Value Approximation for Two-Player General-Sum Differential Games With State Constraints

Lei Zhang, *Student Member, IEEE*, Mukesh Ghimire, *Student Member, IEEE*, Wenlong Zhang, *Member, IEEE*, Zhe Xu, *Member, IEEE*, and Yi Ren, *Member, IEEE*

***Abstract*—Solving Hamilton–Jacobi–Isaacs (HJI) PDEs numerically enables equilibrial feedback control in two-player differential games, yet faces the curse of dimensionality (CoD). While physics-informed neural networks (PINNs) have shown promise in alleviating CoD in solving PDEs, vanilla PINNs fall short in learning discontinuous solutions due to their sampling nature, leading to poor safety performance of the resulting policies when values are discontinuous due to state or temporal logic constraints. In this study, we explore three potential solutions to this challenge: 1) a hybrid learning method that is guided by both supervisory equilibria and the HJI PDE, 2) a value-hardening method where a sequence of HJIs are solved with increasing Lipschitz constant on the constraint violation penalty, and 3) the epigraphical technique that lifts the value to a higher dimensional state space where it becomes continuous. Evaluations through 5-D and 9-D vehicle and 13-D drone simulations reveal that the hybrid method outperforms others in terms of generalization and safety performance by taking advantage of both the supervisory equilibrium values and co-states, and the low cost of PINN loss gradients.**

***Index Terms*—General-sum differential game, physics-informed neural network (PINN), safe human–robot interactions.**

I. Introduction

HUMAN–ROBOT interactions (HRIs) become prevalent in safety-critical applications, such as transportation [1], healthcare [2], and rescue [3]. Conventionally, safety is achieved by incorporating state constraints in a model predictive control (MPC) framework. The constraints are usually derived from a two-player zero-sum game formulation so that the ego player avoids all system states from which the fellow player can successfully launch attacks should it be adversarial [4]. There are two limitations to this approach as follows. First, the zero-sum setting can often be overly conservative since fellow players in civil applications are not always adversarial. Second, real-time MPC is required on top of value approximation of the zero-sum games, limiting the speed and quality of the player's decision making.

To address the first limitation, it is tempting to consider HRI as general-sum differential games with state constraints and incomplete information, where players have private types (e.g., reward parameters). In this setting, players can overcome unnecessary conservatism by updating their beliefs about each other's type based on observations of their previous actions. To address the second limitation, one would ideally need to obtain the value of the game, which then enables feedback control that intrinsically satisfies the state constraints while optimizing the expected payoff, either obsoleting or at least accelerating MPC.

A theoretical challenge toward these idealistic goals, however, is that we do not have the existence proof or the characterization of values for general-sum differential games with incomplete information and state constraints [5]. Hence, we take a step back and consider games with complete information, for which Nash equilibrium exists [6] and therefore values are governed by the Hamilton–Jacobi–Isaacs (HJI) equations. Computing values, however, is known to encounter the curse of dimensionality (CoD) using mesh-based dynamic programming (DP) solvers [7]. Physics-informed neural network (PINN) has thus been introduced to approximate values while circumventing CoD [8]. Nonetheless, recent studies showed that while PINN is successful at approximating Lipschitz continuous PDE solutions [8], [9], [10], they encounter convergence issues when applied to discontinuous ones [11]. In the context of HJI, such value discontinuity arises when state constraints and temporal logic specifications are imposed.

Within this context, this article investigates three PINN-based solutions for approximating values of state-constrained differential games.

The first solution, called hybrid learning (HL), is developed based on the insight that discontinuity in value causes sampling-based methods, such as PINN, to deviate from the true solutions almost surely, since the measure of the discontinuous boundaries is zero (or close to zero when we approximate discontinuities with large-Lipschitz functions in practice). The solution is thus to augment PINN with supervised equilibrium data that cover discontinuous regions of the value landscape in space and time. These equilibria are generated by solving boundary value problems (BVPs) following Pontryagin's maximum principle (PMP) [12]. This solution requires human insights on where the informative equilibrium trajectories with discontinuous values (e.g., collisions) lie and the global optimality of

Received 15 January 2024; accepted 27 March 2024. Date of publication 10 June 2024; date of current version 7 November 2024. This work was supported in part by the NSF under Grant CMMI-1925403 and in part by the NSF under Grant CNS-2101052. This article was recommended for publication by Associate Editor Eduardo Montijano and Editor Javier Civera upon evaluation of the reviewers' comments. *(Corresponding author: Yi Ren.)*

Lei Zhang, Mukesh Ghimire, Zhe Xu, and Yi Ren are with the Department of Mechanical and Aerospace Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: lzhan300@asu.edu; mghimire@asu.edu; xzhe1@asu.edu; yiren@asu.edu).

Wenlong Zhang is with the School of Manufacturing Systems and Networks, Ira A. Fulton Schools of Engineering, Arizona State University, Mesa, AZ 85212 USA (e-mail: wenlong.zhang@asu.edu).

Digital Object Identifier 10.1109/TRO.2024.3411850

the BVP solutions. The challenge with sampling discontinuous boundaries leads to the loss of spatiotemporal causality during value approximation. Hence, the second solution, called value hardening (VH), following curriculum learning [13], aims to improve the chance of learning the discontinuous boundaries by gradually increasing the Lipschitz constant of a constraint violation penalty. The third solution, called epigraphical learning (EL), is based on the epigraphical technique that transforms discontinuous values of state-constrained games into Lipschitz continuous ones defined in an augmented state space [14]. We extend the existing technique from zero-sum games [15] to the general-sum setting and apply PINN to approximate the smooth augmented values.

We summarize the systemic design of experiments to be used to evaluate and compare these solutions. **Methods:** vanilla PINN, supervised learning (SL), HL, VH, EL methods, and their variants. **PDE dimensionalities:** 5-D, 9-D, and 13-D. As of writing, 13-D is largest dimensionality among existing test cases of HJ equations in the differential game context. **Dynamics:** linear and nonlinear vehicle and drone dynamics. **Information settings:** complete- and incomplete-information two-player general-sum games. **Performance metrics:** both in- and out-of-distribution generalization and safety performance.

We claim the following contributions.

1) We show that HL scales better than SL, VH, and EL to high-dimensional cases in terms of both generalization (value and action prediction) and safety (when values are used for feedback control). The key factors for its success are: i) the supervision on the co-state landscape, which is directly related to the control policy, ii) the low cost of PINN training in comparison to SL via solving BVPs.
2) Consistent with [16], [17], and [18], our ablation studies highlight the sensitivity of generalization and safety performance to the choice of neural activation functions, and the need for adaptive activations. In particular, `tanh` and continuously differentiable variants of `relu`, such as `gelu` [19], achieve the best empirical performance when combined with HL and adaptive activation.
3) While existing studies on solving HJ equations using machine learning have shown promising results for reachability analysis (e.g., [20]), the safety performance of the resultant value networks when used as closed-loop controllers is rarely investigated. We show in this article that low approximation errors in value do not necessarily indicate high safety performance when the approximated value is used for closed-loop control.

This work is extended from its conference version [21] in the following significant ways.

1) A thorough investigation of the efficacy of the EL technique when applied to solving differential games.
2) New studies to demonstrate and explain the convergence challenge encountered when applying VH to 9-D and 13-D problems.
3) New studies that demonstrate the importance of co-state loss for high safety performance.
4) Extension of the existing DP solver [22] from zero-sum to general-sum setting, which enables comparisons between values approximated by DP, BVP, and PINN variants.

These comparisons allow us to show that values obtained from all three are similar in the test cases and therefore guiding PINN by open-loop BVP solutions through HL is reasonable.

The rest of this article is organized as follows. Section II provides an overview of the relevant literature on value approximation, PINN, and complete- and incomplete-information differential games. In Section III, we present the formulation of two-player general-sum differential games with state constraints, its HJ PDEs, and explain the challenge in approximating its discontinuous values through a toy case. We then discuss the three potential solutions. The experimental results are presented and analyzed in Section IV. We give discussion including safety guarantee, consistency between BVP and HJI values, and efficacy of co-state loss for safety performance in Section V. Finally, Section VI concludes this article.

II. Related Work

A. Value Approximation and PINN

The values of a general-sum differential game with two-player and complete-information are viscosity solutions to HJI equations [23], which are a set of first-order nonlinear PDEs. The conventional approach to solving such equations involves essentially nonoscillatory (ENO) schemes [24] and level set methods [25], [26], which are known to provide accurate approximations of both temporal and spatial derivatives. However, these approaches suffer from CoD [27]. Recent studies have shown that using PINN to approximate PDE solutions can effectively circumvent the CoD due to its Monte Carlo nature, provided that the solution is smooth [8]. PINN trains neural nets as PDE-governed fields, where the training loss is defined by network-induced residuals with respect to: a) the boundary conditions [28], [29], b) the governing equations [18], [20], and/or c) supervisory data drawn from the ground-truth solutions [30]. Initial studies on convergence and generalization performance have emerged for a) and b), under the assumption that both the solution and the network are Lipschitz continuous [9], [10], [28]. Recent studies have explored the effectiveness of PINN for solving PDEs with discontinuous solutions, such as Burgers' equation, where both initial and terminal boundaries are specified [18]. However, we demonstrate in Section III-E that PDEs with only terminal or initial boundary conditions, such as HJIs, present an unidentifiability challenge.

B. Differential Games With Incomplete Information

One driving motivation for approximating values of differential games is to use the values for fast belief updates on unknown player types in incomplete-information settings. The update follows Bayes inference and relies on modeling player control policies as a type-conditioned distribution shaped by their values (see Section IV-A for details). In the case study on uncontrolled intersection (see Section IV-A), we evaluate the safety performance induced by the value networks, which influence both players' control policies and their belief updates about the types of their fellow players. In addition, we examine safety performance when players are "empathetic," i.e., when they share common beliefs about each other, and when they are "nonempathetic," i.e., when they falsely assume that their true types are known by their fellow players. Our study shares the same motivation as [31] in that both seek fast computation of equilibrium during interactions. We take the approach of precomputing values offline (which then enables 500 Hz policy generation frequency during inference time), while Fridovich-Keil et al. [31] proposed to simplify games as linear–quadratic which then facilitates fast (20 Hz) equilibrium approximation online. Our investigation into differential games with incomplete information sets us apart from previous HRI studies that resort to various simplifications of the games in order to balance theoretical soundness and practicality. These simplifications involve modeling the games as optimal control problems or complete-information ones [32], [33], [34], [35], [36], [37]. While some also use belief updates to adapt motion planning, they are limited to empirical best responses of the uninformed player in one-sided information settings [38], [39], [40], [41], [42]. A recent study proposes to synthesize safety control policies that account for evolving uncertainty by considering both physical and belief dynamics [43]. This framework is currently constrained to one-sided information settings, while this article studies cases where both players lack information. It is necessary to point out, however, that we will only investigate best-response policies of players, i.e., the players choose the best responses based on their *current* belief about their fellows (via their common knowledge about the values of the games) without considering the *future* dynamics of beliefs. This is because the existence of value and player policies for general-sum differential games with incomplete information is still an open question, unlike their zero-sum or discrete-time counterparts [44], [45], [46], [47].

III. Discontinuous Value Approximation

A. Notations

In a two-player differential game with complete-information, player i has a state space $\mathcal{X}_i \subset \mathbb{R}^n$ and an action space $\mathcal{U}_i \subset \mathbb{R}^m$. The time-invariant state dynamics of player i is denoted by

$$\dot{x}_i = f_i(x_i, u_i) \tag{1}$$

where $x_i \in \mathcal{X}_i$ and $u_i \in \mathcal{U}_i$. We omit dependence on time whenever possible and use $\mathbf{a}_i = (a_i, a_{-i})$ to concatenate variables a_i from player i and a_{-i} from the fellow player. We denote the partial derivative with respect to x by $\nabla_x \cdot$ and the joint state space by $\mathcal{X} := \bigcup_{i=1,2} \mathcal{X}_i$. The fixed time horizon of the game is $[0, T]$. The instantaneous loss of player i is denoted by $l_i(x_i, u_i)$ and the terminal loss $g_i(x_i)$. Feasible states from player i's perspective are defined by the subzero level set $\{\mathbf{x}_i \in \mathcal{X} \,|\, c_i(\mathbf{x}_i) \leq 0\}$. We will consider $c_i(\cdot)$ a scalar function that measures the worse state constraint violation in case multiple constraints are present, i.e., if $c_i(\mathbf{x}_i) > 0, \mathbf{x}_i$ violates at least one of the constraints. The value function of player i is denoted by $\vartheta_i(\mathbf{x}_i, t) : \mathcal{X} \times [0, T] \to \mathbb{R}$. To simplify notation, we will use f_i, l_i, g_i, c_i, and ϑ_i to refer to the dynamics, losses, state constraint, and the value function of player i. Denote by $\alpha_i \in \mathcal{A} : \mathcal{X} \times [0, T] \to \mathcal{U}_i$ player i's control policy, where the policy space $\mathcal{A}$ is assumed to be common. We use $x_s^{x_i,t,\alpha_i}$ as the state of player i at time s if it follows policy α_i and dynamics f_i starting from (x_i, t). We denote states for two players at time s as $\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha_{-i}} := (x_s^{x_i,t,\alpha_i}, x_s^{x_{-i},t,\alpha_{-i}})$. All acronyms are summarized in Appendix A.

B. Assumptions

Throughout this article, we assume that $\mathcal{U}_i$ is compact and convex; $f_i : \mathcal{X}_i \times \mathcal{U}_i \to \mathbb{R}^n$ and $c_i : \mathcal{X} \to \mathbb{R}$ are Lipschitz continuous; $l_i : \mathcal{X}_i \times \mathcal{U}_i \to \mathbb{R}$ and $g_i : \mathcal{X}_i \to \mathbb{R}$ are Lipschitz continuous and bounded.

C. Preliminary

HJI equations: Let (α_i, α_{-i}) be a pair of equilibrium policies. The values for a two-player general-sum differential game are viscosity solutions to the HJI equations denoted by (L) in (2), and satisfy the boundary conditions denoted by (D) [48]

$$\begin{aligned} L(\vartheta_i, \nabla_{\mathbf{x}_i}\vartheta_i, \mathbf{x}_i, t, \alpha_{-i}) &:= \nabla_t \vartheta_i + \max_{u_i \in \mathcal{U}_i} \left\{ \nabla_{\mathbf{x}_i} \vartheta_i^T \mathbf{f}_i - l_i \right\} = 0 \\ D(\vartheta_i, \mathbf{x}_i) &:= \vartheta_i(\mathbf{x}_i, T) - g_i = 0, \quad \text{for} \quad i = 1, 2. \end{aligned} \tag{2}$$

With the values, the players' equilibrium policies can then be derived by $\alpha_i(\mathbf{x}_i, t) = \text{arg max}_{u_i \in \mathcal{U}_i}\{\nabla_{\mathbf{x}_i} \vartheta_i^T \mathbf{f}_i - l_i\}$. Notice that L for player i depends on the equilibrium policy α_{-i} of its fellow.

PMP: Although solving the HJI equations would give a feedback control policy, it is often more practical to compute open-loop policies for a specific initial state $(\bar{x}_1, \bar{x}_2) \in \mathcal{X}$ by solving a BVP following Pontryagin's minimum principle (PMP)[1]:

$$\begin{aligned} \dot{x}_i &= f_i, \quad x_i(0) = \bar{x}_i \\ \dot{\lambda}_i &= -\nabla_{x_i} h_i, \quad \lambda_i(T) = -\nabla_{x_i} g_i \\ u_i &= \underset{u_i \in \mathcal{U}_i}{\text{arg max}} \{h_i\} \quad \text{for} \quad i = 1, 2 \end{aligned} \tag{3}$$

where λ_i is the time-dependent co-state for player i. The co-state connects PMP and HJI through $\lambda_i = \nabla_{x_i} \vartheta_i$. Solutions to (3) are specific to the given initial states. Although PMP characterizes local open-loop solutions, empirical studies (see Section V-B) show that with an effort to search for global solutions, BVP values are consistent with those governed by the HJI equations.

