IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Inferring Foresightedness in Dynamic Noncooperative Games

Cade Armstrong*, Ryan Park*, Xinjie Liu, Kushagra Gupta, and David Fridovich-Keil

***Abstract*—Dynamic game theory is an increasingly popular tool for modeling multi-agent, e.g. human-robot, interactions. Game-theoretic models presume that each agent wishes to minimize a private cost function that depends on others' actions. These games typically evolve over a fixed time horizon, specifying how far into the future each agent plans. In practical settings, however, decision-makers may vary in foresightedness, or how much they care about their current cost in relation to their past and future costs. We conjecture that quantifying and estimating each agent's foresightedness from online data will enable safer and more efficient interactions with other agents. To this end, we frame this inference problem as an *inverse* dynamic game. We consider a specific objective function parametrization that smoothly interpolates myopic and farsighted planning. Games of this form are readily transformed into parametric mixed complementarity problems; we exploit the directional differentiability of solutions to these problems with respect to their hidden parameters to solve for agents' foresightedness. We conduct three experiments: one with synthetically generated delivery robot motion, one with real-world data involving people walking, biking, and driving vehicles, and one using high-fidelity simulators. The results of these experiments demonstrate that explicitly inferring agents' foresightedness enables game-theoretic models to make 33% more accurate models for agents' behavior.**

I. Introduction

Robot planning problems often involve strategic interactions between human and robotic decision makers. Dynamic game theory captures many of these problems' complexities, such as conflicting goals, coupled strategies, and long-term consequences. However, for an agent to use game theoretic planning algorithms, it must understand other agents' hidden objectives, or objectives that are not known *a priori*. To remedy this, we can cast inferring hidden objectives as an inverse game [1]–[3]. In practice, agents can use algorithms to simultaneously infer others' objectives and plan trajectories that seamlessly interact with one another over time.

Existing inverse game solvers often assume that decision makers care equally about each moment in time, but in practice, humans and robots weigh future costs differently than present costs. For example, in an urban driving game modeling a busy intersection, a farsighted driver may slow down further in advance of the intersection than a short-sighted, or myopic, driver would. Robots incorrectly predicting the myopic driver's foresight may take prematurely evasive actions, resulting in unexpected and unnecessarily risky behavior.

This work was supported by the National Science Foundation under awards 2336840 and 2211548, and by the Army Research Laboratory under Cooperative Agreement Number W911NF-25-2-0021. All authors are with the Oden Institute for Computational Sciences and Engineering, University of Texas at Austin, 201 E 24th St, Austin, TX. {cadearmstrong, ryanjpark, xinjie-liu, kushagrag, dfk}@utexas.edu Digital Object Identifier (DOI): see top of this page.

*indicates equal contribution (Corresponding author: Cade Armstrong).

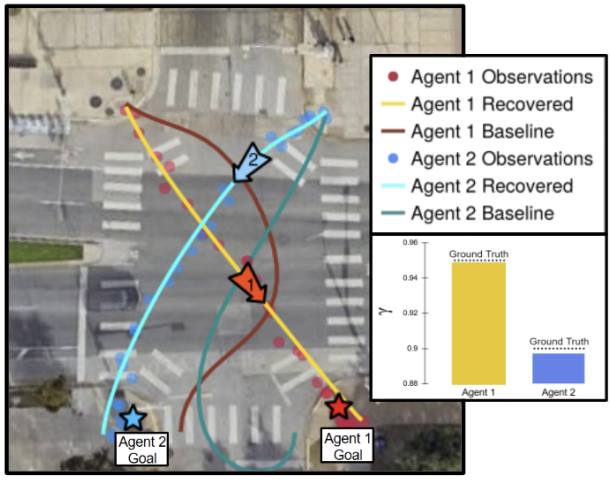


Fig. 1. Representative example showcasing how the proposed formulation improves our ability to model agents' behavior in noncooperative interactions. Here, two simulated delivery robots are crossing an intersection, and by correctly inferring their degree of foresightedness (i.e., the γ parameter in the inset), our method recovers their trajectories than a baseline approach.

This paper aims to address the gap between the assumptions of existing game theoretic algorithms and decision makers' foresightedness. In particular, existing methods typically model each agent's objective as a sum of cost functions which can depend upon all agents' states and actions. In this work, we consider a time-discounted cost formulation, which smoothly interpolates foresight for each agent. Additionally, we develop a gradient-based algorithm for inverting the corresponding game and identifying each agent's degree of foresightedness. Our specific contributions are: 1) an explicitly foresighted game formulation, 2) an efficient method to infer each agent's degree of foresightedness from data collected online, and 3) and a series of simulated and real-world experiments that demonstrate the benefits of modeling agents' foresight.

In particular, we conduct one simplified experiment with two myopic delivery robots (presented in Figure 1) and two simulated experiments with real-world data (presented in Figure 2 and Figure 3). Results demonstrate that our method recovers more accurate models of foresighted agents than existing models, as well as being more robust to noise in partially observable settings. Specifically, our method reduces trajectory error by over 33% on average in comparison to cutting-edge baseline methods.

II. Related Works

A. Noncooperative Dynamic Games

An N-agent dynamic game models each agent's choice of strategy in terms of a optimization problem, whose objective and/or constraints can depend upon other agents' strategies. Dynamic games admit different solutions depending on the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

information available to each agent at every time step [4]. In this work, we consider the *Nash* equilibrium concept in the setting of *open-loop* strategies, i.e., we presume that at each time t agents only have information about the initial state of the game. Early works in this setting focused primarily on problems with convex, quadratic costs and linear game dynamics, without any additional constraints [4]–[7]. However, finding a Nash equilibrium in more general settings is often intractable [8]. Consequently, many recent works develop iterative trajectory optimization methods to find *approximate, local* open-loop Nash equilibria in games with nonlinear costs and complex constraint structures. Such approximations satisfy first-order optimality conditions of the underlying optimization problem for each agent, and second-order conditions for local optimality must generally be checked *a posteriori*. Some well-known methods include root-finding techniques for the underlying joint N-agent Karush-Kuhn-Tucker (KKT) system [9]–[11] and iterative best-response algorithms to identify local Nash equilibria [12], [13].

