2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Approximating Discontinuous Nash Equilibrial Values of Two-Player General-Sum Differential Games

Lei Zhang[1], Mukesh Ghimire[1], Wenlong Zhang[2], Zhe Xu[1], Yi Ren[1]

Abstract— **Finding Nash equilibrial policies for two-player differential games requires solving Hamilton-Jacobi-Isaacs (HJI) PDEs. Self-supervised learning has been used to approximate solutions of such PDEs while circumventing the curse of dimensionality. However, this method fails to learn discontinuous PDE solutions due to its sampling nature, leading to poor safety performance of the resulting controllers in robotics applications when player rewards are discontinuous. This paper investigates two potential solutions to this problem: a hybrid method that leverages both supervised Nash equilibria and the HJI PDE, and a value-hardening method where a sequence of HJIs are solved with a gradually hardening reward. We compare these solutions using the resulting generalization and safety performance in two vehicle interaction simulation studies with 5D and 9D state spaces, respectively. Results show that with informative supervision (e.g., collision and near-collision demonstrations) and the low cost of self-supervised learning, the hybrid method achieves better safety performance than the supervised, self-supervised, and value hardening approaches on equal computational budget. Value hardening fails to generalize in the higher-dimensional case without informative supervision. Lastly, we show that the neural activation function needs to be continuously differentiable for learning PDEs and its choice can be case dependent.**

I. Introduction

Problem statement. Human-robot interactions (HRIs) in real time can be modeled as general-sum differential games. The Nash equilibrial value of the game is a viscosity solution to the Hamilton-Jacobi-Isaacs (HJI) equations [1], and is a function of the players' states and time. Conventional algorithms for solving Hamilton-Jacobi PDEs are known to suffer from the curse of dimensionality [2], i.e., values become hard to compute for high-dimensional state spaces. Recent attempts consider self-supervised (i.e., physics-informed) learning that forces the consistency between the value and the PDE, and have achieved empirical success on a variety of differential games [3], [4]. While convergence of this method has been proven for Lipschitz and Hölder continuous player rewards [4], [5], our experiments suggest that convergence to the true values cannot be achieved when players' rewards, and thus the values, are discontinuous with respect to states and time. Such discontinuity can occur when players receive both continuous rewards, e.g., energy consumption or path-following losses, and those encoded from temporal logic, e.g., safety penalties. We discuss the challenge of learning discontinuous values using a toy case in Sec. III.

Solutions. We examine two potential solutions to this challenge: (1) A hybrid method where we leverage both supervised equilibrial data generated by Pontryagin's Maximum Principle (PMP) [6] and the HJI equations; (2) A modified self-supervised method where we gradually harden an initially softened reward. Comparisons are performed on two vehicle interaction studies: An uncontrolled intersection case with a 5D value function with complete information where players know each other's types and incomplete information where player types are private and inferred, and a collision avoidance case with a 9D value function.

Contributions. Our study leads to the following new findings: (1) We show that the hybrid method achieves better generalization performance on value prediction and safer control policies than other methods under the same computational budget. On the other hand, value-hardening fails to generalize in the higher-dimensional case, indicating the importance of informative supervision. To the authors' best knowledge, this is the first paper that addresses the challenge in approximating values of HJI PDEs with discontinuous reward; (2) We show that the choice of neural activation in the value network is important: `relu` does not generalize well due to its discontinuous derivative. Performance of `sin` activation varies across case studies, indicating the need for hyperparameter tuning. `tanh` and continuously differentiable variants of `relu`, e.g., `gelu`, generalize well using the hybrid method. These new findings add counter examples to the existing literature that promotes the use of `sin` activation for value approximation [3], and support recent discussions on the need for adaptive activations for neural network-based PDE solvers [7].

II. Related Work

Solving Hamilton-Jacobi PDEs via deep learning. With the development of auto-differentiation [8], deep neural networks have become means to solving PDEs when analytical methods suffer from the curse of dimensionality [9]. Existing methods of this kind can be grouped into three types: "Boundary matching" methods first reformulate PDEs as backward stochastic differential equations and solve them by forcing a match between the terminal states of the resultant stochastic processes and the boundary conditions [10], [11]. "Equation matching" methods, which are often called physics-informed neural nets (PINN), directly force a match

This work was supported by the National Science Foundation under Grant CMMI-1925403.

[1]L. Zhang, M. Ghimire, Z. Xu, and Y. Ren are with Department of Mechanical and Aerospace Engineering, Arizona State University, Tempe, AZ, 85287, USA. Email: {lzhan300, mghimire, xzhe1, yiren}@asu.edu

[3]W. Zhang is with The Polytechnic School, Ira A. Fulton Schools of Engineering, Arizona State University, Mesa, AZ, 85212, USA. Email: wenlong.zhang@asu.edu

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10160219

between a surrogate function and the PDE [3], [12]. Both boundary and equation matching are self-supervised, and have proven convergence to the ground truth solutions when both the network and the governing equations are continuous [4], [11], [12]. Lastly, supervised learning methods train a network based on sampled solutions to the PDE. To the authors' best knowledge, there is currently few generalization analysis for this method. Recent studies have investigated the effectiveness of "equation matching" at solving PDEs with discontinuous solutions (e.g., Burgers equation [7]) where both initial and terminal boundaries are specified. We show in this paper that PDEs with only terminal (or initial) boundary conditions, such as HJIs, cause an unidentifiability issue.

