In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS 2025), Hangzhou, China, October 2025

Multi-Agent Inverse Reinforcement Learning in Real World Unstructured Pedestrian Crowds

Rohan Chandra[1], Haresh Karnan[2], Negar Mehr[3], Peter Stone[2,4], Joydeep Biswas[2]
[1]University of Virginia, [2]The University of Texas at Austin, [3]University of California, Berkeley, [4]Sony AI
Code and Data at **mairl-uva.github.io**

***Abstract*— Social robot navigation in crowded public spaces such as university campuses, restaurants, grocery stores, and hospitals, is an increasingly important area of research. One of the core strategies for achieving this goal is to understand humans' intent–underlying psychological factors that govern their motion–by learning how humans assign rewards to their actions, typically via inverse reinforcement learning (IRL). Despite significant progress in IRL, learning reward functions of multiple agents simultaneously in dense unstructured pedestrian crowds has remained intractable due to the nature of the tightly coupled social interactions that occur in these scenarios *e.g.* passing, intersections, swerving, weaving, etc. In this paper, we present a new multi-agent maximum entropy inverse reinforcement learning algorithm for real world unstructured pedestrian crowds. Key to our approach is a simple, but effective, mathematical trick which we name the so-called "*tractability-rationality trade-off*" trick that achieves tractability at the cost of a slight reduction in accuracy. We compare our approach to the classical single-agent MaxEnt IRL as well as state-of-the-art trajectory prediction methods on several datasets including the ETH, UCY, SCAND, JRDB, and a new dataset, called Speedway, collected at a busy intersection on a University campus focusing on dense, complex agent interactions. Our key findings show that, on the dense Speedway dataset, our approach ranks 1^{st} among top 7 baselines with $> 2\times$ improvement over single-agent IRL, and is competitive with state-of-the-art large transformer-based encoder-decoder models on sparser datasets such as ETH/UCY (ranks 3^{rd} among top 7 baselines).**

I. Introduction

Robot navigation in densely populated human environments has garnered significant attention, especially in scenarios involving tightly coupled social interactions among pedestrians, scooters, and vehicles. These scenarios [1] occur in both outdoor settings [2], [3], [4], [5] and indoor spaces such as restaurants, grocery stores, hospitals, and university campuses [6], [7], [8], [9], [10], [11]. Effective navigation in such dense and cluttered human environments is challenging and requires robots to be socially compliant which involves predicting and responding to human intentions [12], [13]. Many approaches have approached intent prediction *implicitly* via trajectory forecasting [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24]. While trajectory forecasting has yielded impressive results on sparse crowd datasets such as ETH, UCY, and JRDB, and structured traffic datasets such as the WAYMO Motion Forecasting Dataset [25], it is unclear whether they perform similarly well in more complex settings that include passing, weaving, yielding to others, etc.

Alternatively, some approaches have modeled intent *explicitly* by computing humans' reward functions via inverse reinforcement learning. Recently, Gonon and Billard [26] used maximum entropy IRL (inverse reinforcement learning) [27] for achieving socially compliant navigation in pedestrian crowds. However, inherently a single-agent approach, maximum entropy IRL (MaxEnt IRL) assumes a single objective function for all the agents, which works reasonably well in sparse crowds that lack many tightly coupled interactions, as shown in [26], but fails to accommodate humans' objectives in denser and more unstructured pedestrian crowds, as we show in our experiments in Section V. This limitation leads us to question: *Can MaxEnt IRL be adapted to learn effective robot navigation policies for real-world multi-agent unstructured pedestrian crowds?*

We address this question using a multi-agent IRL approach, for which numerous solutions already exist in the dynamic game theory and reinforcement learning literature [28], [29], [30], [31], [32], [33], [34], [35], [36]. Key to making these methods work is the critical assumption of exactly known dynamics/system models of all the agents in the environment, which, however, rarely holds in practice. To relax this assumption, one may consider approximating the dynamics with, say, a simpler model (*e.g.* constant velocity or constant acceleration models). The downside of such an approximation is that the optimization-based planner does not converge.