B. Inverse Dynamic Games

Inverse dynamic games seek to identify unknown aspects of game models, e.g., agents' intentions and constraints, from observed interactions. Early approaches minimize the residual of the KKT system to identify possible game parameter values [14]–[16]. However, this type of approach requires *full* state-action demonstration to evaluate the residuals, posing a challenge for online estimation tasks, where not all agents' states and actions are available. Worse, the assumption that the demonstrations satisfy first-order optimality conditions means that such residual minimization approaches can perform poorly in the presence of observation noise [17]. To mitigate these issues, recent work formulates inverse game problems in terms of maximum likelihood estimation (MLE), and explicitly impose first-order optimality conditions as constraints [17]–[20]. In particular, our work builds directly upon this line of work, and specifically extends the solution approach of [18], which develops a differentiable game solver that enables inverse game problems to be solved via gradient descent on the unknown game parameters.

While MLE approaches to inverse games demonstrate strong performance in various scenarios, they are fundamentally limited to providing point estimates of unknown parameters. To this end, another line of work seeks to infer full Bayesian posterior *distributions* for the unknown game parameters, e.g., via particle filtering [21] and unscented Kalman filtering [22]. More recently, work in [23] proposes to embed a differentiable Nash solver into generative models to solve variational inference problems. However, high computational costs hinder these approaches from scaling to larger systems.

Finally, we note that inverse games have also been formulated in maximum-entropy settings [24], inferring opinion dynamics model [20], and in combination with neural network components [25].

C. Non-Game Theoretic Multi-Agent Modeling

Many non-game theoretic approaches have been proposed for modeling multi-agent robotic systems. Social forces have been used to model multi-agent interactions, where agent objectives are introduced through virtual forces (e.g., spring forces to attract an agent) [26], [27]. Potential fields have been used similarly; instead of virtual forces to encode goals and obstacles, the method uses potential fields to repel and attract agents [28]. Relevant works model multi-agent behavior by encoding agent objectives in velocity space rather than position space with velocity obstacles [29], [30]. Additionally, recent work applies machine learning to predict human driving and pedestrian behavior. Maximum entropy inverse reinforcement learning [31], [32] and history-dependent LSTM-based models [33], [34] are examples of learning methods that can forecast vehicle and pedestrian trajectories. Finally, recent works use generative trajectory models to predict pedestrian behavior [35], [36]. Unlike game-theoretic methods, these methods do not explicitly reason about coupling effects in agents' decision-making. Additionally, it should be noted that our approach exhibits the interpretability of classical methods, such as social force models, but can flexibly accommodate arbitrary specifications of agents' objective functions, similar to inverse reinforcement learning methods.

III. Background

For notational convenience, lowercase bold variables refer to aggregations over time and the omission of agent indices denotes aggregations of all agents. In addition, agent indices are always written as superscripts while time indices are always written as subscripts.

A. Dynamic Games

A dynamic game [4] is characterized by N agents, with the i^{th} agent's control input denoted $u_t^i \in \mathbb{R}^{m^i}$ for all discrete times $t \in [T] := \{1, 2, ..., T\}$ and a joint state variable $x_t \in \mathbb{R}^n$ following given dynamics $x_{t+1} = f_t(x_t, u_t^1, u_t^2, \dots, u_t^N)$. Each agent has a cost function

$$J^i := \sum_{t=1}^{T} \Gamma^i(t; \gamma^i) C^i(x_t, u_t^i, u_t^{\neg i}; \theta^i), \tag{1}$$

which depends upon the state and its own actions, as well as others' actions $u_t^{\neg i}$, hidden parameters $\theta^i \in \mathbb{R}^k$, and *discount factor* γ^i. At each time t, agent i's cost is comprised of parametrized function $C^i(\cdot)$ and scaled by a parametrized *discounting function* $\Gamma^i(t; \gamma^i)$, quantifying the importance of agent i's cost at time t.

In principle, the formulation of the game in (1) is general enough to handle arbitrary functions Γ^i, which are twice differentiable in γ^i. In this work, we assume that this discounting function is an exponential, i.e. $\Gamma^i(t; \gamma^i) = (\gamma^i)^t$, where $\gamma^i \in (0, \infty]$.

We shall refer to all agents' discount factors as a vector $\gamma = (\gamma^1, \dots, \gamma^N) \in [0, \infty]^N$ for convenience. Observe that the function $\Gamma^i(t; \gamma^i)$ rapidly approaches 0 as $t \to T$ when $\gamma^i < 1$. Thus, the i^{th} agent's actions do not depend upon costs incurred past some effective time horizon $\tilde{T}^i \in [T]$.

Thus, we refer to the sequence of game states as $\mathbf{x} = (x_1^\top, x_2^\top, \dots, x_T^\top)^\top$, agent i's control sequence as $\mathbf{u}^i =$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

$(u_1^{i,\top}, u_2^{i,\top}, \ldots, u_T^{i,\top})^\top$, the sequence of all agents' actions as $\mathbf{u} = (\mathbf{u}^{1,\top}, \mathbf{u}^{2,\top}, \ldots, \mathbf{u}^{N,\top})^\top$, and all agents' hidden parameters as $\theta = (\theta^{1,\top}, \theta^{2,\top}, \ldots, \theta^{N,\top})^\top$. Then, we can write each cost function as $C_t^i(x_t, u_t^i, u_t^{\neg i}; \theta^i) = C_t^i(x_t, u_t; \theta^i)$ and thus agent i's overall cost is $J^i(\mathbf{x}, \mathbf{u}; \gamma^i, \theta^i)$ with a slight abuse of notation. In general, we may also assign each agent a set of inequality constraints $I^i(\mathbf{x}, \mathbf{u}; \theta^i) \geq 0$ and a set of equality constraints $E^i(\mathbf{x}, \mathbf{u}; \theta^i) = 0$. Finally, we can define a game as a tuple:

$$\mathcal{G}(\theta, \gamma) = (\{J^i(\cdot; \gamma^i, \theta^i)\}_{i\in[N]}, \{I^i(\cdot; \theta^i)\}_{i\in[N]}, \{E^i(\cdot; \theta^i)\}_{i\in[N]}, x_1, T, N). \tag{2}$$

Note that (2) does not include the dynamics $f_t(\cdot)$ explicitly; this is because the dynamics generate $(T-1)$ equality constraints of the form $x_{t+1} - f_t(x_t, u_t^1, u_t^2, \ldots, u_t^N) = 0$ for each of the N agents. For clarity, we assume that $E^i(\cdot)$ contains these terms.