Games with incomplete information. Zero-sum dynamic games with incomplete- or imperfect-information can be solved by counterfactual regret minimization (CFR) [13] or Regularized Nash Dynamics (RND) [14], which have been successfully applied to heads-up Texas hold'em poker [15], [16] and Stratego [14]. However, there have so far been few applications of CFR and RND to continuous-time interactions with incomplete information, e.g., through time discretization. On the other hand, attempts to directly solve HRIs are often facilitated by simplifications of the underlying games that balance theoretical soundness and practical utility. Most attempts of this type simplify games as optimal control problems or complete-information games (e.g., [17]–[22]), and some use belief updates to adapt motion planning (e.g., [23]–[28]). For real-time control, [29] proposed to solve linear-quadratic games iteratively to approximate the equilibrial policy of a complete-information differential game. Unlike existing studies, in this paper we achieve real-time belief update and feedback control in incomplete-information games by learning off-line the Nash equilibrial values of hypothetical games to be played by all possible combinations of player types. The values then enable online Bayesian belief update to correct players' (false) beliefs about each other's type.

III. Methods

Notations. Let $\mathcal{X}_i$ (resp. $\mathcal{U}_i$) be the state (resp. action) space for Player i. The time-independent state dynamics of Player i is denoted by $\dot{x}_i[t] = f(x_i[t], u_i[t])$ for $x_i[t] \in \mathcal{X}_i$ and $u_i[t] \in \mathcal{U}_i$. Dependence of time will be omitted when possible. For a fixed-time interaction within $[0, T]$, the instantaneous and terminal losses of Player i are denoted by $l((x_i, x_{-i}), u_i)$ and $c(x_i, x_{-i})$, respectively. For a complete-information differential game between two players, the value function of Player i is $\nu_i(\cdot,\cdot,\cdot) : \mathcal{X}_i \times \mathcal{X}_{-i} \times [0, T] \to \mathbb{R}$ where the arguments are in the order of the ego's (Player i's) states, the other player's states, and the current time. To reduce notational burden, we will use f_i, l_i, c_i, and ν_i respectively for the player-wise dynamics, losses, and the value, and use $\mathbf{a}_i = (a_i, a_{-i})$ to concatenate variables from the ego (i) and the other players. We use $\nabla_x \cdot$ to denote partial derivative with respective to x.

Preliminaries. *Hamilton-Jacobi-Isaacs equations*: The Nash equilibrial values of a two-player general-sum differential game solve the following HJI equations (L) and satisfy the boundary conditions (D) [30]:

$$\begin{aligned} &L(\nu_i, \nabla_{\mathbf{x}_i}\nu_i, \mathbf{x}_i, t) := \nabla_t \nu_i + \nabla_{\mathbf{x}_i}\nu_i^T \mathbf{f}_i - l_i = 0 \\ &D(\nu_i, \mathbf{x}_i) := \nu_i(\mathbf{x}_i, T) - c_i = 0, \quad \text{for} \quad i = 1, 2. \end{aligned} \tag{1}$$

Here players' policies are derived by maximizing the equilibrial Hamiltonian $h_i(\mathbf{x}_i, \nabla_{x_i}\nu_i, t) = \nabla_{x_i}\nu_i^T f_i - l_i$: $u_i = \arg\max_{u \in \mathcal{U}_i} \{h_i\}$ for $i = 1, 2$.

Pontryagin's Maximum Principle: While solving the HJI would provide a feedback control policy, it is often more tractable to compute open-loop policies for specific initial state $(\bar{x}_1, \bar{x}_2) \in \mathcal{X}_1 \times \mathcal{X}_2$ by solving the following boundary value problem (BVP) according to PMP[1]:

$$\begin{aligned} &\dot{x}_i = f_i, \quad x_i[0] = \bar{x}_i, \\ &\dot{\lambda}_i = -\nabla_{x_i} h_i, \quad \lambda_i[T] = -\nabla_{x_i} c_i, \\ &u_i = \arg\max_{u \in \mathcal{U}_i} \{h_i\} \quad \text{for} \quad i = 1, 2. \end{aligned} \tag{2}$$

Here λ_i is the time-dependent co-state for Player i. The co-state connects PMP and HJI through $\lambda_i = \nabla_{x_i}\nu_i$. Solutions to Eq. (2) are specific to the given initial states.