Main contribution: In this work, we show that multi-agent MaxEnt IRL provides accurate predictions of pedestrians' motion in real world unstructured crowds, with only few training trajectories. Our key insight is a simple, but effective, mathematical trick which we name the so-called "*tractability-rationality trade-off*" trick that relaxes the assumption of known system dynamics tractably at the cost of a slight reduction in accuracy. Bounded rationality (assumption of agents exhibiting some randomness in their decision-making) is typically, characterized by the covariance matrix of the probabilistic policy. At a high level, the trick uses matrix conditioning (a textbook technique in numerical linear algebra) to improve the condition number of the covariance matrix, thus ensuring tractability, but at the cost of reducing uncertainty, thereby reducing bounded rationality. Notably, this trade-off not only makes multi-agent IRL tractable in unstructured settings but also paves the way for new research directions, such as computing exact bounds for this trade-off.

We use off-the-shelf dynamic game theory solvers to roll out trajectories that maximize the learned reward functions. We evaluate this new approach using a dataset collected at a busy intersection on a University campus comprising

This work has taken place in the Autonomous Mobile Robotics Laboratory (NSF CAREER-2046955). A portion of this work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (FAIN-2019844, NRT-2125858), ONR (N00014-24-1-2550), ARO (FAIN W911NF-17-2-0181, W911NF-23-2-0004, W911NF-25-1-0065), DARPA (Cooperative Agreement HR00112520004 on Ad Hoc Teamwork) Lockheed Martin, and UT Austin's Good Systems grand challenge. Peter Stone serves as the Chief Scientist of Sony AI and receives financial compensation for that role. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research. Negar Mehr is supported by NSF ECCS-2438314 CAREER and CCF-2423134.

of pedestrians, scooters, bicycles, particularly focusing on tightly coupled interactions such as when agents walk towards each other, walk past each other, squeeze past each other, etc. We compare our approach to the classical single-agent MaxEnt IRL as well as state-of-the-art trajectory prediction methods on several datasets including the ETH, UCY, JRDB, and a new dataset collected at a busy intersection on a University campus focusing on complex agent interaction.

In the remainder of this paper, we discuss related work in Section II, formulate the problem in Section III, present our technical approach in Section IV, discuss the experiments and results in Section V, and conclude in Section VI.

II. Related Work

A. Inverse Reinforcement Learning

Kalman initially explored the inference of an agent's cost function within the context of inverse optimal control for linear quadratic systems with a linear control law [37]. Subsequent works [38], [39] considered learning the agent's cost, assuming demonstrations adhered to optimality conditions. However, multiple cost functions can rationalize the expert's behavior, leading to inherent ambiguities. MaxEnt IRL [27] addresses this by incorporating the principle of maximum entropy, ensuring that the derived cost function is both unique and maximally uncertain.

Multi-agent IRL: While previous approaches focused on learning in the single-agent regime, multi-agent IRL has been extensively studied in various settings. Initial efforts focused on discrete state and action spaces [28], while later research extended to general-sum games with discrete spaces [29]. Adversarial machine learning techniques were employed for high-dimensional state and action spaces [30], and generative adversarial imitation learning was extended to the multi-agent setting [31]. Additionally, the problem has been approached as an estimation task [32], [33], and linear quadratic games were considered with known equilibrium strategies [34], [35]. The maximum entropy IRL [36] framework was introduced to accommodate bounded rationality and noisy human demonstrations.

B. Trajectory Prediction

We focus on deep learning-based trajectory and intent prediction which dates back ten years to the seminal SocialLSTM work [40]. TraPHic and RobustTP [41], [42] use an LSTM-CNN framework to predict trajectories in dense and complex environments. TNT [43] uses target prediction, motion estimation, and ranking-based trajectory selection to predict future trajectories. DESIRE [44] uses sample generation and trajectory ranking for trajectory prediction. PRECOG [45] combines conditioned trajectory forecasting with planning objectives for agents. Yoo et al. [46] uses a Seq2Seq framework to encode agents' observations over neighboring agents as their social context for trajectory forecasting and decision-making. Cao et al. [47] uses temporal smoothness in attention modeling for interactions and a sequential model for trajectory prediction. However, a major limitation of this class of methods is that they employ behavior cloning to try and mimic the data, without inherently capturing humans' internal reward functions.

C. Intent-aware Multi-agent Reinforcement Learning

Related to our task, intent-aware multi-agent reinforcement learning [48] estimates an intrinsic value that represents opponents' intentions for communication [49] or decision-making. Many intent inference modules are based on Theory of Mind (ToM) [50] reasoning or social attention-related mechanisms [51], [52]. Wu et al. [53] uses ToM reasoning over opponents' reward functions from their historical behaviors in performing multi-agent inverse reinforcement learning (MAIRL). Chandra et al. [9] uses game theory ideas to reason about other agents' incentives and help decentralized planning among strategic agents. However, many prior works oversimplify the intent inference and make some prior assumptions about the content of intent.