State-constrained value function: With state constraints, the value function for player i with some equilibrium policy pair (α_i, α_{-i}) is

$$\vartheta_i(\mathbf{x}_i, t) = \int_t^T l_i\left(x_s^{x_i,t,\alpha_i}, \alpha_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha_{-i}}, s\right)\right) ds + g_i\left(x_T^{x_i,t,\alpha_i}\right) \tag{4}$$

if $c_i(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha_{-i}}) \leq 0, \forall s \in [t, T]$, or $+\infty$ otherwise. Thus, state constraints introduce discontinuity in the value landscape.

[1] It should be noted that solving the BVP has its own numerical challenges, particularly when the equilibrium involves singular arcs [49]. However, these challenges are beyond the scope of this article.

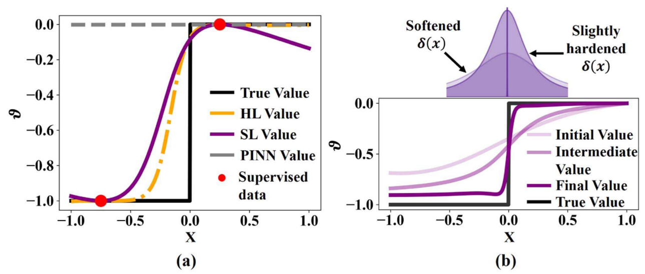


Fig. 1. (a) Value comparison among the learning methods for a simple 1-D case. Red dots are the supervised data. (b) Evolution of the value function due to gradually hardening delta function. Delta functions are shown on top. Transparency reduces with hardening.

D. PINN for Solving HJ Equation

PINN trains neural networks $\hat{\vartheta}_i(\cdot,\cdot) : \mathcal{X} \times [0,T] \to \mathbb{R}$ to approximate ϑ_i. We denote by $\mathcal{D} = \{(x_1^{(k)}, x_2^{(k)}, t^{(k)})\}_{k=1}^K$ a dataset consisting of uniform samples in $\mathcal{X}_1 \times \mathcal{X}_2 \times [0,T]$. The formulation of the training problem in (5) extends from PINN for solving zero-sum games [20]

$$\min_{\hat{\vartheta}_1,\hat{\vartheta}_2} \; L_1\left(\hat{\vartheta}_1,\hat{\vartheta}_2;\theta\right) := \sum_{k=1}^{K}\sum_{i=1}^{2}\left\| L(\hat{\vartheta}_i^{(k)}, \nabla_{\mathbf{x}_i}\hat{\vartheta}_i^{(k)}, \mathbf{x}_i^{(k)}, t^{(k)})\right\| + C_1\phi\left(D(\hat{\vartheta}_i, \mathbf{x}_i^{(k)})\right) \tag{5}$$

where $\hat{\vartheta}_i^{(k)}$ is an abbreviation for $\hat{\vartheta}_i(\mathbf{x}_i^{(k)}, t^{(k)})$ and C_1 balances the L1 PDE residual loss ($\|L\|$) and the boundary loss ($\phi(D)$). It is worth noting that in each iteration of solving (5), a subroutine is needed to find the control policies by maximizing the Hamiltonian.

E. Challenge in Approximating Discontinuous HJI Values

We use the following toy case to explain the challenge in approximating discontinuous values using PINN. Consider a 1-D function $\vartheta(x)$, which is the solution to a differential equation $\nabla_x\vartheta - \delta(x) = 0$ with the boundary condition $\vartheta(1) = 0$ in the interval $x \in [-1,1]$. $\delta(x)$ is a delta function that peaks at $x = 0$. Notice that with uniform samples for $\mathcal{D}$, the PINN loss (L_1) can be minimized almost surely by incorrect solutions, e.g., $\hat{\vartheta}(x) = 0$. This unidentifiability issue is due to the differential nature of the governing equation: the accuracy of $\hat{\vartheta}$ at one point in space and time depends solely on that of its neighbors. However, informative neighbors, i.e., those at $x = 0$ in this toy case, have zero probability to be sampled.

F. Solutions

1) Hybrid Learning: In the above-mentioned toy case, we can learn a much improved approximation to the solution using only two informative data points sampled from each side of 0 (as shown by the SL curve in Fig. 1). Indeed, Nakamura-Zimmerer et al. [30] showed that SL can be used for value approximation. A drawback of this approach, when applied to solving HJIs, is its high data acquisition costs due to the need for repeatedly solving BVPs to acquire state-value pairs. We hypothesize that this drawback can be reduced by combining SL and PINN, since evaluating and differentiating the latter only require one forward pass of $\hat{\vartheta}$, which is usually much cheaper than calling the Newton-type iterative algorithms involved in solving BVPs.

To implement this hybrid method, we define a dataset $\mathcal{D}_s = \{(\mathbf{x}_i^{(k)}, t^{(k)}, \vartheta_i^{(k)}, \nabla_{\mathbf{x}_i}\vartheta_i^{(k)}) \text{ for } i = 1,2\}_{k=1}^K$ derived from solving (3) with initial states uniformly sampled in $\mathcal{X}$. We define the supervised loss as follows:

$$\min_{\hat{\vartheta}_1,\hat{\vartheta}_2} \quad L_2\left(\hat{\vartheta}_1,\hat{\vartheta}_2;\mathcal{D}_s\right) := \sum_{k=1}^{K}\sum_{i=1}^{2}\left|\hat{\vartheta}_i^{(k)} - \vartheta_i^{(k)}\right| + C_2\left\|\nabla_{\mathbf{x}_i}\hat{\vartheta}_i^{(k)} - \nabla_{\mathbf{x}_i}\vartheta_i^{(k)}\right\| \tag{6}$$

where C_2 is a hyperparameter that balances the losses on value and its gradient. The hybrid method minimizes $L_1 + L_2$.

2) Value Hardening: The second solution is to introduce a surrogate differential equation, which has a continuous solution that approximates the ground truth. We can then approximate the true solution by gradually "hardening" this surrogate. For the toy case, we can improve the solution by gradually hardening a softened delta function, as shown in Fig. 1(b). Just like HL, this method also introduces additional computation, as we turn one learning problem into a sequence of easier learning problems. In Section IV, we show that with a limited budget, VH fails to converge for high-dimensional value approximation tasks where HL succeeds. Finally, we note that VH is similar to [11], where the authors introduce a gradually hardening diffusion term to address the same discontinuity issue when solving nonlinear two-phase hyperbolic transport equations using PINN.

3) Epigraphical Learning: Recall that the discontinuity of value in our context is caused by state constraints in differential games. It is shown that a smooth augmented value can be derived through the EL technique for state-constrained differential games [14], [15]. Our last approach utilizes this technique to facilitate continuous value approximation in an augmented state space and compute the value for the original game based on the approximation. While HJ PDEs with state constraint have been investigated in zero-sum settings and numerical approximation of their values have been attempted via DP and conservative Q-learning [15], [50], this article is among the first to solve general-sum differential games with state constraints using a combination of PINN and the EL technique. For completeness, we briefly introduce the EL technique in the following section.

G. Epigraphical Technique for General-Sum Differential Games With State Constraints

Let (α_1, α_2) be a pair of equilibrium policies. The Epigraphical technique introduces an augmented value $V_i : \mathcal{X} \times \mathbb{R} \times [0,T]$

$$V_i(\mathbf{x}_i, z_i, t) := \max\left\{ \max_{s\in[t,T]} c_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha_{-i}}\right) \; g_i\left(x_T^{x_i,t,\alpha_i}\right) - z_i(T)\right\}. \tag{7}$$

The auxiliary state z_i follows:

$$\dot{z}_i = -l_i(x_i, u_i) \text{ and } z_i(0) = \bar{z}_i \tag{8}$$

where $\bar{z}_i$ represents the true value of player i at $(\bar{\mathbf{x}}_i, t_0) \in \mathcal{X} \times [0, T]$ and is computed as follows: Find $\bar{z}_i \in [z_{\text{min}}, z_{\text{max}}]$ such that $V_i(\bar{\mathbf{x}}_i, \bar{z}_i, t_0) = 0$. If $V_i(\bar{\mathbf{x}}_i, z, t_0) > 0$ for all $z \in [z_{\text{min}}, z_{\text{max}}]$, then $\bar{z}_i = +\infty$. Lemma 1 (Lemma 1 of [15]) formally establishes this connection between the augmented value V_i and the true value $\vartheta_i(\mathbf{x}_i, t)$.

Lemma 1: Suppose assumptions in Section III-B hold. For all $(\mathbf{x}_i, z_i, t) \in \mathcal{X} \times \mathbb{R} \times [0, T]$, ϑ_i and V_i are related as follows:

$$\begin{aligned} &\vartheta_i(\mathbf{x}_i, t) - z_i \le 0 \iff V_i(\mathbf{x}_i, z_i, t) \le 0 \\ &\vartheta_i(\mathbf{x}_i, t) = \min z_i \quad \text{s.t. } V_i(\mathbf{x}_i, z_i, t) \le 0. \end{aligned} \tag{9}$$

Proof: See Appendix B.

Lemma 2 (Lemma 2 of [15]) provides the optimality condition for $V_i(\mathbf{x}_i, z_i, t)$, which is the basis for the derivation of HJ equations with state constraints.

Lemma 2: Suppose assumptions in Section III-B hold. For all $(\mathbf{x}_i, z_i, t) \in \mathcal{X} \times \mathbb{R} \times [0, T]$, for small enough $h > 0$ such that $t + h \le T$ we have

$$\begin{aligned} V_i(\mathbf{x}_i, z_i, t) = \min_{\alpha_i \in \mathcal{A}} \max \Bigg\{ &\max_{s \in [t, t+h]} c_i\left(\mathbf{x}_s^{\mathbf{x}_i, t, \alpha_i, \alpha_{-i}}\right) \\ &V_i\left(\mathbf{x}_i(t+h), z_i(t+h), t+h\right) \Bigg\} \end{aligned} \tag{10}$$

where $\mathbf{x}_s^{\mathbf{x}_i, t, \alpha_i, \alpha_{-i}}$ and $\mathbf{x}_i(t+h)$ are solutions to (1) using $(\mathbf{x}_i, t, u_i)$ and $z_i(t+h)$ is a solution to (8). α_{-i} is the equilibrium policy of the fellow player.

Proof: See Appendix C.

Theorem 1 presents the HJ equations for players in a general-sum differential game with state constraints.

Theorem 1 (HJ PDE with state constraints for general-sum differential games): For all $(\mathbf{x}_i, z_i, t) \in \mathcal{X} \times \mathbb{R} \times [0, T]$, $V_i(\mathbf{x}_i, z_i, t)$ in (7) is a viscosity solution to the following HJ PDE and boundary conditions:

$$\begin{aligned} &\max\{c_i\left(\mathbf{x}_i\right) - V_i(\mathbf{x}_i, z_i, t) \\ &\quad \nabla_t V_i - \mathcal{H}_i(\mathbf{x}_i, z_i, \nabla_{\mathbf{x}_i} V_i, \nabla_{z_i} V_i, t)\} = 0 \end{aligned} \tag{11}$$

where $\mathcal{H}_i$ is the augmented Hamiltonian

$$\mathcal{H}_i = \max_{u_i \in \mathcal{U}_i} -\nabla_{\mathbf{x}_i} V_i^T f_i + \nabla_{z_i} V_i^T l_i \tag{12}$$

and $V_i(\mathbf{x}_i, z_i, T) = \max\{c_i(\mathbf{x}_i), g_i(T) - z_i(T)\}$.

Proof: See Appendix D.

To solve state-constrained HJ PDEs using PINN, we define residuals similar to (2)

$$\begin{aligned} &\tilde{L}(V_i, \mathbf{x}_i, z_i, t) := \max\left\{c_i\left(\mathbf{x}_i\right) - V_i(\mathbf{x}_i, z_i, t)\right. \\ &\left.\nabla_t V_i - \mathcal{H}_i(\mathbf{x}_i, z_i, \nabla_{\mathbf{x}_i} V_i, \nabla_{z_i} V_i, t)\right\} \\ &\tilde{D}(V_i, \mathbf{x}_i, z_i) := V_i(\mathbf{x}_i, z_i, T) - \max\left\{c_i(\mathbf{x}_i)\right. \\ &\left.g_i(T) - z_i(T)\right\}, \text{ for } i = 1, 2. \end{aligned} \tag{13}$$

Thus, the overall loss can be expressed using the same formulation as in (5)

$$\begin{aligned} \min_{\hat{V}_1, \hat{V}_2} \quad L_3\left(\hat{V}_1, \hat{V}_2; \theta\right) := &\sum_{k=1}^{K} \sum_{i=1}^{2} \left\| \tilde{L}(\hat{V}_i^{(k)}, \mathbf{x}_i^{(k)}, z_i^{(k)}, t^{(k)}) \right\| \\ &+ C_3 \tilde{\phi}\left(\tilde{D}(\hat{V}_i^{(k)}, \mathbf{x}_i^{(k)}, z_i^{(k)})\right). \end{aligned} \tag{14}$$

To take advantage of the structure of V_i, we introduce two networks $A_i : \mathcal{X} \times [0, T] \to \mathbb{R}$ and $B_i : \mathcal{X} \times [0, T] \to \mathbb{R}$

$$\hat{V}_i(\mathbf{x}_i, z_i, t) := \max\left\{A_i(\mathbf{x}_i, t), B_i(\mathbf{x}_i, t) - z_i\right\}. \tag{15}$$

Essentially, A_i predicts the worse-case future constraint violation and B_i predicts the value of the game for player i without considering the constraint. If $A_i > 0$, then $\hat{V}_i > 0$ and ϑ does not exist, i.e., state constraint cannot be satisfied.

IV. Case Study

We conduct empirical studies to compare the generalization and safety performance of value approximation models using five different learning methods: vanilla PINN (shortened as PINN), HL, VH, EL, and SL. We use both vehicle and drone simulations to formulate the games. The first simulation involves an interaction between two players (i.e., vehicle) at an uncontrolled intersection, which leads to HJIs with coupled value functions defined on a 5-D state space. We study both complete- and incomplete-information settings using this simulation. The second and third studies investigate model safety performance on a 9-D state space. The former models a collision-avoidance case and the latter a double-lane change case. It should be noted that our settings, in terms of the dynamical models and the state space dimensions, are similar to those of [20] and [4], yet we extend from their optimal control or zero-sum settings to general-sum differential games. The last case study on drone collision avoidance investigates performance of PINN variants on a higher dimensional state space (13-D) and on nonlinear dynamics.

Data acquisition: The methods under comparison involve diverse data acquisition algorithms (supervised data via iterative BVP solving and PINN data via random sampling) and learning algorithms (supervised and curriculum learning). Hence, we use the total wall-clock time for data acquisition and learning as a unified measure of the computational cost. To ensure a fair comparison, the data size for each method is chosen to keep their computational costs as close to each other as possible. Computational costs of all the case studies are summarized in Table I. To improve training convergence, we normalize the input data to lie in $[-1, 1]$.

Network architecture: For all cases, we will present results obtained using fully connected networks with three hidden layers, each comprising 64 neurons, and with `tanh`, `relu`, or `sin` activation functions. The following experimentations on network architecture were conducted but omitted to keep this article concise. 1) Experiments on deeper and wider networks did not lead to significant improvement in generalization and safety performance, or qualitative changes to the conclusions we will present. 2) We observe that `gelu` performs similarly to `tanh` in terms of the generalization and safety performance.

TABLE I
COMPUTATIONAL COSTS FOR ALL LEARNING METHODS IN ALL CASE STUDIES

Case Study No.	Computational Cost (minutes)	Learning Method				
		HL	VH	EL	SL	PINN
Case 1	Data Acquisition	83	-	-	142	-
	Model Training	110	195	600	52	195
	Total Cost	193	195	600	194	195
Case 2	Data Acquisition	250	-	-	363	-
	Model Training	165	420	840	60	420
	Total Cost	415	420	840	423	420
Case 3	Data Acquisition	250	-	-	363	-
	Model Training	180	430	880	70	430
	Total Cost	430	430	880	433	430
Case 4	Data Acquisition	500	-	-	625	-
	Model Training	210	-	-	85	716
	Total Cost	710	-	-	710	716

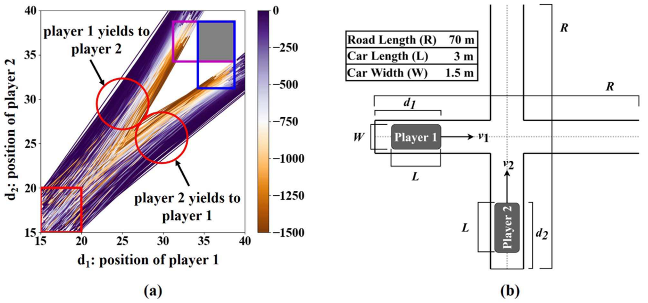


Fig. 2. (a) State trajectories of players projected to (d_1, d_2). Solid gray box: collision area from the perspective of aggressive players; hollow boxes (magenta for player 1 and blue for player 2): collision areas from the perspectives of nonaggressive players. Red box: sampling domain for initial states. Color: Actual values of player 1. (b) Uncontrolled intersection setup.