Self-supervised learning for differential games. This approach directly trains a neural network that approximately satisfies the governing equations. Let $\hat{\nu}_i(\cdot,\cdot,\cdot) : \mathcal{X}_i \times \mathcal{X}_{-i} \times [0, T] \to \mathbb{R}$ be a neural network that approximates ν_i, and let $\mathcal{D} = \left\{\left(x_1^{(k)}, x_2^{(k)}, t^{(k)}\right)\right\}_{k=1}^{K}$ be uniform samples in $\mathcal{X}_1 \times \mathcal{X}_1 \times [0, T]$. We extend the existing formulation for solving zero-sum games [3] to the general-sum setting:

$$\begin{aligned} \min_{\hat{\nu}_1, \hat{\nu}_2} \quad L_1(\hat{\nu}_1, \hat{\nu}_2; \theta) := &\sum_{k=1}^{K}\sum_{i=1}^{2} \left\| L(\hat{\nu}_i, \nabla_{\mathbf{x}_i}\hat{\nu}_i, \mathbf{x}_i^{(k)}, t^{(k)}) \right\| \\ &+ C_1 \phi\left(D(\hat{\nu}_i, \mathbf{x}_i^{(k)})\right), \end{aligned} \tag{3}$$

where $\hat{\nu}_i^{(k)}$ is a shorthand for $\hat{\nu}_i\left(x_i^{(k)}, x_{-i}^{(k)}, t^{(k)}\right)$ and C_1 balances the inconsistencies of the value network to the HJI (L) and to the boundary conditions (D). It is worth noting that in each iteration of solving Eq. (3), a sub-routine is needed to find the control policies that maximize the Hamiltonian. [12] proved the convergence of $\hat{\nu}_i$ to ν_i via solving Eq. (3) when $\phi(\cdot)$ is a $H^{3/2}$-norm; [4] provided convergence and generalization analyses of the learning in forms of Eq. (3) for linear second-order elliptic and parabolic type PDEs (which include HJI) with continuity assumptions on L, D, and $\hat{\nu}_i$.

Challenge in approximating discontinuous HJI values. We use a toy case to explain the challenge in approximating discontinuous solutions of a differential equation with only terminal boundary conditions using self-supervised learning: Consider a 1D function $\nu(\cdot)$ as a solution to $\nabla_x \nu - \delta(x) = 0$ with the boundary condition $\nu(1) = 0$ within $x \in [-1, 1]$, where $\delta(x)$ is a delta function that peaks at $x = 0$ (see Fig. 1). With uniform samples, the self-supervised loss (L_1) can be minimized almost surely using a horizontal line

[1]We note that solving BVP has its own numerical challenges when the equilibrium involves singular arcs [31]. These challenges are not explored in this paper.

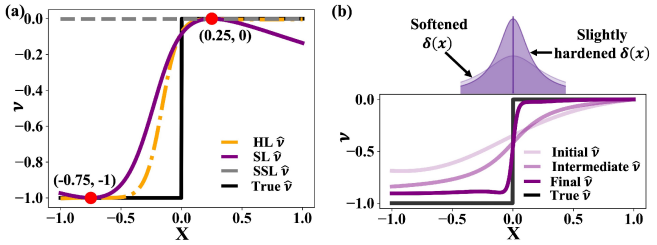

Fig. 1: (a) Value comparison among the learning methods for a simple 1D case. Red dots are the supervised data. (b) Evolution of the value function due to gradually hardening delta function. Delta functions are shown on top. Transparency reduces with hardening.

$\hat{\nu}(x) = 0$. This is because due to the differential nature of the governing equation, the accuracy of a self-supervised model at one point in space and time solely relies on that of its neighboring samples, yet informative neighbors (here at $x = 0$) have low (here zero) probability to be sampled. We also note that when both initial and terminal boundaries are specified (here $\nu(-1) = -1$ and $\nu(1) = 0$), self-supervised learning can work, e.g., in the case of a Burgers equation.

Solutions. (1) *Hybrid method*: Supervised data reveals the structure of the solution. In the toy case, an approximation to the solution can be learned based on two informative data points sampled from each side of 0 (see the SL curve in Fig. 1). When solving HJIs, the drawbacks of supervised learning are (1) its high data acquisition cost due to the need of solving BVPs and (2) its lack of generalization proof for solving PDEs. We hypothesize that these drawbacks can be addressed by combining supervised and self-supervised learning, since the latter only requires evaluating the HJI equations and is generalizable under continuity assumptions. This leads to the following hybrid method: Let $\mathcal{D}_s = \left\{\left(\mathbf{x}_i^{(k)}, t^{(k)}, \nu_i^{(k)}, \nabla_{\mathbf{x}_i}\nu_i^{(k)}\right) \text{ for } i = 1, 2\right\}_{k=1}^{K}$ be a dataset derived from solving Eq. (2) with initial states sampled in $\mathcal{X}_1 \times \mathcal{X}_2$. The supervised loss is defined as:

$$L_2\left(\hat{\nu}_1, \hat{\nu}_2; \mathcal{D}_\theta\right) := \sum_{k=1}^{K}\sum_{i=1}^{2}\left|\hat{\nu}_i^{(k)} - \nu_i^{(k)}\right| + C_2\left\|\nabla_{\mathbf{x}_i}\hat{\nu}_i^{(k)} - \nabla_{\mathbf{x}_i}\nu_i^{(k)}\right\|, \tag{4}$$