D. Opponent Modeling

Opponent modeling [54] has been studied in multi-agent reinforcement learning and usually deploys various inference mechanisms to understand and predict other agents' policies. Opponent modeling could be done by either estimating others' actions and safety via Gaussian Process [55] or by generating embeddings representing opponents' observations and actions [56]. Inferring opponents' policies helps to interpret peer agents' actions [57] and makes agents more adaptive when encountering new partners [58]. Notably, many works [59], [60] reveal the phenomenon whereby ego agents' policies also influence opponents' policies. To track the dynamic variation of opponents' strategies made by an ego agent's influence, [61], [62] propose the latent representation to model opponents' strategies and influence based on their findings on the underlying structure in agents' strategy space. Jaques et al. [63] provides a causal influence mechanism over opponents' actions and defines an influential reward over actions with high influence over others' policies. Kim et al. [64] proposes an optimization objective that accounts for the long-term impact of ego agents' behavior in opponent modeling. A considerable limitation of many current methodologies is the underlying assumption that agents continually interact with a consistent set of opponents across episodes.

Placing our work in the literature: Our work directly builds on top of the multi-agent IRL literature. It is complementary to the trajectory prediction, multi-agent reinforcement learning, and opponent modeling literature, which implicitly model other agent's objectives, whereas our approach explicitly models humans' intent via reward functions.

III. Background and Problem Formulation

We define a game, $G \coloneqq \langle k, \mathcal{X}, \mathcal{T}, \{\mathcal{U}^i\}, \{\mathcal{J}^i\}\rangle$, where k denotes the number of agents. Hereafter, i will refer to the index of an agent and appear as a superscript whereas t will refer to the current time-step and appear as a subscript. The general state space $\mathcal{X}$ (*e.g.* SE(2), SE(3), etc.) is continuous; the i^{th} agent at time t has a state $x_t^i \in \mathcal{X}^i$. Over a finite horizon T, each agent starts from $x_0^i \in \mathcal{X}^i$ and reaches a goal state $x_T^i \in \mathcal{X}^i$. An agent's transition function is a mapping, $\mathcal{T}^i : \mathcal{X} \times \mathcal{U}^i \longrightarrow \mathcal{X}$, where $u_t^i \in \mathcal{U}^i$ is the continuous control input for agent i. A discrete trajectory for agent i is specified by the sequence $\Gamma^i = \left(x_0^i, u_0^i, x_1^i, u_1^i, \ldots, x_{T-1}^i, u_{T-1}^i, x_T^i\right)$.

Each agent has a parameterized running cost[1] $\mathcal{J}^i : \mathcal{X} \times \mathcal{U} \longrightarrow \mathbb{R}$ where $\mathcal{U} = \mathcal{U}^1 \times \mathcal{U}^2 \times, \ldots, \times \mathcal{U}^k$. The cost $\mathcal{J}^i$ acts on joint state x_t and control $u_t \in U$ at each time step measuring the distances between agents, deviation

[1]Note that broadly speaking, reward maximization is equivalent to cost minimization, which is what we model in this work.