B. Generalized Open-Loop Nash Equilibria

For a given set of parameters θ and discount factors γ, a generalized open-loop Nash equilibrium (GOLNE) of the game $\mathcal{G}(\theta, \gamma)$ in (2) is given by a point $(\mathbf{x}^*, \mathbf{u}^*)$ which jointly solves the following coupled optimization problems:

$$\forall i \in [N] \begin{cases} \min\limits_{\mathbf{x}, \mathbf{u}^i} \sum\limits_{t=1}^{T} \Gamma^i(t; \gamma^i) C^i(x_t, u_t^i, u_t^{\neg i}; \theta^i) & \text{(3a)} \\ \text{s.t. } E^i(\mathbf{x}, \mathbf{u}; \theta^i) = 0 & \text{(3b)} \\ \quad\quad I^i(\mathbf{x}, \mathbf{u}; \theta^i) \geq 0. \end{cases}$$

The strategies $\mathbf{u}^* = (\mathbf{u}^{1*}, \mathbf{u}^{2*}, ..., \mathbf{u}^{N*})$ have the property that $J^i(\mathbf{x}, \mathbf{u}^i, \mathbf{u}^{\neg i*}; \theta^i, \gamma^i) \geq J^i(\mathbf{x}^*, \mathbf{u}^*; \theta^i, \gamma^i)$ for all feasible $\mathbf{u}^i$ and $\mathbf{x}$, for all agents $i \in [N]$.

C. Mixed Complementarity Problem (MiCP)

Finding a generalized Nash equilibrium is computationally intractable [37]; therefore, it is common to relax the condition in Section III-B to hold only within an open neighborhood of the point $(\mathbf{x}^*, \mathbf{u}^*)$ [10], [11], [38]–[40]. Such points are called local Nash equilibria [41]. Practically, such points can be identified by solving agents' first order necessary conditions, which constitute a Mixed Complementarity Problem (MiCP) [42]. An MiCP is defined by decision variables $r \in \mathbb{R}^{\eta_r}, z \in \mathbb{R}^{\eta_z}$, as well as functions $c(r, z)$ and $h(r, z)$, such that $c(r, z) = 0$ and $0 \leq z \perp h(r, z) \geq 0$. MiCPs can often be solved efficiently via off-the-shelf solvers such as PATH [43].

IV. Problem Statement

We presume that an observer has obtained noisy sensor measurements of the game state over time, and denote these observations $\mathbf{y} = \big[y_1(x_1), y(x_2), y_3(x_3), \ldots, y_T(x_T)\big]$. We assume observations are drawn independently from Gaussian models $y_t(x_t) \sim \mathcal{N}\big(h_t(x_t), \Sigma_t\big)$ with known covariances Σ_t, where $h_t(x_t)$ describes the expected output of the sensor at time t.

Given measurements $\mathbf{y}$, we seek to identify game parameters $\hat{\theta}, \hat{\gamma}$ that maximizes the likelihood to have observed $\mathbf{y}$ from a GOLNE of $\mathcal{G}(\hat{\theta}, \hat{\gamma})$. Equivalently, one can minimize covariance-weighted deviations from expected measurements, which we denote $\mathcal{P}$.

Problem 1. ***Inverse Game Problem:*** *Given a sequence of observations* $\mathbf{y}$*, find parameters* $\hat{\theta}$ *which solve*

$$\min_{\mathbf{x}, \mathbf{u}, \theta, \gamma} \overbrace{\sum_{t=1}^{T} (h_t(x_t) - y_t)^\top \Sigma_t^{-1} (h_t(x_t) - y_t)}^{\mathcal{P}(\mathbf{x}(\theta, \gamma))} \tag{4a}$$

$$\text{s.t.} \quad (\mathbf{x}, \mathbf{u}) \text{ is a GOLNE of } \mathcal{G}(\theta, \gamma). \tag{4b}$$

Intuitively, Problem 1 can be separated into an outer *inverse* problem and an inner *forward* problem. The forward GOLNE problem (4b) is parametrized by the game parameters θ and discount factors γ, which are decision variables in the inverse problem (4a). This formulation is general enough to handle partial state observations, making it amenable to realistic observation models such as cameras or GPS, in principle. However, solving (4a) entails computational challenges. In particular, the KKT conditions of (4b) are certainly nonlinear in γ, making the overall problem non-convex. In addition, (4b) may contain inequality constraints, which imply that its KKT conditions involve complementarity conditions which make the overall problem non-smooth.

V. Solution Approach

In this section, we present a constrained gradient descent algorithm for identifying unknown parameters θ, γ in Problem 1. This technique will require us to take derivatives of solutions $(\mathbf{x}, \mathbf{u})$ to (4b) with respect to parameters (θ, γ). Therefore, we begin with a discussion of transforming a GOLNE into an MiCP, whose solutions are directionally differentiable with respect to problem parameters [42, Ch. 5].

A. Equilibrium Constraint as an MiCP

In this subsection, we show how to convert a GOLNE into a MiCP which encodes its first-order necessary conditions. Similar techniques have been used in [18], [44]–[46] to solve inverse optimal control and game problems.

We start with writing the agents' first-order necessary conditions. First, we introduce Lagrange multipliers λ^i and μ^i for the i^{th} agent's inequality and equality constraints, respectively, and write its Lagrangian as

$$\mathcal{L}^i(\mathbf{x}, \mathbf{u}, \lambda^i, \mu^i; \theta^i, \gamma^i) = C^i(\mathbf{x}, \mathbf{u}; \theta^i, \gamma^i) - \lambda^{i,\top} I^i(\mathbf{x}, \mathbf{u}; \theta^i) - \mu^{i,\top} E^i(\mathbf{x}, \mathbf{u}; \theta^i). \tag{5}$$

Then, when the gradients of the constraints are linearly independent at a candidate solution point (i.e., the linear independence constraint qualification is satisfied [42, Chapter 3.2]), the following Karush-Kuhn-Tucker (KKT) conditions must hold for each agent i:

$$\nabla_{\mathbf{x}} \mathcal{L}^i = 0, \nabla_{u^i} \mathcal{L}^i = 0, E^i = 0 \tag{6a}$$

$$0 \leq \lambda^i \perp I^i(\mathbf{x}, \mathbf{u}; \theta^i) \geq 0. \tag{6b}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