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 first introduce a surrogate differential equation, the continuous solution of which approximates the ground truth. Then we approximate the true solution by gradually "hardening" this surrogate. In the toy case, convergence of the solution can be achieved by gradually hardening a softened delta function (Fig. 1). Nonetheless, a potential issue with this approach is that as we harden the delta function, the set of informative samples (that specifies non-zero $\nabla_x \nu$ in the toy case) reduces. Importance sampling becomes necessary to avoid convergence towards $\hat{\nu} = 0$. In practice, it might improve the performance of value hardening by dynamically sampling states space, which leads to high PDE residuals [32]

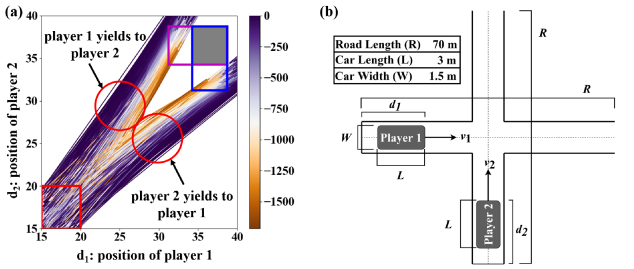

Fig. 2: (a) State trajectories of players projected to (d_1, d_2). Solid gray box: collision area from the perspective of aggressive players; hollow boxes (magenta for Player 1 and blue for Player 2): collision areas from the perspectives of non-aggressive players. Red box: sampling domain for initial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

IV. Case Studies

We compare generalization performance from self-supervised, supervised, hybrid, and value-hardening methods on two vehicle interaction simulation studies. The first considers an interaction between two players at an uncontrolled intersection, which leads to HJIs with coupled value functions defined on a 5D state space. With proper coordinate transformation [33], this case represents a broad range of challenging two-vehicle interactions including roundabouts and unprotected left turns. For completeness, we study both a complete- and an incomplete-information setting. We then consider a collision-avoidance case with higher state dimensionality similar to [3]. The difference is that here we encounter value discontinuity when players are required to both minimize effort and avoid collision.

A. Case 1: Uncontrolled intersection

Experiment setup. The schematics of the interaction is shown in Fig. 2, where Player i's states are composed of two scalars representing its location (d_i) and speed (v_i): $x_i := (d_i, v_i)$. The shared dynamics follows $\dot{d}_i = v_i$ and $\dot{v}_i = u_i$, where $u_i \in [-5, 10] m/s^2$ is a scalar control input that represents the acceleration. The instantaneous loss considers the control effort and a collision penalty:

$$l_i(\mathbf{x}_i, u_i; \theta_i) = u_i^2 + b\sigma(d_i, \theta_i)\sigma(d_{-i}, 1), \tag{5}$$

where $\sigma(d, \theta) = 1$ iff $d \in [R/2 - \theta W/2, (R + W)/2 + L]$ or otherwise $\sigma(d, \theta) = 0$; $b = 10^4$ sets a high loss for collision; R, L, and W are the road length, car length, and car width, respectively (see Fig. 2). $\Theta = \{1, 5\}$ represents the aggressive (`a`) and non-aggressive (`na`) types of players, respectively. Note that the collision penalty is the source of discontinuity (or numerically, large Lipschitz constant, which contributes to the rapid value change in Fig. 2). The terminal loss is defined to incentivize players to move across the intersection and restore nominal speed:

$$c_i(x) = -\alpha d_i(T) + (v_i(T) - \bar{v})^2, \tag{6}$$

where $\alpha = 10^{-6}$, $\bar{v} = 18 m/s$, and $T = 3s$.

Data. For a fair comparison, data sizes for each learning method are chosen so that the total wall-clock computational costs for data acquisition and learning combined are similar across different methods. Note that we do not use computational complexity to measure the computational budget due

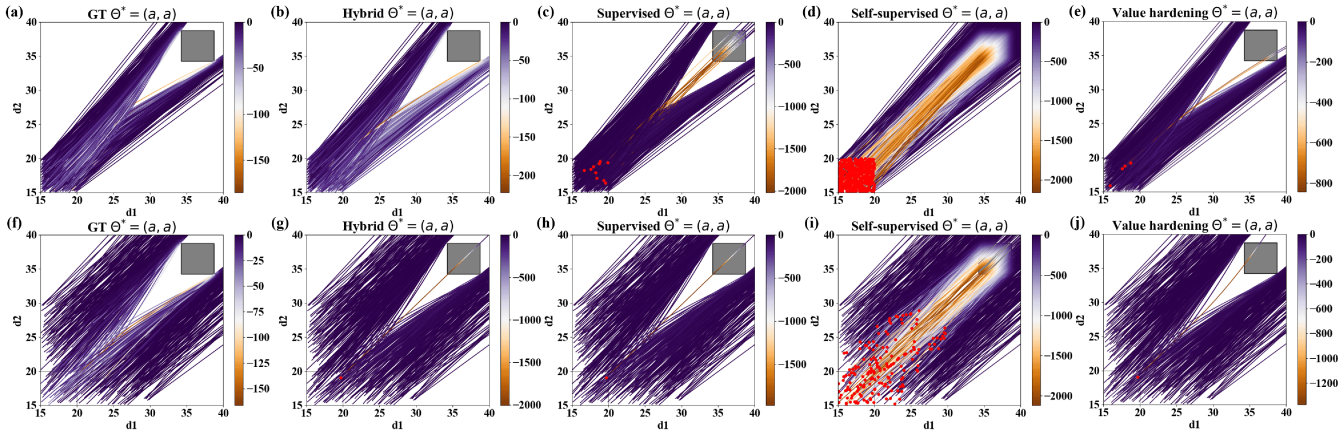

Fig. 3: (a), (f): Ground truth trajectories (projected to d_1-d_2) for $\mathcal{X}_{GT}$ and $\mathcal{X}_{XP}$, respectively. (b-e), (g-j): Trajectories generated using hybrid, supervised, self-supervised, and value-hardening methods under $\mathcal{X}_{GT}$ and $\mathcal{X}_{XP}$, respectively. Color: 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.

