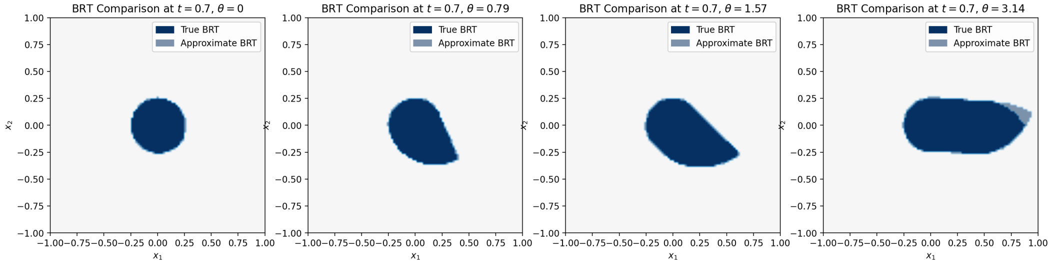

Figure 6: Comparison of BRT obtained by pre-trained DeepReach model and analytical BRT from [15]. Note the discrepancy between the two sets at $\theta = \pi$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction, both theoretical and empirical, are justified in the paper. Theoretical proofs are in Section 4 and Appendix A, and empirical results are in Section 5 and Appendix C.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of the assumptions underlying theoretical results are discussed and qualified in Sections 3 and 4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All assumptions and complete proofs are found in Appendix A, and intuitive proof sketches are in Section 4.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 and Appendix C describe how to replicate all numerical results, as well as the connection between numerical results and theoretical guarantees.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Section 5 provides a link to a Colab notebook that allows users to view all experimental code and replicate this paper's results. Appendix C contains additional instructions for replicating the paper's results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3, Section 5, Appendix C and the linked Colab notebook describe all training and test details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: When applicable, the paper discusses statistical significance of experiments in Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5and Appendix C describe all computational resources used in numerical experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper abides by the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 1 and Section 6 describe how the theoretical guarantees in this paper may allow for the development of more robust collision-avoidance systems.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All original creators and owners of code used in this paper's numerical experiments are properly credited in Section 5, Appendix C, and the linked Colab notebook.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: New assets are not introduced in this paper, although a link to the code used for numerical experiments can be found in Section 5.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.