Hardware: For all case studies, all methods except EL are conducted on one workstation with 3.50 GHz Xeon E5-1620 v4 CPU and four GeForce GTX 1080 Ti GPU with 11GB memory. Due to the increased dimensionality of the augmented value in EL, we use an A100 GPU with 40GB memory to achieve convergence. Our empirical results suggest that EL is not as data efficient as the hybrid method even with this advantage.

Performance metrics: Since all case studies involve collision avoidance as their state constraints, our analysis will focus on collision rate (Col.%) as a safety metric. Specifically, collision rate is the probability of sampling an initial state for which closed-loop control of both players using the value network leads to a collision: Col.% $= N_{\text{pred}}/N_{gt}$, where N_{pred} is the number of collision trajectories resulted from the value network and N_{gt} is the number of collision-free trajectories resulted from solving BVPs. Both share the same uniform samples of initial states. In addition, we report in Case 1 generalization performance of the value networks in terms of their mean absolute approximation errors in value and control inputs along the test state trajectories. The ground truth value and control inputs are derived from BVP.

Hypotheses: The following hypotheses will be tested empirically through the case studies.

1) *With the same computational budget, HL yields better generalization and safety performance than vanilla PINN, VH, SL, and EL across all presented cases and settings. The key ingredient for high safety performance is the co-state loss.*
2) *The choice of the activation function and its parameters is critical to the safety performance. In general, continuously differentiable activations, e.g.,* `tanh` *and* `sin`*, are better than activations with discontinuous derivatives, e.g.,* `relu`.

A. Case 1: Uncontrolled Intersection

Experiment setup: The schematics of the uncontrolled intersection case and the parameters (R, L, and W for road length, car length, and car width, respectively) are depicted in Fig. 2. Each player is represented by two state variables: location (d_i) and speed (v_i), which together form the state of the player as $x_i := (d_i, v_i)$. The shared dynamics between the players follow the equations $\dot{d}_i = v_i$ and $\dot{v}_i = u_i$, where $u_i \in [-5, 10]$m/s^2 represents the scalar control input, i.e., the acceleration of the player. The instantaneous loss is

$$l_i(u_i) = u_i^2 \tag{16}$$

and the player type-dependent state constraint is

$$c_i(\mathbf{x}_i; \theta) = \delta(d_i, \theta_i)\delta(d_{-i}, 1) \tag{17}$$

where $\delta(d, \theta) = 1$ iff $d \in [R/2 - \theta W/2, (R + W)/2 + L]$ or otherwise $\delta(d, \theta) = 0$. $\theta \in \Theta := \{1, 5\}$ represents the aggressive (`a`) or nonaggressive (`na`) type of a player, where the nonaggressive player adopts a larger collision zone, see hollow boxes in Fig. 2. The terminal loss is defined to incentivize players to move across the intersection and restore nominal speed

$$g_i(x_i) = -\mu d_i(T) + (v_i(T) - \bar{v})^2 \tag{18}$$

where $\mu = 10^{-6}$, $\bar{v} = 18$ m/s, and $T = 3$ s. For hybrid, VH, and vanilla PINN, we treat the state constraint as a penalty in a modified instantaneous loss

$$\tilde{l}_i(\mathbf{x}_i, u_i; \theta) = l_i(u_i) + b\sigma(d_i, \theta_i)\sigma(d_{-i}, 1) \tag{19}$$

where

$$\begin{aligned} \sigma(d, \theta) = &\ (1 + \exp(-\gamma(d - R/2 + \theta W/2)))^{-1} \\ &\ (1 + \exp(\gamma(d - (R + W)/2 - L)))^{-1} \end{aligned} \tag{20}$$

$\gamma = 5$ is a shape parameter and $b = 10^4$ is chosen to be large enough to avoid collisions, and cause a large Lipschitz constant in the resulting value functions.

Data: For SL, 1.7k ground truth trajectories are generated from initial states uniformly sampled in $\mathcal{X}_{GT} := [15, 20]$m $\times$ $[18, 25]$m/s by solving (3). Each trajectory consists of 31 $\times$ 2 data points (sampled with a time interval of 0.1s and for two players), resulting in a total of 105.4 k data points. For vanilla PINN and VH, 122 k states are sampled uniformly in $\mathcal{X}_{HJ} :=$ $[15, 105]$m $\times$ $[15, 32]$m/s. For HL, 1k ground truth trajectories (62 k data points) are uniformly sampled in $\mathcal{X}_{GT}$, and 60k states are uniformly sampled in $\mathcal{X}_{HJ}$. For EL, we first gather a sample of 200 k states from $\mathcal{X}_{HJ}$ to ensure adherence to the boundary conditions. Subsequently, additional 110 k states are sampled from $\mathcal{X}_{HJ}$ every 30 k training iterations, resulting in a total of 1300 k sampled data points upon completion of the training process.

For the auxiliary state, recall that its initial value represents the player's value of the game. In the intersection case, the best-case loss is -1.05×10^{-4} with zero collisions and control efforts, while the worst-case loss without collision is 300 where the player constantly uses the maximum acceleration or deceleration. Hence, we uniformly sample $z_i \in [-1.05 \times 10^{-4}, 300]$. The same sampling procedure is applied to all subcases with enumeration of player types: (a, a), (na, a), (a, na), and (na, na).

The selection of state spaces to sample from, namely $\mathcal{X}_{GT}$ and $\mathcal{X}_{HJ}$, is based on various factors: In the case of ground truth trajectories, the initial states for both players are uniformly sampled from identical domains. This is because informative collision and near-collision cases often occur when players start from similar states. In addition, the range of locations for supervised data is chosen as $[15, 20]$ m to increase the likelihood of sampling informative trajectories within the specified time window. The speed range of $[18, 25]$ m/s is selected based on typical vehicle speed limits. For PINN and variants, the sample space $\mathcal{X}_{HJ}$ approximately covers all states that players can reach within the time window. It is noteworthy that within $\mathcal{X}_{HJ} \times \mathcal{X}_{HJ}$, about 20% of the states will induce collisions. Adaptive sampling for PINN, such as in [18], can potentially improve the data efficiency further but is not studied in this article.

Training: All training problems except EL are solved using the Adam optimizer with a fixed learning rate of 2×10^{-5}. For SL, the networks are trained for 100 k iterations. For vanilla PINN, we adopt the curriculum learning method proposed in [20]. Specifically, we first train the networks for 10 k iterations using 122 k uniformly sampled boundary states at the terminal time. We then refine the networks for 260 k gradient descent steps, with states sampled from an expanding time window starting from the terminal. For VH, we follow the same learning procedure, but we soften the collision penalty using sigmoid functions and gradually increased the shape parameter of the sigmoid to harden the penalty. To keep the computational cost of VH similar to that of the hybrid, we use 5.4k training iterations for each hardening step for a total of 50 steps. For the hybrid method, we pretrain the networks for 100 k iterations using the supervised data and combine the supervised data with states sampled from an expanding time window starting from the terminal time to minimize $L_1 + L_2$ for 100k iterations. For EL, we first train the network to fulfill the boundary condition over 50 k iterations. Subsequently, we refine the network through 3 k gradient steps per epoch, encompassing a total of ten epochs for every 30 k training iterations. The network refinement process spans 300 k training iterations in total.

It should be noted that our initial experiment with EL led to poor generalization and safety performance. In the results we will present, adaptive activations [18] and adaptive learning rates are implemented, in addition to the use of a larger computational budget, to slightly improve the performance, which still falls short of that of HL.

TABLE II
GENERALIZATION AND SAFETY PERFORMANCE (COLLISION RATE) ON COMPLETE-INFORMATION GAMES

Test Domain	Player Types	Learning Method	Metrics		
			$\lvert\vartheta - \hat{\vartheta}\rvert \downarrow$	$\lvert u - \hat{u}\rvert \downarrow$	Col.% ↓
$\mathcal{X}_{GT}$	(a, a)	HL	**0.46**	**0.09 ± 0.10**	**0.00%**
		VH	4.17	0.34 ± 0.19	0.67%
		EL	28.30	0.85 ± 3.92	42.3%
		SL	0.57	0.12 ± 0.36	1.67%
		PINN	3.39	0.96 ± 4.19	84.8%
	(a, na)	HL	**9.43**	**0.49 ± 3.55**	3.50%
		VH	79.35	1.10 ± 5.42	**0.50%**
		EL	123.79	2.24 ± 20.8	42.7%
		SL	10.58	0.54 ± 3.92	4.50%
		PINN	15.33	1.27 ± 7.16	83.3%
	(na, na)	HL	**1.00**	**0.04 ± 0.03**	**1.33%**
		VH	21.76	0.34 ± 1.33	8.50%
		EL	130.53	0.66 ± 5.66	16.5%
		SL	3.49	0.10 ± 0.46	4.33%
		PINN	114.67	1.88 ± 13.72	83.5%
$\mathcal{X}_{XP}$	(a, a)	HL	**0.41**	**0.09 ± 0.08**	**0.20%**
		VH	2.03	0.20 ± 0.07	**0.20%**
		EL	11.93	0.34 ± 1.62	19.0%
		SL	0.69	0.17 ± 0.28	**0.20%**
		PINN	1.54	0.37 ± 1.88	35.2%
	(a, na)	HL	**17.39**	**0.46 ± 3.17**	**0.10%**
		VH	32.64	0.57 ± 2.71	0.20%
		EL	62.62	0.96 ± 8.64	10.5%
		SL	19.01	0.56 ± 3.09	0.60%
		PINN	19.57	0.58 ± 3.89	31.3%
	(na, na)	HL	**1.80**	**0.10 ± 0.12**	**0.00%**
		VH	11.54	0.24 ± 0.68	6.40%
		EL	63.73	0.41 ± 3.02	2.33%
		SL	4.25	0.30 ± 0.72	2.20%
		PINN	60.39	0.95 ± 7.31	36.0%

HL, VH, EL, SL, PINN are for HL, VH, epigraphical, supervised, and vanilla PINN methods, respectively.

1) Results for Complete-Information Games: We generate a separate set of 600 ground truth trajectories for each of the four player type configurations by solving BVPs, with initial states uniformly sampled from $\mathcal{X}_{GT}$. To evaluate generalization performance, we measure the mean absolute errors (MAEs) of value and control input predictions, denoted by $|\vartheta - \hat{\vartheta}|$ and $|u - \hat{u}|$, respectively, across the test trajectories. For safety performance, we use the learned value networks to compute the players' closed-loop control inputs and the state trajectories. From all resulting trajectories computed based on test initial states, we report the percentage of collisions that are *avoidable* according to BVP solutions. The performance results are summarized in Table II, where we averaged the performance of (a, na) and (na, a) due to their symmetry. Sample trajectories for (a, a) are shown in Fig. 3.

To further evaluate the out-of-distribution performance of supervised and HL, we repeat the tests using 500 uniformly sampled initial states in $\mathcal{X}_{XP} := [15, 30]\text{m} \times [18, 25]\text{m/s}$. The results are summarized in the same table and figure. In both tests, the hybrid method demonstrates the best generalization and safety performance. Notably, the vanilla PINN exhibits poor generalization due to value discontinuity. EL performs only better than vanilla PINN in regard of safety. Further inspection shows that EL can actually identify the backward reachable sets (i.e., unsafe zones) well, see Fig. 4. To elaborate, given

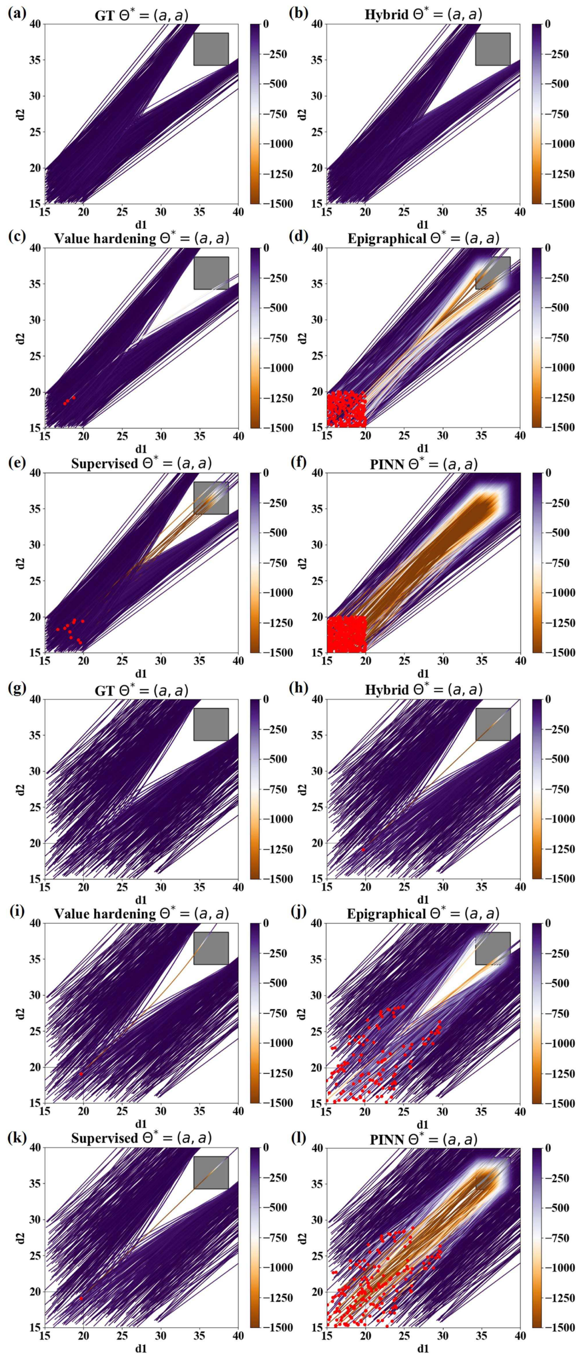


Fig. 3. (a), (g) Ground truth trajectories (projected to d_1-d_2) for $\mathcal{X}_{GT}$ and $\mathcal{X}_{XP}$, respectively. (b)–(f), (h)–(l) Trajectories generated using hybrid, VH, epigraphical, supervised, and vanilla PINN methods under $\mathcal{X}_{GT}$ and $\mathcal{X}_{XP}$, respectively. Color: Actual equilibrial values of player 1 along the trajectories. Trajectories with inevitable collisions are removed for clearer comparison on safety performance. Red dots represent initial states with *avoidable* collisions.

$t \in [0, T]$ and a value network $\hat{V}$ trained for fix player types (a, a), the unsafe zone is defined as $\{\mathbf{x} \in \mathcal{X}_{XP} | \hat{V}(\mathbf{x}, t) > 0\}$. We approximate the ground truth unsafe zone by computing trajectories of sample initial states in $\mathcal{X}_{XP}$ by solving (3) [see Fig. 4(a)]. We compare the ground truth with the approximations from hybrid and EL in Fig. 4(b) and (c). The results here reveal an important limitation of values approximated through PINN:

TABLE III
SAFETY PERFORMANCE (COLLISION RATE) W/ DIFFERENT ACTIVATION FUNCTIONS (W/ L^1) AND BOUNDARY NORMS (W/ tanh)

Method	Activation			Boundary Norm	
	tanh	relu	sin	L^1	L^2
Hybrid	**0.00%**	19.8%	28.7%	**0.00%**	0.4%
Value hardening	0.67%	85.1%	84.6%	-	-
Epigraphical	42.3%	78.8%	89.8%	-	-
Supervised	1.67%	**2.50%**	**19.5%**	-	-
Physics-informed	84.8%	84.0%	84.7%	-	-

High empirical accuracy in characterizing the unsafe zone does not necessarily imply high safety performance, such as in the case of EL. This is potentially because feedback control requires accurate approximation of the value *gradients* instead of the segmentation of value in space–time [see $|u - \hat{u}|$ comparison in Fig. 4(i)]. For the same reason, high safety performance does not imply high accuracy in characterizing the unsafe zone either, such as in the case of HL. We further verify that adding the supervised co-state loss to EL improves its safety performance to be comparable with that of HL. See Section V-C for details.