to the different nature of computations involved in acquiring supervised data (which requires iterative BVP solving) and self-supervised data (which only requires random sampling). Specifically, all training sessions are around 300 minutes on one GeForce GTX 1080 Ti GPU with 11 GB memory[2]. Detailed data acquisition methods and data sizes are as follows: For supervised learning, we generate 1.7k ground truth trajectories from initial states uniformly sampled in $\mathcal{X}_{GT} := [15, 20]m \times [18, 25]m/s$ and by solving Eq. (2). Each trajectory contains 31 $\times$ 2 data-points (sampled with a time interval of $0.1s$ and for two players), which leads to a total of 105.4k data points. For self-supervised learning and its value-hardening variant, we sample 122k states uniformly in $\mathcal{X}_{HJ} := [15, 105]m \times [15, 32]m/s$. For hybrid learning, we sample 1k ground truth trajectories (62k data points) uniformly in $\mathcal{X}_{GT}$ and sample 60k states uniformly in $\mathcal{X}_{HJ}$. We repeat the same sampling procedure for all four player type configurations (`a`, `a`), (`na`, `a`), (`a`, `na`), and (`na`, `na`).

The choices of the state spaces to sample from are based on the following rationale: The ground truth trajectories are derived by sampling initial states from identical domains of the two players. This is because collision and near-collision cases, which are informative for the learning, often happen when players start with similar distances and speeds. Similarly, the location range for supervised data ($[15, 20]m$) is calibrated so as to increase the chance of sampling informative trajectories within the specified time window. The speed range ($[18, 25]m/s$) is set based on nominal vehicle speed limits. For self-supervised learning, we use a space ($\mathcal{X}_{HJ}$) that approximately covers all states players can reach in $[0, T]$. Adaptive sampling is yet to be studied.

Network Architecture. Due to space limit, we only report results using fully-connected networks with 3 hidden layers, each containing 64 neurons, and with `tanh`, `relu`, or `sin` activations. Deeper or wider networks do not change our conclusions. `gelu` achieves similar performance as `tanh`. We normalize input data to $[-1, 1]$ to improve training convergence. Case 2 uses the same architecture.

[2]An exception is self-supervised learning, which is allowed to take around 500 minutes due to its poor generalization performance.

Training. All training problems are solved using the Adam optimizer with a fixed learning rate of 2×10^{-5}. For supervised learning, we train networks for a fixed 100k iterations. For self-supervised learning, we follow the curriculum learning method in [3] by first training the networks for 10k iterations using 1k uniformly sampled boundary states (at terminal time). The networks are then refined for 260k gradient descent steps, with states sampled from an expanding time window starting from the terminal. The value-hardening variant follows the same learning procedure but softens the collision penalty using sigmoid functions and gradually increases the shape parameter of the sigmoid to harden the penalty. For fair comparison, we use 5.4k training iterations for each hardening step for a total of 50 steps. For the hybrid method, we pre-train the networks for 100k 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.

Results for complete-information games. We generate a separate set of 600 ground truth trajectories with initial states in $\mathcal{X}_{GT}$ for each of the four player type configurations. For generalization performance, we measure the MAEs of value and control input predictions (denoted by $|\nu - \hat{\nu}|$ and $|u - \hat{u}|$ respectively) across the test trajectories. For safety performance, we use the learned value networks to compute equilibrial Hamiltonian, and maximize the Hamiltonian to derive players' closed-loop control inputs. We report the percentage of *avoidable* collisions. Initial states follow the test data. Performance results are summarized in Table I and sample trajectories for the (`a`, `a`) case shown in Fig. 3. Performance of (`a`, `na`) and (`na`, `a`) are averaged due to the symmetry of players in these settings.

To examine the out-of-distribution performance of the models, we repeat the experiments in an expanded state space $\mathcal{X}_{XP} = [15, 30]m \times [18, 25]m/s$ and use 500 uniformly sampled initial states for tests. Results are summarized in the same table and figure. For both tests, the results show that the hybrid method achieves the best generalization and safety performance. Poor generalization of the self-

TABLE I: Generalization and safety performance on complete-information games. SL, SSL, HL, VH are for supervised, self-supervised, hybrid, value hardening methods, respectively.

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

supervised model is notable.