We can then structure agents' joint KKT conditions as a parametric MiCP, in which the primal and dual variables are concatenated as $r = (\mathbf{x}^\top, \mathbf{u}^\top, \mu^{1,\top}, \mu^{2,\top}, \dots, \mu^{N,\top})^\top$ and $z = (\lambda^{1,\top}, \lambda^{2,\top}, \dots, \lambda^{N,\top})^\top$. For brevity, define $v = (r^\top, z^\top)^\top$. Then, with a slight abuse of notation, the parameterized MiCP for each agent can be written as described in Section III-C, where $c(v;\theta,\gamma) = [\left(\nabla_{\mathbf{x}} L^i\right)^\top_{i\in[N]}, \left(\nabla_{\mathbf{u}^i} L^i\right)^\top_{i\in[N]}, \left(E^i\right)^\top_{i\in[N]}]^\top$ and $h(v;\theta,\gamma) = [\left(I^i\right)_{i\in[N]}]^\top$. For brevity, we will define $F(v;\theta,\gamma) = [c(\cdot)^\top, h(\cdot)^\top]^\top$. For additional details on connections between GOLNEs and MiCPs in the context of open loop dynamic games, we direct the reader to [18], [42].

B. Optimizing Parameters with Gradient Descent

In this section, we present our algorithm for solving Problem 1. We first replace (4b) with the KKT conditions in (6), and transcribe them into an MiCP according to Section V-A. Then, we compute the total derivative of objective function $\mathcal{P}$ with respect to parameters (θ,γ), and update their values accordingly. These gradients can be computed via the chain rule:

$$\nabla_{(\theta,\gamma)}\mathcal{P}(\mathbf{x}(\theta,\gamma)) = (\nabla_{(\theta,\gamma)} v)^\top (\nabla_v \mathbf{x})^\top (\nabla_{\mathbf{x}}\mathcal{P}). \tag{7}$$

The only non-trivial term in (7) is $\nabla_{(\theta,\gamma)} v$. Next, we show how to take directional derivatives of MiCPs at a solution v^* where strictly complementarity holds.

First, we must consider the complementarity constraints on z and $h(\cdot)$. To this end, we construct an index set $\mathcal{I}$ which records all inactive inequality constraint dimensions in $h(v;\theta,\gamma)$. Indexing F at elements of this set yields a vector $[F]_{\mathcal{I}}$. From Section III-C, we know that these inactive inequalities $[F]_{\mathcal{I}}$ are strictly positive, and the Lagrange multipliers associated with these constraints are exactly 0. Then, presuming the continuity of F, small changes in (θ,γ) preserve the positivity of elements of $[F]_{\mathcal{I}}$ and force the corresponding Lagrange multipliers in $[v]_{\mathcal{I}}$ to remain 0. Thus, we find that $\nabla_{(\theta,\gamma)}[v]_{\mathcal{I}} = 0$.

Consider the remaining constraints from (6), which must be active due to strict complementarity; denote the indices corresponding to these constraint as $\mathcal{S}$. We can use the implicit function theorem and stationarity of F with respect to v to write:

$$0 = \nabla_{(\theta,\gamma)}[F]_{\mathcal{S}} = \nabla_{(\theta,\gamma)}[F]_{\mathcal{S}} + (\nabla_v [F]_{\mathcal{S}})(\nabla_{(\theta,\gamma)} v) \tag{8a}$$

$$\implies \nabla_{(\theta,\gamma)} v = -(\nabla_v [F]_{\mathcal{S}})^{-1} \nabla_{(\theta,\gamma)}[F]_{\mathcal{S}}. \tag{8b}$$

Then, when $\nabla_v [F]_{\mathcal{S}}$ is invertible, we can find exact values of $\nabla_{(\theta,\gamma)} v$. For a more complete treatment, including a discussion of weak complementarity, readers are encouraged to consult [18], [42], [47].

Thus equipped, we iteratively update our estimate of (θ,γ) according to the gradient of (4a)—incorporating the aforementioned implicit derivatives as needed. Algorithm 1 terminates after a maximum number of iterations (i.e., when $k > K$) or when parameters have converged (i.e., $\|\theta_{k+1} - \theta_k\|_2 \le \epsilon$). In our experiments, $K = 500$ and $\epsilon = 10^{-4}$. At each step, also note that we project the discount factor γ onto the set $[0,\infty)^N$ to ensure feasibility.

Algorithm 1: Foresight-aware inverse game

Hyper-parameters: Learning rate α
Input: initial θ, initial γ, observations $\mathbf{y}$
$\theta_0 \leftarrow \theta$
$\gamma_0 \leftarrow \gamma$
$k \leftarrow 0$
while *not converged* **do**
  $v_k \leftarrow \text{solveInnerMCP}(\theta_k, \gamma_k)$ ▷ Section V-A
  $\nabla_{(\theta_k,\gamma_k)}\mathcal{P} \leftarrow \text{calcGrad}(v_k, \theta_k, \gamma_k)$ ▷ Section V-B
  $\theta_{k+1} \leftarrow \theta_k - \nabla_{\theta_k}\mathcal{P} \cdot \alpha$
  $\gamma_{k+1} \leftarrow \max(0, \gamma_k - \nabla_{\gamma_k}\mathcal{P} \cdot \alpha)$
  $k \leftarrow k + 1$
return $(\theta_k, \gamma_k, \mathbf{x}, \mathbf{u})$

VI. Experiments

In this section, we test the performance and robustness of the proposed approach in Section V and compare it to baseline inverse game approaches in simulated and real-world experiments. Results show that Algorithm 1 performs better than an existing inverse game formulation across various performance metrics.

A. Example Game - Crosswalk

Our first experiment involves $N = 2$ pedestrians navigating a crosswalk at an intersection on the UT Austin campus at W Dean Keaton St. and Whitis Ave. This intersection becomes safe for all pedestrian crossing at once, encouraging students to cross diagonally and interact with one another. We simulate two short-sighted delivery robots who begin at the top right and left corners, respectively, and their goals are to reach the bottom left and right corners, respectively.

We model agents as delivery robots wishing to minimize the distance to their goal and their control effort, such that the i^{th} agent's cost can be written as:

$$J^i_{\text{stud}} = \sum_{t=0}^{T} (\gamma^i)^t \left[w^i_{\text{goal}} \, \|p^i_t - \theta^i_{\text{goal}}\|_2^2 + w^i_{\text{ctrl}} \, \|u^i_t\|_2^2 \right]. \tag{9}$$

For brevity, we define $p^i_t \in \mathbb{R}^2$ to denote agent i's position at time t, weigh the goal and control cost terms with w^i_{goal} and w^i_{ctrl}, and parameterize the game with the i^{th} agent's target position $\theta^i_{\text{goal}} \in \mathbb{R}^2$. We model the robots as point masses with bounds on the velocity and control inputs, assume a time discretization of $\Delta = 0.1\,\text{s}$, and play the game over $T = 25$ time steps. Explicitly, the game's state at time t is composed as $x_t = [x_t^{1\top}, x_t^{2\top}]^\top$, and each agent's physical dynamics are modeled with a standard double integrator linear system.