Ablation studies: We conduct ablation studies to understand the effects of activation functions and the norm of the boundary loss on model performance. Safety results are summarized in Table III for player types (a, a) and using the HL method, with training and testing conducted in $\mathcal{X}_{GT}$. The corresponding trajectories are visualized in Fig. 5. The results indicate that: 1) the choice of the activation function significantly affects the resultant models, with tanh outperforming relu and sin, and 2) the choice of the boundary norm does not have a significant influence.

Remarks: We note that relu networks have been shown to converge to piecewise smooth functions in a supervised setting [51]. However, convergence in the PINN setting requires continuity of the network and its gradient [9], which relu does not offer. Our results are consistent with those of [18], where relu underperforms in solving PDEs. We note, however, that smooth variants of relu, such as gelu, can achieve performance comparable to that of tanh. We also note that while sin does not perform well for Case 1, it achieves comparable performance to tanh in Cases 2–4 (see Sections IV-B, IV-C, and IV-D). This result suggests that fine-tuning of the frequency parameter of sin is necessary and case-dependent [52].

2) Results for Incomplete-Information Games: In games with incomplete information, we investigate the effectiveness of using value networks both for closed-loop control and for belief updates: Each player is uncertain about the types of the other players and therefore holds a belief about their fellow player's types. A belief is a probability distribution over the type space and is updated over time as the player observes new actions from their fellow player. We examine two belief update settings: the first assumes that players have common prior belief and synchronized belief dynamics [53]. In other words, player i knows about player j's uncertainty about player i's type. We refer to players in this setting as "empathetic." The second setting is nonempathetic, where player i falsely assumes that player j has

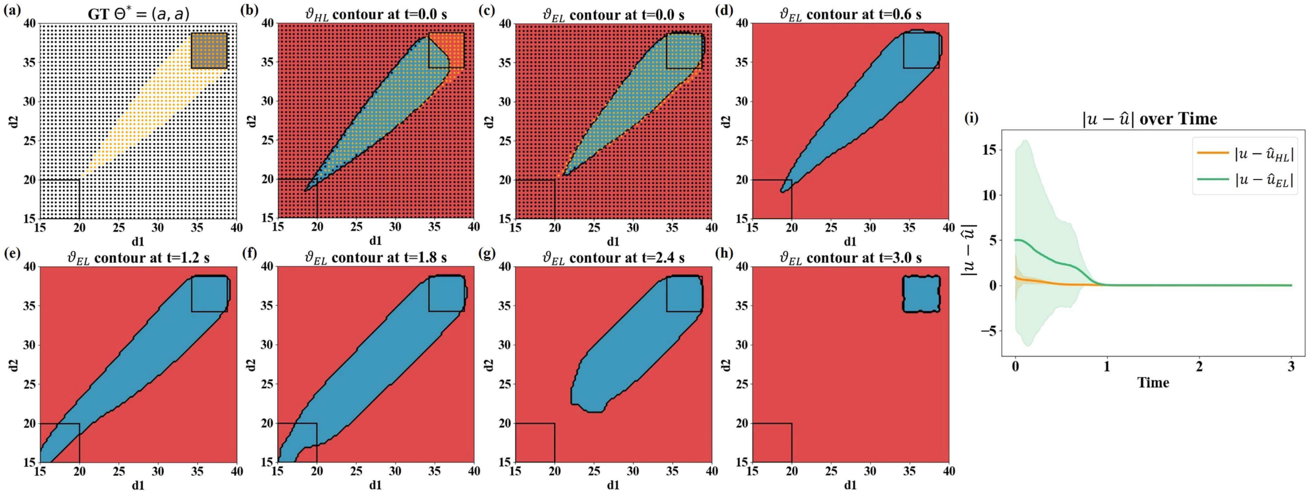
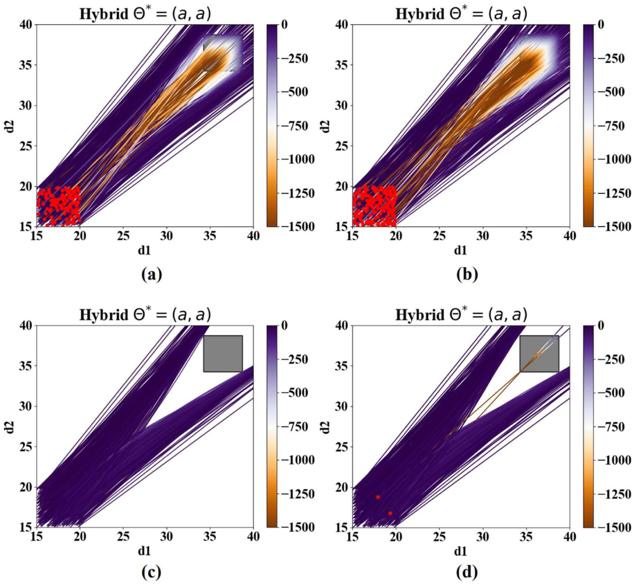


Fig. 4. (a) Ground truth safe/unsafe initial states projected to d_1-d_2 frame, where black dots represent collision-free trajectories while orange dots depict trajectories with collision. (b) Value contours at initial time to classify safe/unsafe zones using HL. (c)–(h) Value contours along time using EL. Blue (red) regions represent unsafe (safe) states. (i) Comparison of mean and standard deviation of $|u - \hat{u}|$ from HL and EL across test trajectories sampled from $\mathcal{X}_{GT}$.



(a)

(b)

(c)

(d)

Fig. 5. Trajectories generated using neural networks with (a) `relu` and (b) `sin` activation functions and using L^1 for boundary norm (for `tanh`, refer to Fig. 3); trajectories generated using (c) L^1- and (d) L^2-norms for the boundary values and using `tanh` for activation. All trajectories are based on HL.

full knowledge about player i's type. We follow [54] to simulate the state and belief dynamics: We model player i to continuously update its belief based on observations, and then determine its next control inputs based on the value network parameterized by the most likely type of player j as well as player i's truth type. We evaluate the efficacy of the hybrid and supervised methods, which achieve the best performance across cases, by measuring their safety performance in incomplete-information settings. The simulations use the same initial states as tests in the complete-information games.

Empathetic belief update: We consider the case where players can take one of the two types: $\Theta = \{a, na\}$. Let $\mathcal{D}_t = \{(\mathbf{x}(k), \mathbf{u}(k))\}_{k=1}^{t}$ be a finite set of observed states and control inputs of both players accumulated up to time t. Let $p_i(t) := \Pr(\theta_i = a \,|\, \mathcal{D}_{t-1})$ be the belief of player j about player i at the beginning of time step t, and $q_i^{\hat{\boldsymbol\theta}}(t) := \Pr(u_i(t) \,|\, \mathbf{x}(t), \hat{\boldsymbol\theta})$ where $\hat{\boldsymbol\theta} \in \{(a,a), (a,na), (na,a), (na,na)\}$ is a point estimate of $\boldsymbol\theta$ based on the current beliefs $\mathbf{p}$.

We assume player i's control policy follows a Boltzmann distribution:

$$q_i^{\hat{\boldsymbol\theta}}(t) = \frac{e^{h_i(\mathbf{x}_i(t),u_i(t),t;\hat{\boldsymbol\theta})}}{\sum_{\mathcal{U}} e^{h_i(\mathbf{x}_i(t),u_i',t;\hat{\boldsymbol\theta})}} \tag{21}$$

where

$$h_i(\mathbf{x}_i(t), u_i(t), t; \hat{\boldsymbol\theta}) = \nabla_{\mathbf{x}_i}\mathbf{f}_i^T \vartheta_i^{\hat{\boldsymbol\theta}} - \tilde{l}_i^{\hat\theta_i} \tag{22}$$

where $\vartheta_i^{\hat{\boldsymbol\theta}}$ is player i's approximated value if the game is played with player types $\hat{\boldsymbol\theta}$, and $\tilde{l}_i^{\hat\theta_i}$ is the instantaneous loss that incorporates the collision penalty if player i is of type $\hat\theta_i$.

Denote the marginal by $q_i^{\hat\theta_i}(t) := \Pr(u_i(t)|\mathbf{x}(t), \hat\theta_i)$, we have

$$q_i^{\hat\theta_i}(t) = q_i^{(\hat\theta_i,a)}(t)p_{-i}(t) + q_i^{(\hat\theta_i,na)}(t)(1-p_{-i}(t)). \tag{23}$$

Given the observations $\mathcal{D}_t$, p_i follows a Bayes update:

$$p_i(t+1) = \frac{q_i^a(t)p_i(t)}{q_i^a(t)p_i(t) + q_i^{na}(t)(1-p_i(t))}. \tag{24}$$

Remarks:

1) If any element of $\mathbf{p}(t)$ is mistakenly assigned a zero probability, this mistake cannot be corrected in future updates. To address this, we modify $\mathbf{p}(t)$ using

$$\mathbf{p}(t) \Leftarrow (1-\epsilon)\mathbf{p}(t) + \epsilon\mathbf{p}(0) \tag{25}$$

before its next update and set the learning rate $1 - \epsilon = 0.95$. $\mathbf{p}(0)$ represents the initial belief.

TABLE IV
COLLISION RATE IN UNCONTROLLED INTERSECTIONS WITH INCOMPLETE INFORMATION: E - EMPATHETIC, NE - NONEMPATHETIC

Belief update model	Initial belief	True type	Hybrid	Supervised
(e,e)	(a,a)	(a,a)	**0.00%**	**0.00%**
(ne,ne)	(a,a)	(a,a)	**0.00%**	**0.00%**
(e,e)	(na,na)	(na,na)	**0.67%**	6.67%
(ne,ne)	(na,na)	(na,na)	**0.67%**	6.67%
(e,e)	(a,a)	(na,na)	**2.00%**	2.67%
(ne,ne)	(a,a)	(na,na)	**2.67%**	**2.67%**
(e,e)	(na,na)	(a,a)	**2.00%**	8.00%
(ne,ne)	(na,na)	(a,a)	**2.67%**	4.00%

2) To make (21) more tractable, we discretize the space of control inputs as $\mathcal{U} := \{-5, -4, \ldots, 0, \ldots, 10\}$m/s^2. In addition, we used discrete time steps with a time interval of 0.05 s to simulate the interactions.
3) We test two settings of initial beliefs. In the first setting, each player believes that the other player has a probability of 80% of being aggressive; in the second setting, the probability is 20%. These initial beliefs correspond to $\mathbf{p}(0) = (0.8, 0.2)$ and $\mathbf{p}(0) = (0.2, 0.8)$, respectively. While a more extensive test over the initial belief space could be interesting, it is beyond the scope of this study.

Nonempathetic belief update: A nonempathetic player updates its belief about the other player's type by assuming that his type is known. Let the true types be $\boldsymbol{\theta}^*$. Player $-i$'s belief about player i's type now becomes a conditional $p_i'(t) := \Pr(\theta_i = a|\mathcal{D}_{t-1}, \theta^*_{-i})$. The Bayes update of $p_i'(t)$ follows:

$$p_i'(t+1) = \frac{q_i^{(a,\theta^*_{-i})}(t)p_i'(t)}{q_i^{(a,\theta^*_{-i})}(t)p_i'(t) + q_i^{(na,\theta^*_{-i})}(t)(1-p_i'(t))}. \quad (26)$$

Consequently, each player starts with its own belief, which are not necessarily common during the interaction.

Control policy: Given the beliefs $p_i(t)$ or $p_i'(t)$, player $-i$ finds the most likely type of player i. The control policy of player i is determined by the value function corresponding to $(\theta_i^*, \hat{\theta}_{-i})$. It is worth noting that player i employs a policy that is consistent with its true type, even if player j holds an incorrect belief about player i, which player i acknowledges in the empathetic setting. This setup allows players to signal their own types through their actions.

Simulation results: We present simulated interactions between two players at an uncontrolled intersection in an incomplete-information setting. The simulations are performed on a grid that enumerates the following settings: (empathetic, nonempathetic) × (correct prior, wrong prior) × (aggressive, nonaggressive), where both players have identical settings to limit the scope. For each setting, we evaluate the safety performance of the value approximation models learned through the hybrid and supervised methods using test samples from $\mathcal{X}_{GT}$. Table IV summarizes the results that the hybrid models have a lower chance of collision than the supervised ones under all settings.

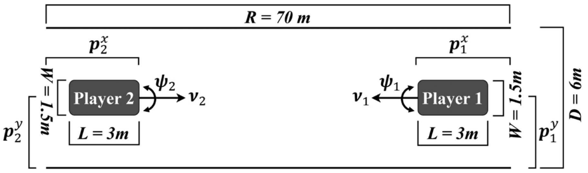


Fig. 6. Narrow road collision avoidance setup with two players.

B. Case 2: Narrow Road Collision Avoidance

Experiment setup: The schematic is depicted in Fig. 6, where the states of player i consist of its location (p_i^x, p_i^y), orientation (ψ_i), and speed (v_i), denoted as $x_i := [p_i^x, p_i^y, \psi_i, v_i]^T$. The system dynamics is modeled using a unicycle model

$$\begin{bmatrix} \dot{p}_i^x \\ \dot{p}_i^y \\ \dot{\psi}_i \\ \dot{v}_i \end{bmatrix} = \begin{bmatrix} v_i \cos(\psi_i) \\ v_i \sin(\psi_i) \\ \omega_i \\ u_i \end{bmatrix} \quad (27)$$

where $\omega_i \in [-1, 1]$rad/s and $u_i \in [-5, 10]$m/s^2 are control inputs that represent angular velocity and acceleration, respectively. The instantaneous loss incorporates control effort

$$l_i(x_i, u_i; \theta_i) = k\omega_i^2 + u_i^2 \quad (28)$$

where $k = 100$. The state constraint is

$$c_i(\mathbf{x}_i) = \eta - \sqrt{((R - p_2^x) - p_1^x)^2 + (p_2^y - p_1^y)^2} \quad (29)$$

where $\eta = 1.5$ m and $R = 70$ m. $c_i(\cdot) > 0$ is considered as a collision incident. The parameter R represents the length of the road, and $\eta = 1.5$ m is the collision threshold. The terminal loss is designed to encourage players to move along the lane and restore nominal speed

$$g_i(x_i) = -\mu p_i^x(T) + (v_i(T) - \bar{v})^2 + (p_i^y(T) - \bar{p}^y)^2 \quad (30)$$

where $\mu = 10^{-6}$, $\bar{v} = 18$ m/s, $\bar{p}^y = 3$ m, and $T = 3$ s. For hybrid, VH, and vanilla PINN, we treat the state constraint as a penalty in a modified instantaneous loss

$$\tilde{l}_i(\mathbf{x}_i, \omega_i, u_i) = k\omega_i^2 + u_i^2 + b\sigma(\mathbf{x}_i, \eta) \quad (31)$$

where the penalty function is defined as

$$\sigma(\mathbf{x}_i, \eta) = (1 + \exp(-\gamma c_i(\mathbf{x}_i)))^{-1}.$$

The parameter b is set to 10^4 to impose a high penalty on collision, while $\gamma = 5$ is a shape parameter.