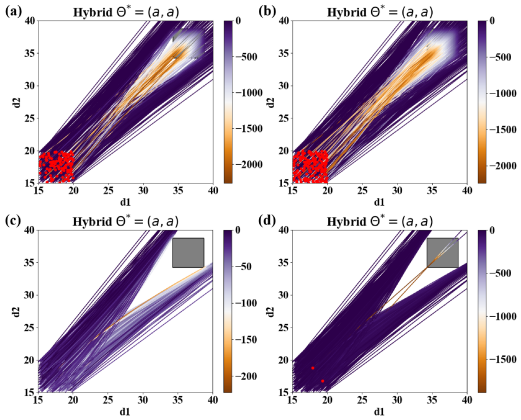

Fig. 4: Trajectories generated using neural networks with (a) `relu` and (b) `sin` activation functions and using L^1 for boundary norm (for `tanh`, refer to Fig. 3); trajectories generated using (c) L^1- and (d) L^2-norms for the boundary values and using `tanh` for activation. All trajectories are based on hybrid learning.

TABLE II: Safety performance w/ different activation functions (w/ L^1) and boundary norms (w/ `tanh`)

Method	Activation			Boundary Norm	
	`tanh`	`relu`	`sin`	L^1	L^2
Supervised	1.67%	2.50%	19.5%	-	-
Self-supervised	84.8%	84.0%	84.7%	-	-
Hybrid	**0.0%**	19.8%	28.7%	**0.0%**	0.4%

Ablation studies. We perform ablation studies to understand the influences of activation functions and the norm of boundary loss on model performance. Safety results are summarized in Table II for player types (a, a) and using the hybrid learning method, with training and test in $\mathcal{X}_{GT}$. Corresponding trajectories are visualized in Fig. 4. Results show that (1) the choice of the activation function has a significant influence on the resultant models, with `tanh` outperforming `relu` and `sin`, and that (2) the choice of the boundary norm does not have significant influence. **Remarks**: (1) `relu` networks have proven convergence to piecewise smooth functions in a supervised setting [34]. However, convergence in the self-supervised setting requires continuity of the network and its gradient [4], which `relu` does not offer. This result is consistent with [7] where `relu` underperforms in solving PDEs. We conjecture that this contributes to the deteriorated performance of `relu` under the hybrid setting. We note, however, that smooth variants of `relu` such as `gelu` can achieve performance comparable to that of `tanh`. (2) `sin` does not perform well for Case 1, yet achieves comparable performance to `tanh` in Case 2. This result suggests that fine-tuning of the frequency parameter of `sin` is necessary and case-dependent.

Results for incomplete-information games. We examine the advantage of the hybrid method when adopting value networks as closed-loop controllers for interactions in an incomplete-information setting. To constrain the scope of tests, we explore two settings of initial beliefs: Everyone believes that everyone is (1) most likely non-aggressive ($p_i := \Pr(\theta_i = a) = 0.2$), or (2) most likely aggressive ($p_i = 0.8$ for $i = 1, 2$). The common belief assumption indicates that Player i knows about Player j's belief about i. Also note that players' initial beliefs can mismatch with their true types. We model players to continuously perform Bayes updates on their beliefs based on observations, and determine the next control inputs based on the value networks that correspond to the most likely player types according to their current beliefs (see [28] for details on belief update and control). This model allows us to simulate state and belief dynamics in turn, although the values are approximated based on complete-information games and are thus decoupled from the belief dynamics. Table III compares safety performance of incomplete-information interactions using the same initial states as previous tests and values learned by the hybrid and the supervised methods. The former outperforms the latter across all belief settings.

TABLE III: Safety performance in uncontrolled intersections with incomplete information

True types	Initial common belief (p_1, p_2)	Hybrid	Supervised
(a,a)	(0.8,0.8)	**0.00%**	**0.00%**
(a,a)	(0.2,0.2)	**2.00%**	8.00%
(na,na)	(0.8,0.8)	**2.00%**	2.67%
(na,na)	(0.2,0.2)	**0.67%**	6.67%

B. Case 2: Collision avoidance

Experiment setup. The schematics is shown in Fig. 7, where Player i's states are composed of its location (p_i^x, p_i^y), orientation (ψ_i) and speed (v_i): $x_i := [p_i^x, p_i^y, \psi_i, v_i]^T$. We consider a unicycle model as the dynamics:

$$\begin{bmatrix} \dot{p}_i^x \\ \dot{p}_i^y \\ \dot{\psi}_i \\ \dot{v}_i \end{bmatrix} = \begin{bmatrix} v_i \cos(\psi_i) \\ v_i \sin(\psi_i) \\ \omega_i \\ u_i \end{bmatrix} \tag{7}$$

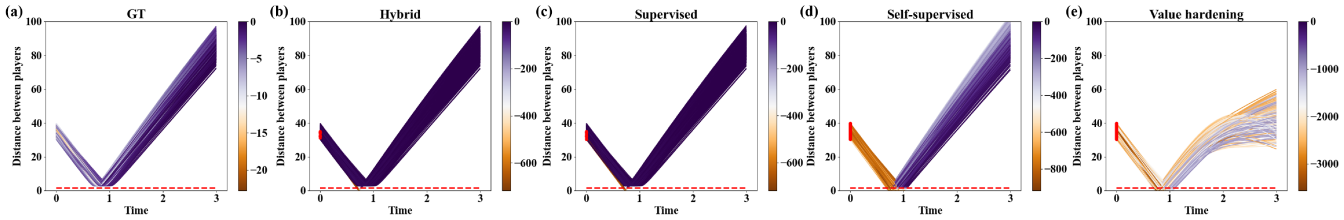

Fig. 5: (a): Ground truth distance between players over time for $\mathcal{X}_{GT}$. (b-e): Distance between players over time using hybrid, supervised, self-supervised, and value hardening models under $\mathcal{X}_{GT}$, respectively. Red dashed line represents the threshold distance for collision.