Additionally, the i^{th} agent has a collision avoidance constraint defined as

$$I^i(\cdot) = \left[\left(\|p^i_t - p^{\neg i}_t\|_2^2 - \delta \cdot d_{\text{min}} \right)_{t\in[T]} \right] \ge 0, \tag{10}$$

where d_{min} denotes the minimum allowed distance between the agents and $\delta > 1$ is a safety margin. However, by moving this constraint to each agent's objective, we can avoid assigning a new Lagrange multiplier to each pair of agents. This avoids

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

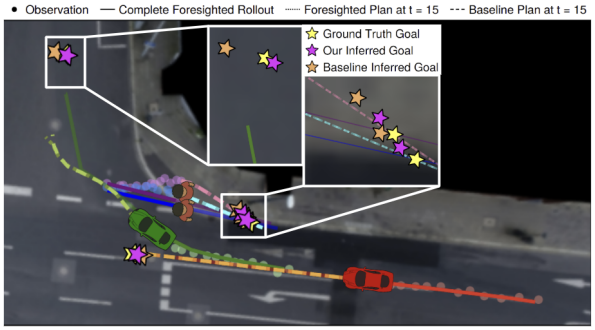


Fig. 2. Intersection from the InD dataset [48], overlaid with trajectories for all four agents generated in a receding horizon fashion using our method and the baseline. Note that we reduced the opacity of the complete foresighted rollout after t = 15 for visual clarity.

quadratic scaling in the number of MiCP variables with respect to number of agents. Therefore, we model i^{th} agent's objective function in the inverse game as $J^i_{\text{ped}} = J^i_{\text{stud}} + J^i_{\text{coll}}$, where

$$J^i_{\text{coll}} = \sum_{t=0}^{T} (\gamma^i)^t \; w^i_{\text{coll}} \; \max(0, \delta \cdot d_{\min} - \|p^i_t - p^{\neg i}_t\|_2^2), \quad (11)$$

in which w^i_{coll} is a weighting parameter quantifying the importance of collision avoidance for the i^{th} agent.

B. Example Game - Intersection

Our second experiment involves $N = 4$ agents navigating an intersection: two pedestrians crossing a crosswalk, a car attempting to turn while the pedestrians are crossing, and a car attempting to go straight while behind the turning car. We employ real-world data of this interaction from the Intersection Drone (InD) dataset, captured by leveLXData via a drone hovering over the intersection found in Aachen, Germany, seen in Figure 2 [48]. Specifically, the interaction takes place over 162 frames at 25 frames per second. The data was downsampled by a factor of 6, resulting in a game horizon of $T = 28$.

The setup for our model in this example is very similar to that discussed in Section VI-A. To begin, all agents have inequality constraints (10).

We model the cars' physical dynamics as bicycles with bounds on the control inputs. Explicitly, the game's state at time t is composed as $x_t = [x_t^{1\top}, x_t^{2\top}]^\top$ in which $x^i_t = [p^i_{x,t}, p^i_{y,t}, v^i_t, \psi^i_t]^\top$, and for the i^{th} agent:

$$\begin{aligned} x^i_{t+1} =& [p^i_{x,t},\ p^i_{y,t},\ v^i_t,\ \psi^i_t]^\top + \\ & \Delta[v^i_t \cos(\psi^i_t),\ v^i_t \sin(\psi^i_t),\ a^i_t,\ v^i_t \tan(\phi^i_t)/l]^\top \end{aligned}$$

with control input $u^i_t = [a^i_t, \phi^i_t]^\top$ representing longitudinal acceleration and steering angle (v^i_t is longitudinal velocity, ψ^i_t is heading, and l is wheelbase). Meanwhile, the pedestrians have the same dynamics as discussed in Section VI-A.

Depending on whether the i^{th} agent is a car or pedestrian, we model the agent's objective function slightly differently. For the pedestrians, their cost is exactly J^i_{ped} while cars also minimize the distance to their lane centers. We model the i^{th} agent's lane center as a set, $\mathcal{C}^i = \{(c^i_{x,t}, c^i_{y,t})\}_{t=0}^T$

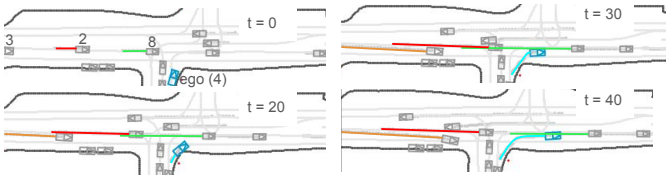


Fig. 3. Snapshots from receding horizon inference and planning with the Waymax simulator, for a game involving the blue, green, yellow, and red agents. The ego robot (blue) employs our method, while the other cars are controlled by the Waymax simulator.

generated from a polynomial curve fit to lane center points. We then can write the cost as:

$$J^i_{\text{car}} = J^i_{\text{ped}} + \sum_{t=0}^{T} (\gamma^i)^t \; w^i_{\text{lane}} \; \|p^i_t - c^i_t\|_2^2, \quad (12)$$

where w^i_{lane} weighs the lane center cost term and c^i_t refers to the t^{th} element of set $\mathcal{C}^i$.

Lastly, the environment in this example has defined regions in which the agents are unable to traverse (e.g. buildings). Thus, we create inequality constraints such that the agents are constrained to the roads/sidewalks. To do so, we model corners as circles and use sigmoids for the sides of the road.

C. Example Game - Waymax Simulation

Our last experiment involves $N = 4$ cars navigating an intersection, one of which is a robot attempting to make a left turn (cf. Figure 3). We employ the Waymax simulator, which has been built atop the Waymo Open Motion Dataset [49], to simulate the humans' responses to the ego's actions. The ego robot infers the humans' objectives and foresightedness and plans to reach its goal safely in a receding horizon setup. Specifically, the scenario takes place over a horizon of $T = 50$ steps with a time discretization of $\Delta = 0.1\,\text{s}$ and a planning horizon $T_{\text{plan}} = 11$ steps.