Data: For SL, we generate 1.45 k ground truth trajectories by uniformly sampling initial states from $\mathcal{X}_{GT} := [15, 20]\text{m} \times [2.25, 3.75]\text{m} \times [-\pi/180, \pi/180]\text{rad} \times [18, 25]\text{m/s}$, resulting in a total of 89.9 k data points. For vanilla PINN and its VH variant, we uniformly sample 122 k states from $\mathcal{X}_{HJ} := [15, 90]\text{m} \times [0, 6]\text{m} \times [-0.15, 0.18]\text{rad} \times [18, 25]\text{m/s}$. For HL, we generate 1 k ground truth trajectories (62k data points) by uniformly sampling initial states from $[15, 20]\text{m} \times [2.25, 3.75]\text{m} \times [-\pi/180, \pi/180]\text{rad} \times [18, 25]$ m/s and sample 60k states uniformly from $[15, 90]\text{m} \times [0, 6]\text{m} \times [-0.15, 0.18]\text{rad} \times [18, 25]$m/s. For EL, we introduce an auxiliary state z_i with a range of $[-9 \times 10^{-5}, 300]$ to account for both the best- and worst-case

TABLE V
COLLISION RATE W/ DIFFERENT ACTIVATION FUNCTIONS

Test Domain	Activation Functions	Learning Method				
		HL	VH	EL	SL	PINN
$\mathcal{X}_{GT}$	`tanh`	**1.67%**	95.2%	48.3%	2.17%	81.3%
	`relu`	**65.2%**	98.2%	70.5%	67.7%	83.8%
	`sine`	**1.67%**	98.5%	69.7%	3.17%	98.3%

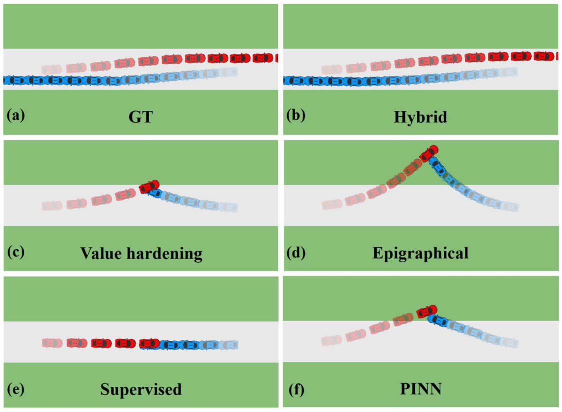


Fig. 7. Narrow road collision avoidance visualization. (a) Ground truth safe trajectory. Transparency reduces along time. (b)–(f) Trajectories generated using hybrid, VH, epigraphical, supervised, and vanilla PINN models, respectively.

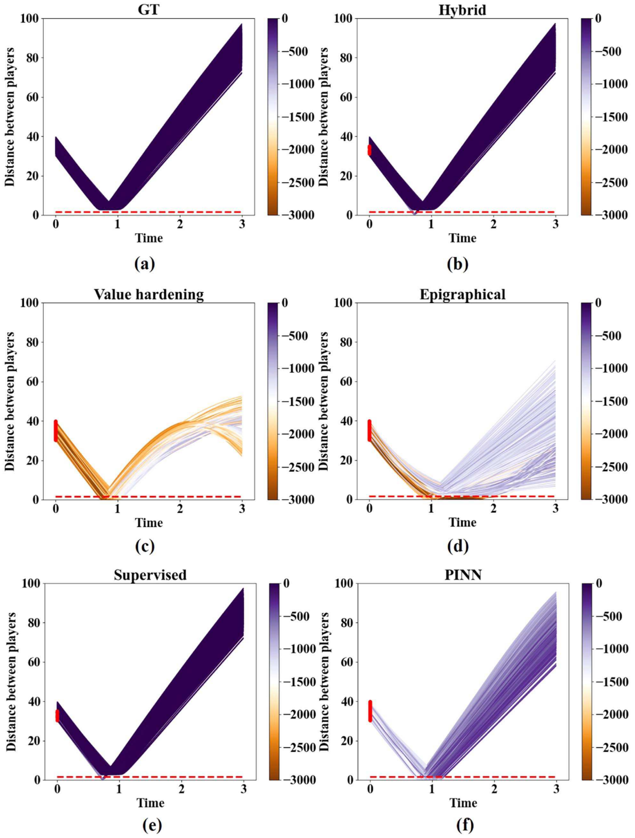
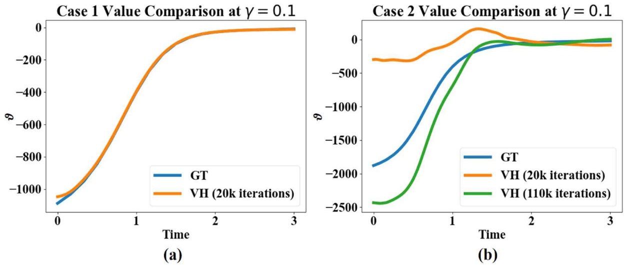


Fig. 8. (a) Ground truth distance between players over time for $\mathcal{X}_{GT}$. (b)–(f) Distance between players over time using hybrid, VH, epigraphical, supervised, and vanilla PINN under $\mathcal{X}_{GT}$, respectively. Red dashed line represents the threshold distance for collision.



Model Training Time Cost	Case 1	20k iterations for each hardening step	Case 2	110k iterations for each hardening step
		80 min		450 min

Fig. 9. (a) VH uses 20k training iterations for each hardening step, for a total of ten steps to converge to ground truth in Case 1. (b) VH uses 20k/110k training iterations for each hardening step, for a total of ten steps to converge to ground truth in Case 2. Compared to Case 1, VH takes around 5.6 times longer to converge to the ground truth in Case 2.

scenarios. We employ the same settings as in Case 1 for the remaining aspects of the experiment.

Training: For vanilla PINN, we pretrain the networks for 10 k iterations using 122 k uniformly sampled boundary states and then train them for 430 k iterations. For VH, we use 8.8 k training iterations for each hardening step and a total of 50 steps for a fair comparison. The remaining settings are the same as those in Case 1.

Results: We evaluate the safety performance of the methods on a test set of 600 ground truth collision-free trajectories with initial states drawn from $\mathcal{X}_{GT}$. The results are summarized in Table V, and the distance between players during interactions is visualized in Fig. 8. Similar to Case 1, the HL method outperformed the others. Fig. 7 demonstrates interactions from the same initial state in which only the hybrid method avoids a collision.

We notice that VH fails to generalize well in this higher dimensional case (and in Case 3). We hypothesize that vanilla PINN, which VH is based on, is less scalable in compute than hybrid PINN as the state dimensionality increases.

While the relationship between learning dynamics of PINN and state dimensionality is yet to be understood, here we empirically show that VH PINN requires significantly higher compute to converge in Case 2 due to its higher state dimensionality. To make this empirical study more tractable, we use a mildly softened collision penalty with $\gamma = 0.1$ in both Case 1 and 2. We uniformly sample 122k states from X_{HJ} and train the model using ten hardening steps until γ reaches 0.1. To visualize the convergence, in Fig. 9 we show the value along a randomly chosen equilibrium trajectory derived from PMP for Case 1 (left) and Case 2 (right). We can see that by 20k iterations, VH already converges to the ground truth in Case 1, while in Case 2, convergence requires more than 110 k iterations.

C. Case 3: Double-Lane Change

Experiment setup: The schematic is shown in Fig. 10, depicting the states of player i as its location (p_i^x, p_i^y), orientation

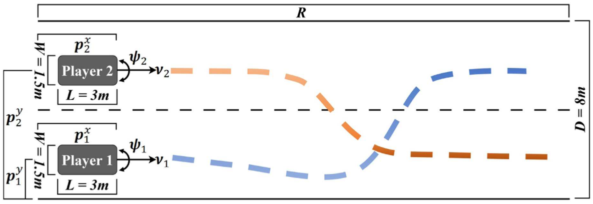


Fig. 10. Double-lane change setup with two players.

(ψ_i), and speed (v_i). $x_i := [p_i^x, p_i^y, \psi_i, v_i]^T$. The dashed blue and orange color (with increasing transparency along the *x*-axis) in the figure represents desired trajectories for both players. We use the same unicycle model and instantaneous loss as in Section IV-B. The terminal loss is set to incentivize players to stay within their respective lanes and regain the nominal speed

$$\begin{aligned} g_i(x_i) = & -\mu p_i^x(T) + (p_i^y(T) - \bar{p}_i^y)^2 \\ & + (v_i(T) - \bar{v})^2 + \kappa(\psi_i(T) - \bar{\psi})^2 \end{aligned} \tag{32}$$

where $\mu = 10^{-6}$, $\kappa = 100$, $\bar{p}_1^y = 6$m for player 1 and $\bar{p}_2^y = 2$m for player 2, $\bar{v} = 18$m/s, $\bar{\psi} = 0$ rad, and $T = 4$s.

Data: In the case of SL, we generate 1.45k ground truth trajectories by uniformly sampling initial states from the set $\mathcal{X}_{GT}^1 := [0, 3]\text{m} \times [1.25, 2.75]\text{m} \times [-\pi/180, \pi/180]\text{rad} \times [18, 25]\text{m/s}$ for player 1, and $\mathcal{X}_{GT}^2 := [0, 3]\text{m} \times [5.25, 6.75]\text{m} \times [-\pi/180, \pi/180]\text{rad} \times [18, 25]\text{m/s}$ for player 2, resulting in a total of 118.9k data points. For vanilla and VH PINN, we uniformly sample 162k states from the set $\mathcal{X}_{HJ}^1 := [0, 95]\text{m} \times [0, 6]\text{m} \times [-0.15, 0.13]\text{rad} \times [17, 26]\text{m/s}$ for player 1, and $\mathcal{X}_{HJ}^2 := [0, 95]\text{m} \times [2, 8]\text{m} \times [-0.13, 0.15]\text{rad} \times [17, 26]\text{m/s}$ for player 2. In the case of HL, we generate 1k ground truth trajectories (82 k data points) by uniformly sampling initial states from $\mathcal{X}_{GT}^1$ for player 1, and $\mathcal{X}_{GT}^2$ for player 2. In addition, we sample 80 k states uniformly from $\mathcal{X}_{HJ}^1$ for player 1, and $\mathcal{X}_{HJ}^2$ for player 2. For EL, we initially gather a sample of 200 k states from $\mathcal{X}_{HJ}$ to ensure adherence to the boundary condition. We set the range of the auxiliary state z_i as $[-9.5 \times 10^{-5}, 400]$. All other settings follow Case 1.

Training: In this experiment, we employ the Adam optimizer with a constant learning rate of 1×10^{-4}. For vanilla PINN, we initiate the pretraining phase with 10k iterations, utilizing 162 k boundary states uniformly sampled. Subsequently, we continue with the training phase, performing 350 k iterations. For VH, we set the training duration for each hardening step to 7.2 k iterations, completing a total of 50 steps to ensure a fair comparison. All other settings remain consistent with those of Case 1.

Results: We assess the safety performance on a test set comprising 600 ground truth collision-free trajectories. These trajectories are generated by sampling initial states from $\mathcal{X}_{GT}$. The results are summarized in Table VI, while the interaction distances between players are visualized in Fig. 11. Similar to Cases 1 and 2, the hybrid method demonstrates superior performance compared to the others. Similar to Case 2, VH fails to generalize effectively within a computational budget similar to HL. Fig. 12 shows interaction trajectories starting from one particular initial state where the hybrid method achieves safe interaction while the others fail.

TABLE VI
COLLISION RATE W/ DIFFERENT ACTIVATION FUNCTIONS

Test Domain	Activation Functions	Learning Method				
		HL	**VH**	**EL**	**SL**	**PINN**
$\mathcal{X}_{GT}$	`tanh`	**0.00%**	23.0%	46.2%	0.33%	30.2%
	`relu`	1.33%	40.3%	61.0%	**0.00%**	52.5%
	`sine`	**0.50%**	11.2%	48.5%	1.00%	17.3%

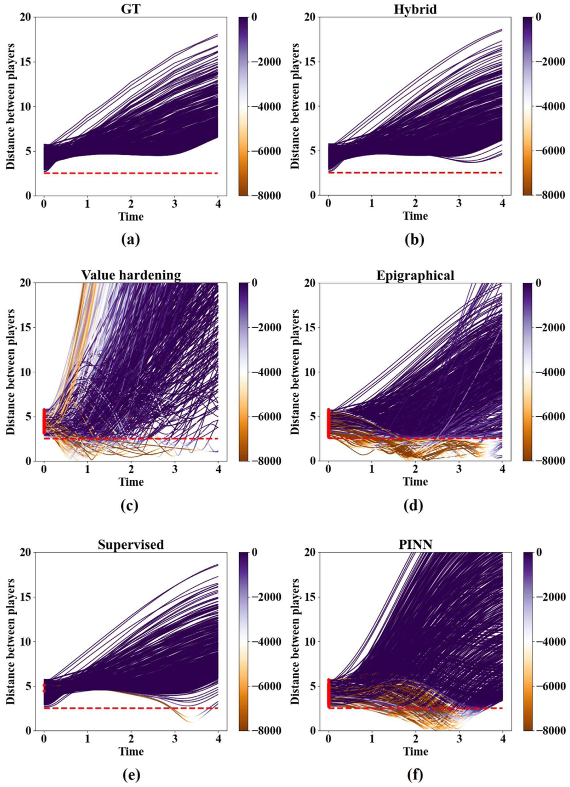


Fig. 11. (a) Ground truth distance between players over time for $\mathcal{X}_{GT}$. (b)–(f) Distance between players over time using hybrid, VH, epigraphical, supervised, and vanilla PINN models under $\mathcal{X}_{GT}$, respectively. Red dashed line represents the threshold distance for collision.

D. Case 4: Two-Drone Collision Avoidance

Experiment setup: In this experiment, we consider that the states of player i consist of its location (p_i^x, p_i^y, p_i^z), and speed (v_i^x, v_i^y, v_i^z), denoted as $x_i := [p_i^x, p_i^y, p_i^z, v_i^x, v_i^y, v_i^z]^T$. We use the flight dynamics (in the near-hover regime, at zero yaw with respect to a global coordinate frame) described in [42]

$$\begin{bmatrix} \dot{p}_i^x \\ \dot{p}_i^y \\ \dot{p}_i^z \\ \dot{v}_i^x \\ \dot{v}_i^y \\ \dot{v}_i^z \end{bmatrix} = \begin{bmatrix} v_i^x \\ v_i^y \\ v_i^z \\ g\tan(\theta_i) \\ -g\tan(\phi_i) \\ \tau_i - g \end{bmatrix} \tag{33}$$

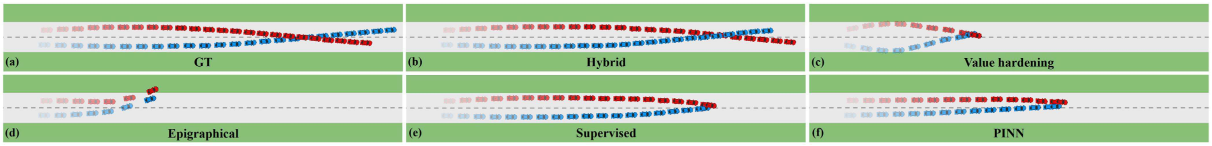


Fig. 12. Double-lane change visualization. (a) Ground truth safe trajectory. Transparency reduces along time. (b)–(f) Trajectories generated using hybrid, VH, epigraphical, supervised, and vanilla PINN models, respectively.

where the tracking control $u_i = (\theta_i, \phi_i, \tau_i)$ corresponds to roll, pitch and thrust. In this experiment, $\theta_i \in [-0.05, 0.05]$rad, $\phi_i \in [-0.05, 0.05]$rad, $\tau_i \in [7.81, 11.81]$m/s^2, and $g = 9.81$ m/s^2. Note that we have assumed a zero yaw angle for the quadrotor. The instantaneous loss considers the control effort and the collision penalty

$$\tilde{l}_i(\mathbf{x}_i, \omega_i, u_i) = k_\theta \tan^2(\theta_i) + k_\phi \tan^2(\phi_i) + (\tau_i - g)^2 + b\sigma(\mathbf{x}_i, \eta) \tag{34}$$

where the penalty function is defined as

$$\sigma(\mathbf{x}_i, \eta) = (1 + \exp(\gamma(S - \eta)))^{-1}$$

$$S = \sqrt{((R_x - p_2^x) - p_1^x)^2 + ((R_y - p_2^y) - p_1^y)^2 + (p_2^z - p_1^z)^2}.$$

$b = 10^4$ and $\gamma = 5$. In addition, the parameters $R_x = 5$ m and $R_y = 5$ m are used to transform the coordinate positions of the two players along the x and y axes, respectively. The values of $k_\theta = 100$ and $k_\phi = 100$ determine the tradeoff between control effort for roll, pitch, and thrust. Furthermore, $\eta = 0.9$ m represents the collision threshold. The terminal loss is set to encourage players to move along their respective x and y directions, to return to 0m on the z-axis, and to remain stationary when the simulation is complete

$$\begin{aligned} g_i(x_i) = & -\mu p_i^x(T) - \mu p_i^y(T) + (p_i^z(T) - \bar{p}_i^z)^2 \\ & + (v_i^x(T) - \bar{v}_i^x)^2 + (v_i^y(T) - \bar{v}_i^y)^2 + (v_i^z(T) - \bar{v}_i^z)^2 \end{aligned} \tag{35}$$

where $\mu = 10^{-6}, \bar{p}_i^z = 0$m, $\bar{v}_i^x = \bar{v}_i^y = \bar{v}_i^z = 0$m/s, and $T = 4$s. In this case study, we only compare the generalization and safety performance between the hybrid and the supervised methods, and use vanilla PINN as a baseline. VH and EL are dropped from the comparison since they do not generalize well in high-dimensional cases as we found in Sections IV-B and IV-C.