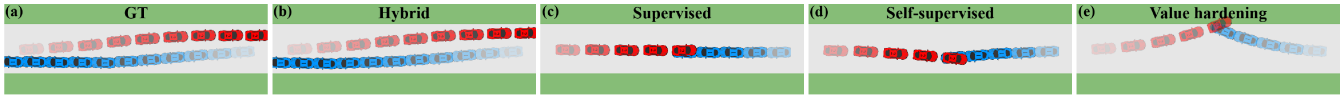

Fig. 6: Collision avoidance visualization: (a): Ground truth safe trajectory. Transparency reduces along time. (b-e): Trajectories generated using hybrid, supervised, self-supervised, and value hardening models, respectively.

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 considers the control effort and the collision penalty:

$$l_i(\mathbf{x}_i, \omega_i, u_i) = k\omega_i^2 + u_i^2 + b\sigma(\mathbf{x}_i, \eta), \tag{8}$$

where the penalty is $\sigma(\mathbf{x}_i, \eta) = \left(1 + \exp(\gamma(\sqrt{((R - p_2^x) - p_1^x)^2 + (p_2^y - p_1^y)^2} - \eta)\right)^{-1}$. $b = 10^4$ sets a high loss for collision; $\gamma = 5$ is a shape parameter; R is road length; $k = 100$ balances the control effort on turning and acceleration; $\eta = 1.5m$ is a threshold for collision. The terminal loss encourages players to move along the lane and restore nominal speed:

$$c_i(x) = -\alpha p_i^x(T) + (v_i(T) - \bar{v})^2 + (p_i^y(T) - \bar{p}^y)^2, \tag{9}$$

where $\alpha = 10^{-6}$, $\bar{v} = 18m/s$, $\bar{p}^y = 3m$, and $T = 3s$.

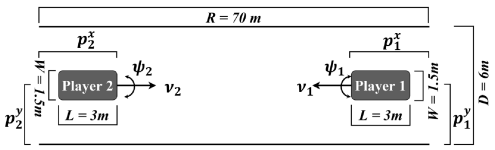

Fig. 7: Collision avoidance setup with two players.

Data. For supervised learning, we generate 1.45k ground truth trajectories from initial states uniformly sampled in $\mathcal{X}_{GT} := [15, 20]m \times [2.25, 3.75]m \times [-\pi/180, \pi/180] rad \times [18, 25] m/s$, which leads to a total of 89.9k data points. For self-supervised learning and its value-hardening variant, we sample 122k states uniformly in $\mathcal{X}_{HJ} := [15, 90]m \times [0, 6]m \times [-0.15, 0.18] rad \times [18, 25] m/s$. For hybrid learning, we generate 1k ground truth trajectories (62k data points) in $[15, 20]m \times [2.25, 3.75]m \times [-\pi/180, \pi/180] rad \times [18, 25] m/s$ and sample 60k states uniformly in $[15, 90]m \times [0, 6]m \times [-0.15, 0.18] rad \times [18, 25] m/s$.

Training. Self-supervised learning adopts pretraining of networks for 10k iterations using 1k uniformly sampled boundary states and trains the network for 350k iterations. The value-hardening variant uses 7.2k training iterations for each hardening step with a total of 50 steps for fair comparison. The rest of the settings follows Case 1.

Results. We use a separate test set of 600 ground truth collision-free trajectories with initial states drawn from $\mathcal{X}_{GT}$. Performance results are summarized in Table IV. Distance between players during the interactions are visualized in Fig. 5. Similar to Case 1, hybrid learning outperforms other methods. Value-hardening fails to generalize in this higher-dimensional case. Comparing with Case 1, we conjecture that the failure is caused by a decreasing probability of capturing informative samples (where rapid value changes happen) as the state dimension increases. Fig. 6 illustrates one initial state where the hybrid method induces a safe interaction while other baselines have collisions.

TABLE IV: Safety performance w/ activation function `tanh` under $\mathcal{X}_{GT}$

	HL	SL	SSL	VH
Col.%	**1.67%**	2.17%	98.8%	95.7%

V. Conclusions

We proposed a hybrid learning method that combines the advantages of supervised and self-supervised learning for approximating discontinuous value functions as solutions to two-player general-sum differential games. Using two vehicles interaction cases, we showed that the proposed method leads to better generalization and safety performance than purely supervised and self-supervised learning when using the same computational cost. In essence, our algorithm allows the value network to first adapt to the structure of the value function through supervised data, and then uses the underlying governing equations to satisfy the PDE in the entire state space. We also tested the idea of value hardening, and showed that without informative supervised data, this method does not guarantee convergence to the true values when discontinuity exists. Lastly, we empirically showed the importance of continuous differentiability of activation functions in solving PDEs with deep learning. It is worth noting that state constraints in differential games, which are the cause of large penalties that motivated this study, can also be incorporated into a different HJ formulation with an extended state space through the epigraph technique [35], [36]. This leads to value approximation in a higher dimensional state space yet allows the resulting value to be continuous on that space. Our future work will investigate the effectiveness of self-supervised learning for this new formulation of HJ equations.