Like in Section VI-B, we model the cars' physical dynamics as bicycles with bounds on the control inputs. However, we modify the costs from the previous examples by: 1) simplifying the lane center term to only consider $p^i_{x,t}$ as the road is a relatively straight line after the turn, and 2) adding a quadratic target velocity tracking cost. Lastly, it should be noted that the game is constructed such that the ego agent has no parameters in its own cost or constraints, so that it is only inferring the parameters of the simulated agents.

D. Experimental Setup

For our first two examples, we aim to show that our method predicts trajectories more accurately than a baseline method, which solves the same inverse problem *but presumes agents have a discount factor of* 1.[1] Thus, the baseline estimator has fewer hidden parameters to estimate than our method.

To ensure a fair comparison, we use the same MiCP backend [43] to satisfy the first-order necessary conditions of the GOLNE at line 7 of Algorithm 1, the same observation sequence, and the same initial guess—our method receives

[1]Full implementation details can be found at https://github.com/cadearmstrxng/InverseGameDiscountFactor.jl.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

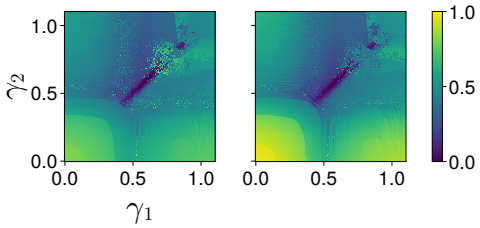
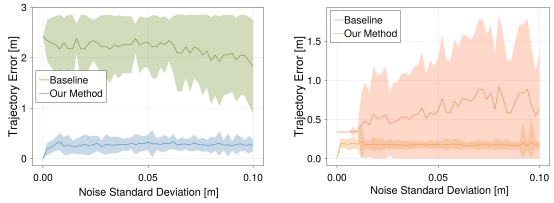


Fig. 4. Heatmap of $\mathcal{P}$ from Problem 1 for the crosswalk experiment for the fully observable (left) and partially observable (right) cases. Both players' γ are varied while holding θ constant at the ground truth values and the resulting cost is scaled to be in between $[0, 1]$.



(a) Only position measurements (b) Full state measurements

Fig. 5. Algorithm 1 reliably improves predicted trajectory error in the crosswalk experiment, for partial and full state observations. Solid/dotted lines denote means and the opaque band indicates standard deviation.

a uniformly random $\gamma \in [0, 1]$ as an initial estimate. To compare the two methods, we calculate their respective error as error $= \|\mathbf{x} - \mathbf{x}_{\text{ground-truth}}\|_2^2$, enabling us to determine how well the methods predicted the agents' trajectories. We gather these results with a Monte Carlo study done over a range of observation noise covariances as previously discussed. For full state experiments, the mean, $h(x_t)$, is the identity function. For partial state experiments, $h(x_t) = [p_t^{i\top} \ \forall i \in N]^\top$. The covariance is given by identity matrices scaled by 50 evenly spaced constants between 0 and $0.01\,\mathrm{m}^2$. For each covariance matrix, we generated 50 noisy observation sequences given to our method and the baseline method, resulting in 2500 observation sequences and parameter estimates. Results are shown in Figure 5 and Figure 6, and discussed below.

For the Waymax example, we aim to show our method can be used in a robotics context safely and efficiently by employing it in a receding horizon planner. At each time t, the ego robot infers other agents' cost parameters from the past 10 observations and plans over the next 10 time steps. The action used at t is the first action taken in the planning trajectory. To evaluate safety and efficiency, we simulate 50 random sequences of observation noise with standard deviation $0.1\,\mathrm{m}$ and plot the ego robot's distance to each other robot and to its goal position over the time horizon, shown in Figure 7. Results are qualitatively similar at lower noise levels.

E. Monte Carlo Results - Crosswalk

Figure 1 shows a representative trial for our running example at a simulated crosswalk. It shows the two players' observations, initial positions, ground-truth goals, inferred goals for the baseline and our method, and current and planned trajectories at a snapshot in time. Note that our method results in a trajectory closer to the observations in L_2 distance than the baseline. In addition, according to the bar graph, our method accurately recovers the discount factor within 0.3%.

Figure 4 shows a heatmap visualizing the objective function for Problem 1 in which θ is fixed to the ground truth values, in both fully- and partially-observed settings. The diagonally-shaped low-cost region implies that the *equilibrium* of the game is relatively insensitive to the agents' absolute discount factors as long as both agents have similar discount factors (and both are greater than ≈ 0.5). We expect this low-cost region to exist because of the game's highly symmetric structure, motivating us to introduce a *regularization* scheme for the inverse game in Problem 1 to encourage better solver performance. To do so, we added the term $c_\gamma \|1 - \gamma\|_2^2$ to (4a), which encodes a prior belief that agents are more farsighted; it is straightforward to instead penalize $\|\gamma\|_2^2$ and encode the belief that agents are more shortsighted. In the following Monte Carlo results, we report results for $c_\gamma = 10^{-3}$. For clarity, we do not regularize in the other, asymmetric experiments.

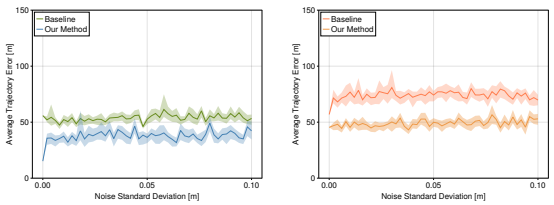
Figure 5 shows the mean trajectory estimation error as a function of the observation noise standard deviation for Algorithm 1 and the $\gamma = 1$ baseline. Our method substantially outperforms the baseline in both cases. Interestingly, we can see that the mean error is only marginally sensitive to noise for both methods. Rather, the most noticeable effect of noise is an increase in the standard deviation of the errors. From these results, we conclude that our foresighted formulation can better predict the behavior of short-sighted agents in comparison to a state-of-the-art inverse game baseline [18].

F. Monte Carlo Results - InD Intersection

As previously discussed, Figure 2 shows a representative trial of our method and the baseline with the real-world data from the InD dataset [48]. It shows the initial positions, the perceived and inferred goals, the observations and recovered trajectories up to a time t = 15, and the planned trajectories for all four players at t = 15 for the baseline and our method. Our visualization shows a closer recovery of the ground-truth goal position than the baseline. In addition, we recover a trajectory closer to the observations in L_2 distance than baseline.