Data: In the case of SL, we generate 1.25k ground truth trajectories by uniformly sampling initial states from the set $\mathcal{X}_{GT} := [0, 1]\text{m} \times [0, 1]\text{m} \times [-0.1, 0.1]\text{m} \times [2, 4]\text{m/s} \times [2, 4]\text{m/s} \times [0, 0.1]\text{m/s}$, resulting in a total of 102.5k data points. For vanilla PINN, we uniformly sample 162k states from the set $\mathcal{X}_{HJ} := [0, 15.5]\text{m} \times [0, 15.5]\text{m} \times [-1.8, 2]\text{m} \times [0.3, 4.5]\text{m/s} \times [0.3, 4.5]\text{m/s} \times [-1.8, 1.8]\text{m/s}$. In the case of HL, we generate 1 k ground truth trajectories (82 k data points) by uniformly sampling initial states from $\mathcal{X}_{GT}$. In addition, we sample 80k states uniformly from $\mathcal{X}_{HJ}$.

Training: We use the Adam optimizer with a fixed learning rate of 1×10^{-4}. For vanilla PINN, we pretrain the networks for 100k iterations using 162k uniformly sampled boundary states and subsequently train them for an additional 400k iterations.

TABLE VII
COLLISION RATE W/ DIFFERENT ACTIVATION FUNCTIONS

Test Domain	Activation Functions	Learning Method		
		HL	**SL**	**PINN**
$\mathcal{X}_{GT}$	tanh	**0.00%**	0.17%	75.8%
	relu	34.0%	**0.67%**	76.2%
	sine	**0.00%**	0.17%	75.7%

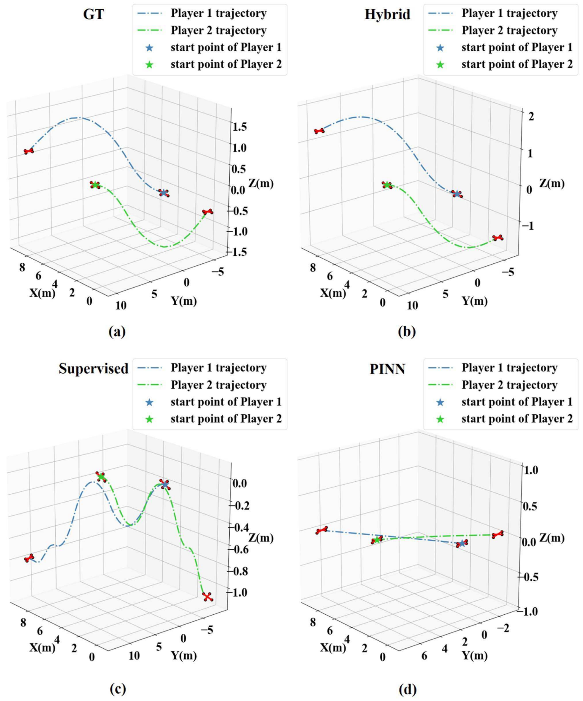
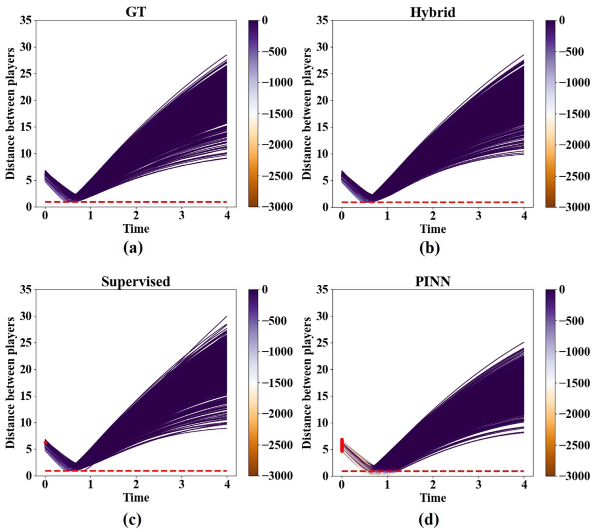


Fig. 13. Two-drone collision avoidance visualization. (a) Ground truth safe trajectory. (b)–(d) Trajectories generated using hybrid, supervised, and vanilla PINN models, respectively.

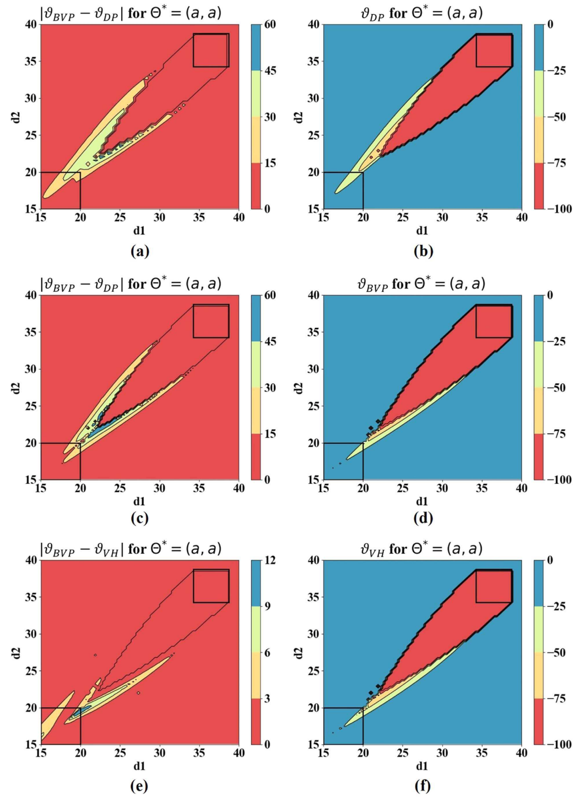
The remaining settings for this experiment align with those used in Case 1.

Results: We assess the safety performance on a test set comprising 600 ground truth collision-free trajectories. These trajectories are generated by uniformly sampling initial states from $\mathcal{X}_{GT}$. The results are summarized in Table VII, while the interaction distances between players are visualized in Fig. 14. Similar to the first three cases, the HL method demonstrates superior performance compared to the other methods. Fig. 13 visualizes the trajectories starting from a particular initial state where the hybrid method achieves safe interaction, while the other baselines yield collisions and undesired trajectories.



(a) (b) (c) (d)

Fig. 14. (a) Ground truth distance between players over time for $\mathcal{X}_{GT}$. (b)–(d) Distance between players over time using hybrid, supervised, and vanilla PINN models under $\mathcal{X}_{GT}$, respectively. Red dashed line represents the threshold distance for collision.



(a) (b) (c) (d) (e) (f)

Fig. 15. (a), (c) Difference $|\vartheta_{\text{BVP}} - \vartheta_{\text{DP}}|$ in (d_1, d_2) frame with $v_{1,2} = 18$ m/s at $t = 0$ using DP spatial resolution $dx = 0.5$ and $dx = 0.3$, respectively. (b), (d) Numerical solutions ϑ obtained through DP and BVP solver, respectively. (e) Difference $|\vartheta_{\text{BVP}} - \vartheta_{\text{VH}}|$ in (d_1, d_2) frame with $v_{1,2} = 18$ m/s at $t = 0$. (f) Approximated solutions ϑ obtained through HJI-based learning approach-VH.

V. Discussion

A. Safety Guarantee

We note that our method does not provide safety certificate in its current form and discuss potential future directions. *Policy certification:* For fixed-time differential games, it is possible to consider the interaction, i.e., the interchanging computation of actions (via approximated value gradients) and states (via an ODE solver), as a neural-network controlled system (NNCS), for which certification tools emerge [55], [56]. It should be noted that reachability analysis of NNCS is currently limited to small state space (due to the exponential growth in the approximation polynomial degree with respect to the state space dimensionality [55]), small Lipschitz constant (due to linear growth of approximation error with respect to the Lipschitz constant of the neural network), and small network sizes (e.g., four layers each with 20 neurons in [55]). Specifically, reachability analysis (e.g., forward [57], backward [58], or automated [59] methods) for NNCS can be applied to a 6-D quadrotor system. However, these analyses are limited to small network sizes and face challenges in achieving real-time verification for each closed-loop policy using trained models. *Post-hoc state-constrained control:* When policy certification becomes intractable, an alternative could be to use a linear–quadratic reformulation of the game with conservative state constraint approximation for online computation of policies. In this setting, the value approximation network offers good initial policy guesses. This method trades off overall performance of policies in attaining Nash equilibrium for a computationally tractable safety guarantee. A recent study [60] explores online value approximations with safety guarantees for zero-sum games, yet it does not cover general-sum games and safe reachable analysis of online policy computation. Hsu et al. [61] proposed a unified framework to review the existing safety analysis approaches for closed-loop policy.

B. Consistency Between BVP and HJI Values

Recognizing that PMP is only necessary conditions for local optimality [62] while HJ solutions satisfy global optimality, we adopted multiple initial guesses to solve BVPs in order to seek global solutions. This treatment was applied to all case studies. Taking Case 1 as an example, we initialize the BVP solver with four state trajectories that follow constant control inputs: $\{(-5, -5)\text{m/s}^2, (-5, 10)\text{m/s}^2, (10, -5)\text{m/s}^2, (10, 10)\text{m/s}^2\}$. These trajectories represent four categories of interactions where each of the players either yield or accelerate through the intersection, and potentially lead to different equilibria. To address the issue with multiplicity of equilibrium, we choose the one that yields the best sum of values (i.e., Pareto optimal Nash equilibrium).

In the following, we empirically show that this treatment leads to consistent value landscapes between BVP and HJI. A visualization of the comparison uses the value contour from Case 1, projected to (d_1, d_2) with fixed $v_{1,2} = 18$ m/s and $t = 0$ and with player types $(\mathtt{a}, \mathtt{a})$. See Fig. 15.

To compute values from the BVP solver, we sample initial states from $[15, 40]\text{m} \times [15, 40]\text{m}$ with fixed $v_{1,2} = 18$ m/s and with the spatial resolution $dx = 0.3$m. For each initial state, we solve the BVP and compute ϑ at $t = 0$.

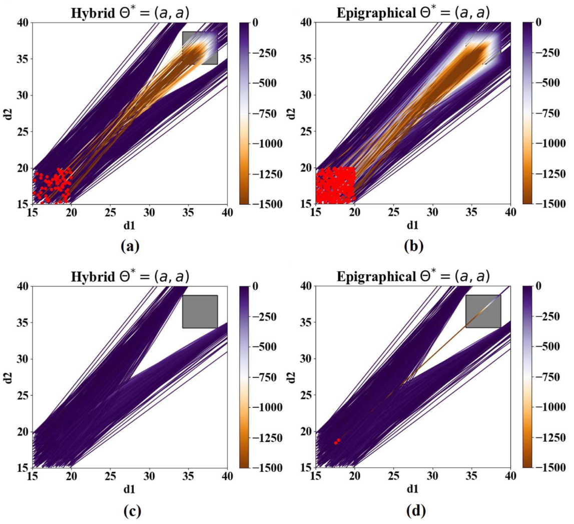


Fig. 16. (a), (c) Trajectories generated using HL w/o and w co-state loss under $\mathcal{X}_{GT}$. (b), (d) Trajectories generated using EL w/o and w co-state loss under $\mathcal{X}_{GT}$.

For HJI values, we consider two approximations of the ground truth. First, we extend an existing HJ PDE solver [22] from zero-sum to general-sum. Since this DP solver has limited scalability with respect to state dimensions (up to 6-D as demonstrated in [22]), we only applied the solver to Case 1 where values are 5-D. The value difference between BVP and the DP solver, $|\vartheta_{\text{BVP}} - \vartheta_{\text{DP}}|$, is visualized for two DP spatial resolution settings: $dx = 0.5$ and $dx = 0.3$ in Fig. 15(a) and (c), respectively. $dx = 0.3$ is the highest resolution supported by our computing hardware. Note that values reported do not take into account constraint violation penalty, and unsafe states are assigned a constant value of -100. We observe that the difference $|\vartheta_{\text{BVP}} - \vartheta_{\text{DP}}|$ decreases as the resolution improves, and expect the trend to continue if the resolution were to be further increased. Since the DP solver has limited resolution, we resort to VH as a second approximation of the ground truth because it achieves relatively good generalization performance without using supervisory data in Case 1. Fig. 15(e) visualizes the value difference between BVP and VH, $|\vartheta_{\text{BVP}} - \vartheta_{\text{VH}}|$, which again shows the similarity between the two.

C. Importance of the Co-State Loss for Safety Performance

Finally, we provide details on the empirical study where we show that achieving good safety performance requires accuracy co-state (value gradient) approximation. While the epigraphical technique facilitates smooth value approximation, it does not explicitly enforce small approximation errors on co-states. In the following, we conducted a comparison between the HL and the EL methods with identical training settings for Case 1: During their training, HL and EL uniformly sample 1 k ground-truth trajectories (62 k data points) in $\mathcal{X}_{GT}$ and 60 k states in $\mathcal{X}_{HJ}$. In addition, we uniformly sample the auxiliary state $z_i \in [-1.05 \times 10^{-4}, 300]$ for EL. Both methods are solved using the Adam optimizer with a fixed learning rate of 2×10^{-5}. We pretrain the networks for 100 k iterations using the supervised data and combine the supervised data with states sampled from an expanding time window starting from the terminal time to minimize $L_1 + L_2$ [(5) and (6)] and $L_2 + L_3$ [(6) and (14)] with 100k iterations for HL and EL, respectively. We show that the safety performance of EL is still worse than HL when using supervised data *without* the co-state loss in L_2 [see Fig. 16(b)], and its safety performance significantly improves when the co-state loss is considered [see Fig. 16(d)]. On the other hand, Fig. 16(a) shows that HL performs worse *without* the co-state loss. Hence, we conjecture that ensuring good safety performance requires not only small approximation errors for values but also for co-states.

VI. Conclusion

We proposed an HL method that combines the strengths of SL and vanilla PINN to approximate discontinuous value functions as solutions to two-player general-sum differential games. The proposed method yields better generalization and safety performance than an array of baselines, including SL, vanilla PINN, VH, and EL, when using the same computational budget. We empirically demonstrate that the co-state loss is the key factor for high safety performance, and the choice of the activation function and its parameters is crucial to the safety performance of learned models. Finally, all results in this article can be reproduced using our code.[2]

Appendix A
Summary of Acronyms

Table VIII summarizes acronyms used in this article.