REFERENCES

[1] M. G. Crandall and P.-L. Lions, "Viscosity solutions of hamilton-jacobi equations," *Transactions of the American mathematical society*, vol. 277, no. 1, pp. 1–42, 1983.

[2] I. M. Mitchell and C. J. Tomlin, "Overapproximating reachable sets by hamilton-jacobi projections," *journal of Scientific Computing*, vol. 19, no. 1, pp. 323–346, 2003.

[3] S. Bansal and C. J. Tomlin, "DeepReach: A deep learning approach to high-dimensional reachability," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1817–1824.

[4] 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," *arXiv preprint arXiv:2004.01806*, 2020.

[5] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

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

[7] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, "Adaptive activation functions accelerate convergence in deep and physics-informed neural networks," *Journal of Computational Physics*, vol. 404, p. 109136, 2020.

[8] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: a survey," *Journal of Marchine Learning Research*, vol. 18, pp. 1–43, 2018.

[9] R. Bellman and R. E. Kalaba, *Dynamic programming and modern control theory*. Citeseer, 1965, vol. 81.

[10] J. Han, A. Jentzen, and E. Weinan, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 34, pp. 8505–8510, 2018.

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

[12] K. Ito, C. Reisinger, and Y. Zhang, "A neural network-based policy iteration algorithm with global h2-superlinear convergence for stochastic games on domains," *Foundations of Computational Mathematics*, vol. 21, no. 2, pp. 331–374, 2021.

[13] M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione, "Regret minimization in games with incomplete information," *Advances in neural information processing systems*, vol. 20, 2007.

[14] J. Perolat, B. De Vylder, D. Hennes, E. Tarassov, F. Strub, V. de Boer, P. Muller, J. T. Connor, N. Burch, T. Anthony *et al.*, "Mastering the game of stratego with model-free multiagent reinforcement learning," *Science*, vol. 378, no. 6623, pp. 990–996, 2022.

[15] O. Tammelin, N. Burch, M. Johanson, and M. Bowling, "Solving heads-up limit texas hold'em," in *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[16] N. Brown, A. Bakhtin, A. Lerer, and Q. Gong, "Combining deep reinforcement learning and search for imperfect-information games," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 057–17 069, 2020.

[17] J. N. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch, "Learning with Opponent-Learning Awareness," *arXiv:1709.04326 [cs]*, Sep. 2017, arXiv: 1709.04326.

[18] 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," *Autonomous Robots*, vol. 42, no. 7, pp. 1405–1426, Oct. 2018.

[19] 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 *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 43–52.

[20] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 50, pp. 24 972–24 978, 2019.

[21] Z. Zahedi, A. Khayatian, M. M. Arefi, and S. Yin, "Seeking nash equilibrium in non-cooperative differential games," *Journal of Vibration and Control*, p. 10775463221122120, 2022.

[22] 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, p. 110076, 2022.

[23] S. Nikolaidis, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in collaborative tasks: Models and experiments," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 618–634, 2017.

[24] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2111–2117.

[25] C. Peng and M. Tomizuka, "Bayesian persuasive driving," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 723–729.

[26] Y. Wang, Y. Ren, S. Elliott, and W. Zhang, "Enabling courteous vehicle interactions through game-based and dynamics-aware intent inference," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, pp. 217–228, 2019.

[27] D. Fridovich-Keil, A. Bajcsy, J. F. Fisac, S. L. Herbert, S. Wang, A. D. Dragan, and C. J. Tomlin, "Confidence-aware motion prediction for real-time collision avoidance1," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 250–265, 2020.

[28] 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 *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2761–2767.

[29] 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 *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 1475–1481.

[30] A. W. Starr and Y.-C. Ho, "Nonzero-sum differential games," *Journal of optimization theory and applications*, vol. 3, no. 3, pp. 184–206, 1969.

[31] E. Cristiani and P. Martinon, "Initialization of the shooting method via the hamilton-jacobi-bellman approach," *Journal of Optimization Theory and Applications*, vol. 146, no. 2, pp. 321–346, 2010.

[32] A. Daw, J. Bu, S. Wang, P. Perdikaris, and A. Karpatne, "Rethinking the importance of sampling in physics-informed neural networks," *arXiv preprint arXiv:2207.02338*, 2022.

[33] X. Chen, Z. Li, Y. Yang, L. Qi, and R. Ke, "High-resolution vehicle trajectory extraction and denoising from aerial videos," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3190–3202, 2020.

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

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

[36] D. Lee, "Safety-guaranteed autonomy under uncertainty," Ph.D. dissertation, UC Berkeley, 2022.