Figure 6 shows the mean trajectory error as a function of the observation noise standard deviation for the previously described methods. On average, our method consistently shows a significant improvement on the baseline in both the fully observable and partially observable cases. However, both methods seem to be relatively insensitive to increases in noise, as they create only a small increase in average error. Furthermore, unlike the synthetic data case, both methods are also relatively consistent in terms of error standard deviation.

Note that in both the full and partial state experiments, the noiseless-observation trajectory error is non-zero. We attribute these errors to model mismatch, referring to the differences between actual human decision-making processes and our recreation of human decision-making. Two likely dominant sources of mismatch are our use of an open-loop information structure—chosen for tractability over a more human closed-loop form—and the expressiveness of our cost structures. More complex cost structures, such as those encoded by neural networks, may also close the gap.



(a) Only position measurements (b) Full state measurements

Fig. 6. Bootstrapped estimates (solid lines) and standard errors for the average trajectory prediction error for the baseline method and our method for the InD intersection experiment. On average, our method produces trajectories with less error in settings with both (a) partial state observations and (b) full state observations.

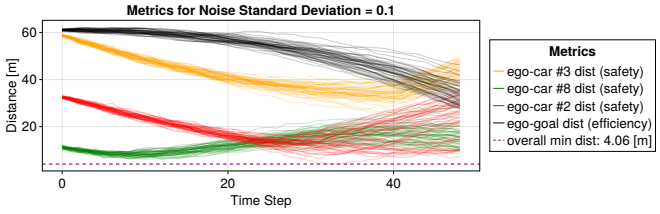


Fig. 7. Several metrics computed on 50 trajectories generated from the Waymax Monte Carlo simulation with an observation noise level of $0.1\,\mathrm{m}^2$. In particular, we focus on the distance between the ego robot and the other most important drivers in the simulation, as well as how far the ego robot is from their goal.

G. Monte Carlo Results - Waymax

Finally, we show how our method can be integrated into a receding horizon motion planner, and present several metrics related to safety and efficiency. In Figure 7, we show that a foresighted game-theoretic receding horizon planner maintains a distance of over $6\,\mathrm{m}$ from all cars at all times across all noise realizations, exhibiting a high level of safety. Meanwhile, the ego agent is reliably able to minimize the distance to its goal across all trials. These results demonstrate that our foresighted formulation yields safe and efficient behavior when employed in receding horizon robotics schemes. A representative trial of this simulation is shown in Figure 3.

VII. Conclusions

This paper formulates a noncooperative game that models potentially foresighted agents by associating each one with an unknown discount factor. Our approach rewrites the equilibrium conditions of the resulting game as an MiCP to leverage the directional differentiability of its equilibrium solutions with respect to unknown parameters. This construction allows us to solve an *inverse game* problem and identify hidden parameters—including agents' unknown discount factors—via a gradient descent procedure. By evaluating our method with noisy partial and full state observations on a simulated crosswalk environment and a real-world intersection and road, we demonstrate superior performance compared to an existing state-of-the-art method.

This work models foresighted behavior through the use of discount factors. In this vein, we find several avenues for further research interesting. First, because neural networks are universal function approximators, they can be used to approximate each player's cost function. This could potentially eliminate model mismatch and show greater improvement on our method. Second, humans often make decisions under the assumption they will have more information in the future, i.e., within feedback information structures. As noted in Section VI-F, future work may extend our open-loop formulation to better model this behavior, with Li *et al.* [19] offering a clear starting point for this direction. Third, humans may adapt how far ahead they plan based on context. On clear highways, they may focus on the near term, but in complex situations like merging or heavy traffic, they consider longer horizons. Future work may explore discounting schemes that vary with state and time.

References

[1] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, "Cooperative inverse reinforcement learning," *Advances in neural information processing systems*, vol. 29, 2016.

[2] V. Kuleshov and O. Schrijvers, "Inverse game theory: Learning utilities in succinct games," in *Web and Internet Economics: 11th International Conference, WINE 2015, Amsterdam, The Netherlands, December 9-12, 2015, Proceedings 11*, Springer, 2015, pp. 413–427.

[3] T. L. Molloy, J. I. Charaja, S. Hohmann, and T. Perez, *Inverse optimal control and inverse noncooperative dynamic game theory*. Springer, 2022.

[4] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory, 2nd Edition*. Society for Industrial and Applied Mathematics, 1998. DOI: 10.1137/1.9781611971132. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611971132. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611971132.

[5] A. W. Starr and Y.-C. Ho, "Nonzero-sum differential games," *Journal of optimization theory and applications*, vol. 3, pp. 184–206, 1969.

[6] J. C. Engwerda, "On the open-loop nash equilibrium in lq-games," *Journal of Economic Dynamics and Control*, vol. 22, no. 5, pp. 729–762, 1998.

[7] J. C. Engwerda, "Computational aspects of the open-loop nash equilibrium in linear quadratic games," *Journal of Economic Dynamics and Control*, vol. 22, no. 8-9, pp. 1487–1506, 1998.

[8] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a nash equilibrium," *Communications of the ACM*, vol. 52, no. 2, pp. 89–97, 2009.

[9] S. L. Cleac'h, M. Schwager, and Z. Manchester, "Algames: A fast solver for constrained dynamic games," *arXiv preprint arXiv:1910.09713*, 2019.

[10] B. Di and A. Lamperski, "First-order algorithms for constrained nonlinear dynamic games," *arXiv preprint arXiv:2001.01826*, 2020.

[11] E. L. Zhu and F. Borrelli, "A sequential quadratic programming approach to the solution of open-loop generalized nash equilibria," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 3211–3217.

[12] R. Spica, E. Cristofalo, Z. Wang, E. Montijano, and M. Schwager, "A real-time game theoretic planner for autonomous two-player drone racing," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1389–1403, 2020.

[13] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game theoretic planning for self-driving cars in competitive scenarios.," in *Robotics: Science and Systems*, 2019, pp. 1–9.

[14] S. Rothfuß, J. Inga, F. Köpf, M. Flad, and S. Hohmann, "Inverse optimal control for identification in non-cooperative differential games," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14 909–14 915, 2017, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2017.08.2538.

[15] C. Awasthi and A. Lamperski, "Inverse differential games with mixed inequality constraints," in *Proc. of the IEEE American Control Conference (ACC)*, 2020.