TABLE VIII
Acronyms Used Throughout this Article

Acronym	Full Name	Acronym	Full Name
HJI	Hamilton-Jacobi-Isaacs	PINN	Physics-Informed Neural Network
CoD	Curse of Dimensionality	PMP	Pontryagin's Maximum Principle
DP	Dynamic Programming	NNCS	Neural-Network Control System
HRI	Human-Robot Interaction	a	aggressive
MPC	Model Predictive Control	na	non-aggressive
BVP	Boundary Value Problem	e	empathetic
ENO	Essentially Non-Oscillatory	ne	non-empathetic
MAEs	Mean Absolute Errors	GT	Ground Truth
EL	Epigraphical Learning	HL	Hybrid Learning
SL	Supervised Learning	VH	Value Hardening

Appendix B
Proof of Lemma 1 (Following Proofs in [15])

Proof:

i) $\vartheta_i(\mathbf{x}_i, t) - z_i \leq 0$ implies that there exists $\alpha_i \in \mathcal{A}$ such that

$$\int_t^T l_i\left(x_s^{x_i,t,\alpha_i}, \alpha_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha_{-i}}, s\right)\right) ds + g_i\left(x_T^{x_i,t,\alpha_i}\right) - z_i \leq 0 \tag{36}$$

[2] [Online]. Available: https://github.com/dayanbatuofu/Value_Appro_Game

and $c_i(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha_{-i}}) \leq 0$ for $s \in [t,T]$. Thus, there exists α_i such that $V_i(\mathbf{x}_i, z_i, t) \leq 0$.

ii) $V_i(\mathbf{x}_i, z_i, t) \leq 0$ and $c_i(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha_{-i}}) \leq 0$ implies that there exists $\alpha_i \in \mathcal{A}$ such that

$$\begin{aligned}&\int_t^T l_i\left(x_s^{x_i,t,\alpha_i}, \alpha_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha_{-i}}, s\right)\right) ds\\ &+ g_i\left(x_T^{x_i,t,\alpha_i}\right) - z_i \leq 0 \end{aligned} \tag{37}$$

which concludes $\vartheta_i(\mathbf{x}_i, t) - z_i \leq 0$. □

Appendix C
Proof of Lemma 2 (Following Proofs in [15] and [63])

Proof: For any policy α_i and a small step $h > 0$, we can use (7) to derive the following relation (α^*_{-i} represents equilibrium policy for the fellow player of player i):

$$\begin{aligned} V_i(\mathbf{x}_i, z_i, t) =\ & \min_{\alpha_i \in \mathcal{A}} \max\Big\{ \max_{s\in[t,T]} c_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}\right)\\ & g_i\left(x_T^{x_i,t,\alpha_i}\right) - z_i(T)\Big\}\\ =\ & \max\Big\{ \max_{s\in[t,t+h]} c_i(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}})\\ & \max\Big\{ \max_{s\in[t+h,T]} c_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}\right)\\ & g_i\left(x_T^{x_i,t,\alpha_i}\right) - z_i(T)\Big\}\Big\}. \end{aligned}$$

There exists two different policies $\alpha_{i_1}, \alpha_{i_2} \in \mathcal{A}$ such that

$$\alpha_i = \begin{cases} \alpha_{i_1}(s), & s \in [t, t+h]\\ \alpha_{i_2}(s), & s \in (t+h, T]. \end{cases}$$

Then, we have

$$\begin{aligned} V_i(\mathbf{x}_i, z_i, t) =\ & \min_{\alpha_{i_1}\in\mathcal{A}, \alpha_{i_2}\in\mathcal{A}} \max\Big\{ \max_{s\in[t,t+h]} c_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}\right)\\ & \max\Big\{ \max_{s\in[t+h,T]} c_i(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}})\\ & g_i\left(x_T^{x_i,t,\alpha_i}\right) - z_i(T)\Big\}\Big\}\\ =\ & \min_{\alpha_{i_1}\in\mathcal{A}} \max\Big\{ \max_{s\in[t,t+h]} c_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}\right)\\ & \min_{\alpha_{i_2}\in\mathcal{A}} \max\Big\{ \max_{s\in[t+h,T]} c_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}\right)\\ & g_i\left(x_T^{x_i,t,\alpha_i}\right) - z_i(T)\Big\}\Big\}\\ =\ & \min_{\alpha_{i_1}\in\mathcal{A}} \max\Big\{ \max_{s\in[t,t+h]} c_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}\right)\\ & V_i(\mathbf{x}_i(t+h), z_i(t+h), t+h)\Big\}\\ =\ & \min_{\alpha_i\in\mathcal{A}} \max\Big\{ \max_{s\in[t,t+h]} c_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}\right)\\ & V_i(\mathbf{x}_i(t+h), z_i(t+h), t+h)\Big\}. \end{aligned}$$

□

Appendix D
Proof of Theorem 1 (Following Proofs in [15] and [63])

Proof:

1) When $t = T$, V_i is easily satisfied based on definition

$$\begin{aligned} V_i(\mathbf{x}_i, z_i, T) &= \max\Big\{c_i\left(\mathbf{x}_T^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}\right),\\ &\qquad g_i\left(x_T^{x_i,t,\alpha_i}\right) - z_i(T)\Big\}\\ &= \max\left\{c_i(\mathbf{x}_i(T)), g_i(T) - z_i(T)\right\}. \end{aligned} \tag{38}$$

2) Let $W_i \in C^\infty(\mathcal{X} \times \mathbb{R} \times [0,T])$, and assume that $V_i - W_i$ has local maximum at $(\mathbf{x}_i(t_0), z_i(t_0), t_0) \in \mathcal{X} \times \mathbb{R} \times [0, T)$ and $(V_i - W_i)(\mathbf{x}_i(t_0), z_i(t_0), t_0) = 0$, we need to prove

$$\begin{aligned} &\max\Big\{c_i(\mathbf{x}_{t_0}^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}) - W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0)\\ &\nabla_t W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) - \mathcal{H}_i(t_0, \mathbf{x}_i(t_0), z_i(t_0)\\ &\nabla_{\mathbf{x}_i} W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0), \nabla_{z_i} W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0))\Big\}\\ &\geq 0. \end{aligned} \tag{39}$$

Suppose not. Then, there exists $\xi > 0$ and $\tilde{\alpha}_i \in \mathcal{A}$ such that

$$\begin{aligned} &c_i(\mathbf{x}_s^{\mathbf{x}_i,t,\tilde{\alpha}_i,\alpha^*_{-i}}) - W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) \leq -\xi,\\ &\nabla_t W_i(\mathbf{x}_i, z_i, t) + \nabla_{\mathbf{x}_i} W_i(\mathbf{x}_i, z_i, t) \cdot \mathbf{f}_i(\mathbf{x}_i, \tilde{\alpha}_i, \alpha^*_{-i})\\ &-\nabla_{z_i} W_i(\mathbf{x}_i, z_i, t) \cdot l_i\Big(x_s^{x_i,t,\tilde{\alpha}_i}, \tilde{\alpha}_i\Big(\mathbf{x}_s^{\mathbf{x}_i,t,\tilde{\alpha}_i,\alpha^*_{-i}}, s\Big)\Big) \leq -\xi. \end{aligned} \tag{40}$$

for all points $(\mathbf{x}_i, z_i, t)$ sufficiently close to $(\mathbf{x}_i(t_0), z_i(t_0), t_0)$: there exists small enough $h_1 > 0$ such that $||\mathbf{x}_i - \mathbf{x}_i(t_0)|| + |z_i - z_i(t_0)| + |t - t_0| < h_1$. According to assumptions in Section III-B, choose a small h such that $||\mathbf{x}_i - \mathbf{x}_i(t_0)|| + |z_i - z_i(t_0)| < h_1 - h$ for $s \in [t_0, t_0 + h]$, then

$$\begin{aligned} &c_i(\mathbf{x}_s^{\mathbf{x}_i,t,\tilde{\alpha}_i,\alpha^*_{-i}}) - W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) \leq -\xi\\ &\nabla_t W_i(\mathbf{x}_i, z_i, s) + \nabla_{\mathbf{x}_i} W_i(\mathbf{x}_i, z_i, s) \cdot \mathbf{f}_i(\mathbf{x}_i, \tilde{\alpha}_i, \alpha^*_{-i})\\ &-\nabla_{z_i} W_i(\mathbf{x}_i, z_i, s) \cdot l_i\Big(x_s^{x_i,t,\tilde{\alpha}_i}, \tilde{\alpha}_i\Big(\mathbf{x}_s^{\mathbf{x}_i,t,\tilde{\alpha}_i,\alpha^*_{-i}}, s\Big)\Big) \leq -\xi. \end{aligned} \tag{41}$$

According to the condition that $V_i - W_i$ has a local maximum at $(\mathbf{x}(t_0), z_i(t_0), t_0)$, then

$$\begin{aligned} &V_i(\mathbf{x}_i(t_0 + h), z_i(t_0 + h), t_0 + h)\\ &\quad - W_i(\mathbf{x}_i(t_0 + h), z_i(t_0 + h), t_0 + h)\\ &\leq V_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) - W_i(\mathbf{x}(t_0), z_i(t_0), t_0)\\ &\Rightarrow V_i(\mathbf{x}_i(t_0 + h), z_i(t_0 + h), t_0 + h)\\ &\quad - V_i(\mathbf{x}_i(t_0), z_i(t_0), t_0)\\ &\leq W_i(\mathbf{x}_i(t_0 + h), z_i(t_0 + h), t_0 + h)\\ &\quad - W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) \end{aligned}$$

$$\begin{aligned}&\Rightarrow V_i(\mathbf{x}_i(t_0+h), z_i(t_0+h), t_0+h)\\&\quad - V_i(\mathbf{x}_i(t_0), z_i(t_0), t_0)\\&\leq \int_{t_0}^{t_0+h} \frac{dW_i}{dt} ds\\&\Rightarrow V_i(\mathbf{x}_i(t_0+h), z_i(t_0+h), t_0+h)\\&\quad - V_i(\mathbf{x}_i(t_0), z_i(t_0), t_0)\\&\leq \int_{t_0}^{t_0+h} \{\nabla_t W_i(\mathbf{x}_i, z_i, s) + \nabla_{\mathbf{x}_i} W_i(\mathbf{x}_i, z_i, s)\cdot \mathbf{f}_i\\&\quad -\nabla_{z_i} W_i(\mathbf{x}_i, z_i, s)\cdot l_i\}\, ds \leq -\xi h. \qquad (42)\end{aligned}$$

Lemma 2 says that

$$\begin{aligned}V_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) = \min_{u_i\in\mathcal{U}_i} \max\Bigg\{&\max_{s\in[t_0,t_0+h]} c_i(\mathbf{x}_i(s))\\&V_i(\mathbf{x}_i(t_0+h), z_i(t_0+h), t_0+h)\Bigg\}. \qquad (43)\end{aligned}$$

Subtract (43) by $W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0)$ on both sides and combine (41) and (42)

$$\begin{aligned}0 &= (V_i - W_i)(\mathbf{x}_i(t_0), z_i(t_0), t_0)\\&= \min_{u_i\in\mathcal{U}_i} \max\{-\xi, -\xi h\} < 0 \qquad (44)\end{aligned}$$

which is a contradiction. Thus, we prove that

$$\begin{aligned}&\max\Big\{c_i(\mathbf{x}_{t_0}^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}) - W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0)\\&\nabla_t W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) - \mathcal{H}_i(t_0, \mathbf{x}_i(t_0), z_i(t_0)\\&\nabla_{\mathbf{x}_i} W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0), \nabla_{z_i} W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0))\Big\}\\&\geq 0. \qquad (45)\end{aligned}$$

3) Let $W_i \in C^\infty(\mathcal{X}\times\mathbb{R}\times[0,T])$, and assume that $V_i - W_i$ has local minimum at $(\mathbf{x}_i(t_0), z_i(t_0), t_0)) \in \mathcal{X}\times\mathbb{R}\times[0,T)$ and $(V_i - W_i)(\mathbf{x}_i(t_0), z_i(t_0), t_0)) = 0$, we need to prove

$$\begin{aligned}&\max\Big\{c_i(\mathbf{x}_{t_0}^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}) - W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0)\\&\nabla_t W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) - \mathcal{H}_i(t_0, \mathbf{x}_i(t_0), z_i(t_0)\\&\nabla_{\mathbf{x}_i} W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0), \nabla_{z_i} W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0))\Big\}\\&\leq 0. \qquad (46)\end{aligned}$$

The definition of V_i says that

$$\begin{aligned}V_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) =\ &\max\Bigg\{\max_{s\in[t_0,T]} c_i(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}})\\&\qquad g_i\left(x_T^{x_i,t,\alpha_i}\right) - z_i(T)\Bigg\}\\\geq\ &\max\Bigg\{c_i(\mathbf{x}_{t_0}^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}})\\&\qquad g_i\left(x_T^{x_i,t,\alpha_i}\right) - z_i(T)\Bigg\} \qquad (47)\end{aligned}$$

for all $\alpha_i \in \mathcal{A}(t_0)$. Subtract (47) by $W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0)$ on both sides to have

$$\begin{aligned}0 &= (V_i - W_i)(\mathbf{x}_i(t_0), z_i(t_0), t_0)\\&\geq \max\Big\{c_i\left(\mathbf{x}_{t_0}^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}\right)\\&\quad - W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0), g_i\left(x_T^{x_i,t,\alpha_i}\right) - z_i(T)\\&\quad - W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0)\Big\}. \qquad (48)\end{aligned}$$

Then, we must prove the following inequality:

$$\begin{aligned}&\nabla_t W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) - \mathcal{H}_i(t_0, \mathbf{x}_i(t_0), z_i(t_0)\\&\nabla_{\mathbf{x}_i} W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0), \nabla_{z_i} W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0)) \leq 0. \qquad (49)\end{aligned}$$

Suppose not. Then, there exists $\xi > 0$ such that

$$\begin{aligned}\nabla_t W_i(\mathbf{x}_i, z_i, t) - \max_{u_i\in\mathcal{U}_i}[&-\nabla_{\mathbf{x}_i} W_i(\mathbf{x}_i, z_i, t)\cdot \mathbf{f}_i\\&+ \nabla_{z_i} W_i(\mathbf{x}_i, z_i, t)\cdot l_i] \geq \xi \qquad (50)\end{aligned}$$

for all points $(\mathbf{x}_i, z_i, t)$ sufficiently close to $(\mathbf{x}_i(t_0), z_i(t_0), t_0)$: there exists small enough $h_1 > 0$ such that $||\mathbf{x}_i - \mathbf{x}_i(t_0)|| + |z_i - z_i(t_0)| + |t - t_0| < h_1$. For any $\alpha_i \in \mathcal{A}$, where

$$\begin{aligned}&\alpha_i \in \arg\max\nolimits_{\alpha_i\in\mathcal{A}} - \nabla_{\mathbf{x}_i} W_i(\mathbf{x}_i, z_i, s)\cdot \mathbf{f}_i(\mathbf{x}_i, \alpha_i, \alpha^*_{-i})\\&+ \nabla_{z_i} W_i(\mathbf{x}_i, z_i, s)\cdot l_i\left(x_s^{x_i,t,\alpha_i}, \alpha_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}, s\right)\right). \qquad (51)\end{aligned}$$

According to assumptions in Section III-B, choose a small h such that $||\mathbf{x}_i - \mathbf{x}_i(t_0)|| + |z_i - z_i(t_0)| < h_1 - h$ for $s \in [t_0, t_0+h]$, then

$$\begin{aligned}&\nabla_t W_i(\mathbf{x}_i, z_i, s) + \nabla_{\mathbf{x}_i} W_i(\mathbf{x}_i, z_i, s)\cdot \mathbf{f}_i(\mathbf{x}_i, \alpha_i, \alpha^*_{-i})\\&-\nabla_{z_i} W_i(\mathbf{x}_i, z_i, s)\cdot l_i\left(x_s^{x_i,t,\alpha_i}, \alpha_i\left(\mathbf{x}_s^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}, s\right)\right) \geq \xi \qquad (52)\end{aligned}$$

for all $s \in [t_0, t_0+h]$. We integrate (52) over $s \in [t_0, t_0+h]$ to get

$$\begin{aligned}&W_i(\mathbf{x}_i(t_0+h), z_i(t_0+h), t_0+h)\\&\quad - W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) \geq \xi h. \qquad (53)\end{aligned}$$

We have the following relation because (53) holds for all $u_i \in \mathcal{U}_i$:

$$\begin{aligned}&\min_{u_i\in\mathcal{U}_i} W_i(\mathbf{x}_i(t_0+h), z_i(t_0+h), t_0+h)\\&\quad - W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) \geq \xi h. \qquad (54)\end{aligned}$$

According to the condition that $V_i - W_i$ has a local minimum at $(\mathbf{x}_i(t_0), z_i(t_0), t_0)$, then

$$\begin{aligned}&\min_{u_i\in\mathcal{U}_i} V_i(\mathbf{x}_i(t_0+h), z_i(t_0+h), t_0+h)\\&- V_i(\mathbf{x}_i(t_0), z_i(t_0), t_0)\\&\geq \min_{u_i\in\mathcal{U}_i} W_i(\mathbf{x}_i(t_0+h), z_i(t_0+h), t_0+h)\end{aligned}$$

$$
\begin{aligned}
&- W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0)\\
&\geq \xi h\\
\Rightarrow\ &\min_{u_i \in \mathcal{U}_i} V_i(\mathbf{x}_i(t_0+h), z_i(t_0+h), t_0+h)\\
&> V_i(\mathbf{x}_i(t_0), z_i(t_0), t_0). \qquad (55)
\end{aligned}
$$

However, Lemma 2 says that

$$
\begin{aligned}
&\min_{u_i \in \mathcal{U}_i} V_i(\mathbf{x}_i(t_0+h), z_i(t_0+h), t_0+h)\\
&\leq V_i(\mathbf{x}(t_0), z_i(t_0), t_0) \qquad (56)
\end{aligned}
$$

which is a contradiction. Thus, we prove that

$$
\begin{aligned}
&\max\Big\{c_i(\mathbf{x}_{t_0}^{\mathbf{x}_i,t,\alpha_i,\alpha^*_{-i}}) - W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0),\\
&\nabla_t W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0) - \mathcal{H}_i(t_0, \mathbf{x}_i(t_0), z_i(t_0),\\
&\nabla_{\mathbf{x}_i} W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0), \nabla_{z_i} W_i(\mathbf{x}_i(t_0), z_i(t_0), t_0))\Big\}\\
&\leq 0. \qquad (57)
\end{aligned}
$$

Hence, we prove that $V_i(\mathbf{x}_i, z_i, t)$ is the viscosity solution. □

Acknowledgment

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation or the U.S. Government.

References

[1] F. Duarte and C. Ratti, "The impact of autonomous vehicles on cities: A review," *J. Urban Technol.*, vol. 25, no. 4, pp. 3–18, 2018.

[2] B. S. Peters, P. R. Armijo, C. Krause, S. A. Choudhury, and D. Oleynikov, "Review of emerging surgical robotic technology," *Surg. Endoscopy*, vol. 32, pp. 1636–1655, 2018.

[3] R. R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Trans. Syst., Man, Cybern., Part C. (Appl. Rev.)*, vol. 34, no. 2, pp. 138–153, May 2004.

[4] K. Leung et al., "On infusing reachability-based safety assurance within planning frameworks for human–robot vehicle interactions," *Int. J. Robot. Res.*, vol. 39, no. 10/11, pp. 1326–1345, 2020.

[5] N. Gammoudi and H. Zidani, "A differential game control problem with state constraints," *Math. Control Related Fields*, vol. 13, no. 2, pp. 554–582, 2023.

[6] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. Hoboken, NJ, USA: Wiley, 2012.

[7] R. Bellman and R. E. Kalaba, *Dynamic Programming and Modern Control Theory*, vol. 81. New York, NY, USA: Academic Press, 1965.

[8] E. Weinan, J. Han, and A. Jentzen, "Algorithms for solving high dimensional PDEs: From nonlinear Monte Carlo to machine learning," *Nonlinearity*, vol. 35, no. 1, 2021, Art. no. 278.

[9] Y. Shin, J. Darbon, and G. E. Karniadakis, "On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs," *Commun. Comput. Phys.*, vol. 28, pp. 2042–2074, 2020.

[10] K. Ito, C. Reisinger, and Y. Zhang, "A neural network-based policy iteration algorithm with global H^2-superlinear convergence for stochastic games on domains," *Found. Comput. Math.*, vol. 21, no. 2, pp. 331–374, 2021.

[11] O. Fuks and H. A. Tchelepi, "Limitations of physics informed machine learning for nonlinear two-phase transport in porous media," *J. Mach. Learn. Model. Comput.*, vol. 1, no. 1, pp. 19–37, 2020.

[12] O. L. Mangasarian, "Sufficient conditions for the optimal control of nonlinear systems," *SIAM J. Control*, vol. 4, no. 1, pp. 139–152, 1966.

[13] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.

[14] A. Altarovici, O. Bokanowski, and H. Zidani, "A general Hamilton-Jacobi framework for non-linear state-constrained control problems," *ESAIM: Control, Optim. Calculus Variations*, vol. 19, no. 2, pp. 337–357, 2013.

[15] D. Lee, "Safety-guaranteed autonomy under uncertainty," Ph.D. dissertation, Univ. of California, Berkeley, Berkeley, CA, USA, 2022.

[16] 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," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.

[17] M. Raissi, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis, "Deep learning of vortex-induced vibrations," *J. Fluid Mechan.*, vol. 861, pp. 119–137, 2019.

[18] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, "Adaptive activation functions accelerate convergence in deep and physics-informed neural networks," *J. Comput. Phys.*, vol. 404, 2020, Art. no. 109136.

[19] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.

[20] S. Bansal and C. J. Tomlin, "DeepReach: A deep learning approach to high-dimensional reachability," in *Proc. 2021 IEEE Int. Conf. Robot. Automat.*, 2021, pp. 1817–1824.

[21] L. Zhang, M. Ghimire, W. Zhang, Z. Xu, and Y. Ren, "Approximating discontinuous Nash equilibrial values of two-player general-sum differential games," in *Proc. IEEE 2023 Int. Conf. Robot. Automat.*, 2023, pp. 3022–3028.

[22] M. Bui, G. Giovanis, M. Chen, and A. Shriraman, "OptimizeDDP: An efficient, user-friendly library for optimal control and dynamic programming," 2022, *arXiv:2204.05520*.

[23] M. G. Crandall and P.-L. Lions, "Viscosity solutions of Hamilton-Jacobi equations," *Trans. Amer. Math. Soc.*, vol. 277, no. 1, pp. 1–42, 1983.

[24] S. Osher and C.-W. Shu, "High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations," *SIAM J. Numer. Anal.*, vol. 28, no. 4, pp. 907–922, 1991.

[25] S. Osher, R. Fedkiw, and K. Piechor, "Level set methods and dynamic implicit surfaces," *Appl. Mech. Rev.*, vol. 57, no. 3, pp. 1–271, 2004.

[26] I. M. Mitchell and J. A. Templeton, "A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems," in *Proc. Hybrid Syst.: Comput. Control: 8th Int. Workshop*, Zurich, Switzerland: Springer, Mar. 9–11, 2005, pp. 480–494.

[27] I. M. Mitchell and C. J. Tomlin, "Overapproximating reachable sets by Hamilton-Jacobi projections," *J. Sci. Comput.*, vol. 19, no. 1, pp. 323–346, 2003.

[28] J. Han and J. Long, "Convergence of the deep BSDE method for coupled FBSDEs," *Probability, Uncertainty, Quantitative Risk*, vol. 5, no. 1, pp. 1–33, 2020.

[29] J. Han, A. Jentzen, and E. Weinan, "Solving high-dimensional partial differential equations using deep learning," *Proc. Nat. Acad. Sci.*, vol. 115, no. 34, pp. 8505–8510, 2018.

[30] T. Nakamura-Zimmerer, Q. Gong, and W. Kang, "Adaptive deep learning for high-dimensional Hamilton-Jacobi-Bellman equations," *SIAM J. Sci. Comput.*, vol. 43, no. 2, pp. A1221–A1247, 2021.

[31] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, "Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games," in *Proc. IEEE 2020 Int. Conf. Robot. Automat.*, 2020, pp. 1475–1481.

[32] J. Foerster, R. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch, "Learning with opponent-learning awareness," *Auton. Agents Multi-Agent Syst.*, 2018.

[33] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for cars that coordinate with people: Leveraging effects on human actions for planning and active information gathering over human internal state," *Auton. Robots*, vol. 42, no. 7, pp. 1405–1426, Oct. 2018.

[34] M. Kwon, E. Biyik, A. Talati, K. Bhasin, D. P. Losey, and D. Sadigh, "When humans aren't optimal: Robots that collaborate with risk-aware humans," in *Proc. 2020 ACM/IEEE Int. Conf. Hum.-Robot Interact.*, 2020, pp. 43–52.

[35] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proc. Nat. Acad. Sci.*, vol. 116, no. 50, pp. 24972–24978, 2019.

[36] Z. Zahedi, A. Khayatian, M. M. Arefi, and S. Yin, "Seeking Nash equilibrium in non-cooperative differential games," *J. Vib. Control*, vol. 29, no. 19/20, pp. 4566–4576, 2023.

[37] J. Li, Z. Xiao, J. Fan, T. Chai, and F. L. Lewis, "Off-policy Q-learning: Solving Nash equilibrium of multi-player games with network-induced delay and unmeasured state," *Automatica*, vol. 136, 2022, Art. no. 110076.

[38] S. Nikolaidis, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in collaborative tasks: Models and experiments," *Int. J. Robot. Res.*, vol. 36, no. 5–7, pp. 618–634, 2017.

[39] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *Proc. 2018 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2111–2117.

[40] C. Peng and M. Tomizuka, "Bayesian persuasive driving," in *Proc. 2019 Amer. Control Conf.*, 2019, pp. 723–729.

[41] Y. Wang, Y. Ren, S. Elliott, and W. Zhang, "Enabling courteous vehicle interactions through game-based and dynamics-aware intent inference," *IEEE Trans. Intell. Veh.*, vol. 5, no. 2, pp. 217–228, Jun. 2020.

[42] D. Fridovich-Keil et al., "Confidence-aware motion prediction for real-time collision avoidance," *Int. J. Robot. Res.*, vol. 39, no. 2/3, pp. 250–265, 2020.

[43] H. Hu, Z. Zhang, K. Nakamura, A. Bajcsy, and J. F. Fisac, "Deception game: Closing the safety-learning loop in interactive robot autonomy," in *Proc. 7th Annu. Conf. Robot. Learn.*, 2023.

[44] R. J. Aumann, M. Maschler, and R. E. Stearns, *Repeated Games With Incomplete Information*. Cambridge, MA, USA: MIT Press, 1995.

[45] P. Cardaliaguet, "Differential games with asymmetric information," *SIAM J. Control Optim.*, vol. 46, no. 3, pp. 816–838, 2007.

[46] P. Cardaliaguet, "Numerical approximation and optimal strategies for differential games with lack of information on one side," in *Advances in Dynamic Games and Their Applications: Analytical and Numerical Developments*. Berlin, Germany: Springer, 2009, pp. 1–18.

[47] P. Cardaliaguet and C. Rainer, "Games with incomplete information in continuous time and for continuous types," *Dyn. Games Appl.*, vol. 2, no. 2, pp. 206–227, 2012.

[48] A. W. Starr and Y.-C. Ho, "Nonzero-sum differential games," *J. Optim. Theory Appl.*, vol. 3, no. 3, pp. 184–206, 1969.

[49] E. Cristiani and P. Martinon, "Initialization of the shooting method via the Hamilton-Jacobi-Bellman approach," *J. Optim. Theory Appl.*, vol. 146, no. 2, pp. 321–346, 2010.

[50] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-learning for offline reinforcement learning," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 1179–1191.

[51] P. Petersen and F. Voigtlaender, "Optimal approximation of piecewise smooth functions using deep ReLU neural networks," *Neural Netw.*, vol. 108, pp. 296–330, 2018.

[52] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 7462–7473.

[53] J. C. Harsanyi, "Games with incomplete information played by "Bayesian" players, I–III Part I. The basic model," *Manage. Sci.*, vol. 14, no. 3, pp. 159–182, 1967.

[54] Y. Chen, L. Zhang, T. Merry, S. Amatya, W. Zhang, and Y. Ren, "When shall I. be empathetic? The utility of empathetic parameter estimation in multi-agent interactions," in *Proc. IEEE 2021 Int. Conf. Robot. Automat.*, 2021, pp. 2761–2767.

[55] C. Huang, J. Fan, W. Li, X. Chen, and Q. Zhu, "ReachNN: Reachability analysis of neural-network controlled systems," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 5s, pp. 1–22, 2019.

[56] D. Manzanas Lopez, P. Musau, N. P. Hamilton, and T. T. Johnson, "Reachability analysis of a general class of neural ordinary differential equations," in *Proc. Int. Conf. Formal Model. Anal. Timed Syst.*, 2022, pp. 258–277.

[57] H. Hu, M. Fazlyab, M. Morari, and G. J. Pappas, "Reach-SDP: Reachability analysis of closed-loop systems with neural network controllers via semidefinite programming," in *Proc. IEEE 2020 59th Conf. Decis. Control*, 2020, pp. 5929–5934.

[58] M. Everett, G. Habibi, C. Sun, and J. P. How, "Reachability analysis of neural feedback loops," *IEEE Access*, vol. 9, pp. 163938–163953, 2021.

[59] T. Entesari and M. Fazlyab, "Automated reachability analysis of neural network-controlled systems via adaptive polytopes," in *Proc. Learn. Dyn. Control Conf.*, 2023, pp. 407–419.

[60] A. Lin and S. Bansal, "Generating formal safety assurances for high-dimensional reachability," in *Proc. IEEE 2023 Int. Conf. Robot. Automat.*, 2023, pp. 10525–10531.

[61] K.-C. Hsu, H. Hu, and J. F. Fisac, "The safety filter: A unified view of safety-critical control in autonomous systems," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 7, pp. 47–72, 2023.

[62] A. Bressan, "Noncooperative differential games," *Milan J. Math.*, vol. 79, pp. 357–427, 2011.

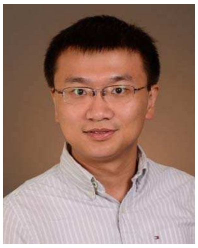
[63] L. C. Evans, *Partial Differential Equations*, 2nd ed., vol. 19. Providence, RI, USA: Amer. Math. Soc., 2022.

Lei Zhang (Student Member, IEEE) received the B.Eng. degree in process equipment and control engineering and the M.Sc. degree in material engineering from Xi'an Jiaotong University, Xi'an, China, in 2010 and 2012, respectively. He is currently working toward the Ph.D. degree in mechanical engineering with Design Informatics Lab, Arizona State University, Tempe, AZ, USA.

He is a Research Assistant with Design Informatics Lab, Arizona State University. His research interests include machine learning, optimization, and game theory, which are applied to human–robot interactions.

Mukesh Ghimire (Student Member, IEEE) received the B.S. degree in mechanical engineering with minors in computer science and mathematics from the University of Mississippi, Oxford, MS, USA, in 2021. He is currently working toward the Ph.D. degree in mechanical engineering with Design Informatics Lab, Arizona State University, Tempe, AZ, USA.

He is a Research Assistant with Design Informatics Lab, Arizona State University, Tempe, AZ, USA. His research interests include game theory, artificial intelligence, and reinforcement learning.



Wenlong Zhang (Member, IEEE) received the B.Eng. degree (hons.) in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2010, and the M.S. degree in mechanical engineering, the M.A. degree in statistics, and the Ph.D. degree in mechanical engineering from the University of California, Berkeley, CA, USA, in 2012, 2013, and 2015, respectively.

He is currently an Associate Professor with the School of Manufacturing Systems and Networks, Arizona State University, Tempe, AZ, USA, where he directs the ASU Robotics and Intelligent Systems Lab. His research interests include dynamic systems and control, interactive robotics, and human–machine collaboration.

Zhe Xu (Member, IEEE) received the B.S. and M.S. degrees from Tianjin University, Tianjin, China, in 2011 and 2014, respectively, and the Ph.D. degree from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2018, all in electrical engineering.

He is currently an Assistant Professor with the School for Engineering of Matter, Transport, and Energy, Arizona State University (ASU), Tempe, AZ, USA. Before joining ASU, he was a Postdoctoral Researcher with the Oden Institute for Computational Engineering and Sciences, University of Texas at Austin, Austin, TX, USA. His research interests include formal methods, autonomous systems, control systems, and reinforcement learning.

Yi Ren (Member, IEEE) received the B.Eng. degree in automotive engineering from Tsinghua University, Beijing, China, in 2007, and the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2012.

He is currently an Associate Professor in mechanical engineering with Arizona State University, Tempe, AZ, USA. From 2012 to 2014, he was a Postdoctoral Researcher with the University of Michigan. His research focuses on safety in AI-enabled engineering systems.

Dr. Ren was the recipient of the Best Paper Award at the 2015 ASME International Design and Engineering Technical Conferences (IDETC).