[16] J. Inga, E. Bischoff, F. Köpf, and S. Hohmann, "Inverse dynamic games based on maximum entropy inverse reinforcement learning," *arXiv preprint arXiv:1911.07503*, 2019.

[17] L. Peters, V. Rubies-Royo, C. J. Tomlin, *et al.*, "Online and offline learning of player objectives from partial observations in dynamic games," in *Intl. Journal of Robotics Research (IJRR)*, 2023. [Online]. Available: https://journals.sagepub.com/doi/pdf/10.1177/02783649231182453.

[18] X. Liu, L. Peters, and J. Alonso-Mora, "Learning to play trajectory games against opponents with unknown objectives," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4139–4146, 2023. DOI: 10.1109/LRA.2023.3280809.

[19] J. Li, C.-Y. Chiu, L. Peters, S. Sojoudi, C. Tomlin, and D. Fridovich-Keil, "Cost inference for feedback dynamic games from noisy partial state observations and incomplete trajectories," in *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2023, pp. 1062–1070.

[20] H. Hu, J. DeCastro, D. Gopinath, G. Rosman, N. E. Leonard, and J. F. Fisac, "Think deep and fast: Learning neural nonlinear opinion dynamics from inverse dynamic games for split-second interactions," *arXiv preprint arXiv:2406.09810*, 2024.

[21] L. Peters, "Accommodating intention uncertainty in general-sum games for human-robot interaction," 2020.

[22] S. Le Cleac'h, M. Schwager, and Z. Manchester, "LUCIDGames: Online unscented inverse dynamic games for adaptive trajectory prediction and planning," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 3, pp. 5485–5492, 2021.

[23] X. Liu, L. Peters, J. Alonso-Mora, U. Topcu, and D. Fridovich-Keil, *Auto-encoding bayesian inverse games*, 2024. arXiv: 2402.08902 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2402.08902.

[24] N. Mehr, M. Wang, M. Bhatt, and M. Schwager, "Maximum-entropy multi-agent dynamic games: Forward and inverse solutions," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1801–1815, 2023. DOI: 10.1109/TRO.2022.3232300.

[25] C. Diehl, T. Klosek, M. Krueger, N. Murzyn, T. Osterburg, and T. Bertram, "Energy-based potential games for joint motion forecasting and control," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2023.

[26] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, May 1995, ISSN: 1095-3787. DOI: 10.1103/physreve.51.4282. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.51.4282.

[27] C. Li, W. Lu, Q. Lin, L. Meng, H. Li, and B. Liang, *Navscope: Swarm robot cooperative perception and coordinated navigation*, 2025. arXiv: 2409.10049 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2409.10049.

[28] O. Khatib, "The potential field approach and operational space formulation in robot control," in *Adaptive and Learning Systems: Theory and Applications*, K. S. Narendra, Ed. Boston, MA: Springer US, 1986, pp. 367–377, ISBN: 978-1-4757-1895-9. DOI: 10.1007/978-1-4757-1895-9_26. [Online]. Available: https://doi.org/10.1007/978-1-4757-1895-9_26.

[29] P. Fiorini, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, pp. 760–, Jul. 1998. DOI: 10.1177/027836499801700706.

[30] J. Snape and D. Manocha, "Goal velocity obstacles for spatial navigation of multiple autonomous robots or virtual agents," *Autonomous Robots and Multi robot Systems, St. Paul, Minn*, pp. 1–17, 2013.

[31] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08)*, AAAI Press, 2008, pp. 1433–1438. [Online]. Available: https://cdn.aaai.org/AAAI/2008/AAAI08-227.pdf.

[32] B. D. Ziebart, N. Ratliff, G. Gallagher, *et al.*, "Planning-based prediction for pedestrians," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 3931–3936. DOI: 10.1109/IROS.2009.5354147.

[33] N. Deo and M. M. Trivedi, *Convolutional social pooling for vehicle trajectory prediction*, 2018. arXiv: 1805.06771 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1805.06771.

[34] D. A. Ridel, N. Deo, D. Wolf, and M. M. Trivedi, *Understanding pedestrian-vehicle interactions with vehicle mounted vision: An lstm model and empirical analysis*, 2019. arXiv: 1905.05350 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1905.05350.

[35] B. Ivanovic and M. Pavone, *The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs*, 2019. arXiv: 1810.05993 [cs.RO]. [Online]. Available: https://arxiv.org/abs/1810.05993.

[36] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, *Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data*, 2021. arXiv: 2001.03093 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2001.03093.

[37] C. H. Papadimitriou, "The complexity of finding nash equilibria," *Algorithmic game theory*, vol. 2, p. 30, 2007.

[38] E. V. Mazumdar, M. I. Jordan, and S. S. Sastry, "On finding local nash equilibria (and only local nash equilibria) in zero-sum games," *arXiv preprint arXiv:1901.00838*, 2019.

[39] E. Mazumdar, L. J. Ratliff, and S. S. Sastry, "On gradient-based learning in continuous games," *SIAM Journal on Mathematics of Data Science*, vol. 2, no. 1, pp. 103–131, 2020.

[40] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[41] L. J. Ratliff, S. A. Burden, and S. S. Sastry, "On the characterization of local nash equilibria in continuous games," *IEEE transactions on automatic control*, vol. 61, no. 8, pp. 2301–2307, 2016.

[42] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer Science & Business Media, 2007.

[43] S. P. Dirkse and M. C. Ferris, "The path solver: A non-monotone stabilization scheme for mixed complementarity problems," *Optimization methods and software*, vol. 5, no. 2, pp. 123–156, 1995.

[44] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion—an inverse optimal control approach," *Autonomous robots*, vol. 28, pp. 369–383, 2010.

[45] P. Englert, N. A. Vien, and M. Toussaint, "Inverse kkt: Learning cost functions of manipulation tasks from demonstrations," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1474–1488, 2017.

[46] L. Peters, V. Rubies-Royo, C. J. Tomlin, *et al.*, "Online and offline learning of player objectives from partial observations in dynamic games," *The International Journal of Robotics Research*, vol. 42, no. 10, pp. 917–937, 2023.

[47] A. L. Dontchev and R. T. Rockafellar, *Implicit functions and solution mappings*. Springer, 2009, vol. 543.

[48] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1929–1934. DOI: 10.1109/IV47402.2020.9304839.

[49] C. Gulino, J. Fu, W. Luo, *et al.*, "Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2023.