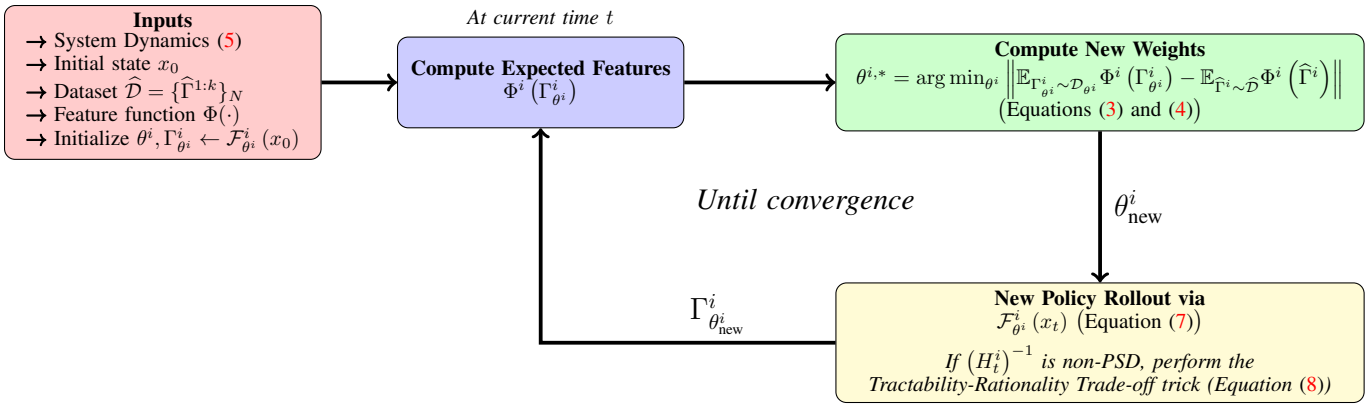


Fig. 1: **Flow diagram of the multi-agent inverse reinforcement learning algorithm.** The algorithm begins by taking as input the system dynamics, initial state, dataset, feature function, and initial parameters. At each time step, it computes the expected features based on the current policy. These features are used to update the weights by minimizing the difference between the expected and ground truth features. The updated weights are then used to update the game-theoretic policy and roll out a new set of trajectories. If the Hessian matrix is non-positive definite, the "Tractability-Rationality trade-off" trick is applied to ensure tractability in unstructured environments.

from the reference path, and the change in the control input. Each agent i follows a stochastic decentralized policy $\mathcal{F}^i_{\theta^i} : \mathcal{X} \longrightarrow \mathcal{U}^i$, which captures the probability of agent i selecting action u^i_t at time t, given that the system is in state x_t. The parameters for the policy $\mathcal{F}^i$ are represented by θ^i, which may correspond to weights of a neural network or coefficients in a weighted linear function. We assume that we are provided with a dataset $\widehat{\mathcal{D}} = \{\widehat{\Gamma}^{1:k}\}$ that consists of expert demonstration trajectories, $\widehat{\Gamma}^i$ for each agent i. Further, we let $\mathcal{D}_{\theta^i} = \{\Gamma^{1:k}_{\theta^i}\}$ denote the dataset of trajectories generated by the policy $\mathcal{F}^i_{\theta^i}$ parameterized by θ^i.

A. Background: MaxEnt IRL

As background, we first briefly review the key details of (single-agent) MaxEnt IRL [26], [27]. As there is only one agent ($k = 1$), we omit the superscript i for our discussion on single-agent MaxEnt IRL. Given expert trajectories and assuming that the human is boundedly rational, the probability of a trajectory Γ under the MaxEnt IRL framework is defined as:

$$P(\Gamma) \propto \exp\left(\sum_{t=0}^{T} \mathcal{J}(x_t, u_t)\right)$$

where $\mathcal{J}(x_t, u_t)$ represents the cost at time t in joint state x_t and taking action u_t.

Let $\Phi(\cdot)$ be a feature function that, given a trajectory, returns a vector of features $[\phi_1, \phi_2, \ldots, \phi_m]^\top$, averaged across the length of the trajectory, where m refers to the number of distinct features. A simple example of a feature function is the weighted linear sum $\Phi(\Gamma_\theta) = \sum_{t=0}^{T-1} \theta^\top \mathcal{J}(x_t, u_t)$. The objective in MaxEnt IRL is to adjust the weights of the cost function such that the expected features under the derived policy match the features of the expert, while also maximizing the entropy of the policy. The cost function is iteratively adjusted, typically using gradient-based methods, to ensure that the expected feature counts from the policy induced by the current reward function align with those from the expert demonstrations. To ease notation, let $\Psi\left(\Gamma_\theta, \widehat{\Gamma}\right) = \Phi(\Gamma_\theta) - \Phi\left(\widehat{\Gamma}\right)$ be a generalized discrepancy function. More formally, MaxEnt IRL finds θ^* that minimizes

$$\left\| \mathbb{E}_{\widehat{\Gamma}, \Gamma_\theta \sim \widehat{\mathcal{D}}, \mathcal{D}_\theta} \Psi\left(\Gamma_\theta, \widehat{\Gamma}\right) \right\| \tag{1}$$

There are several reasons why the MaxEnt IRL has been successful. First, the maximum entropy principle ensures a unique solution to the reward recovery problem, mitigating ambiguities inherent in traditional IRL. Further, resultant policies are inherently stochastic, capturing a broader range of behaviors and offering robustness in dynamic environments. Finally, MaxEnt IRL can generalize better to unseen states, making it particularly suitable for complex environments like pedestrian crowds [26].

B. Problem Formulation

We can generalize (1) to the multi-agent setting. Our problem statement is then,

Problem 1. *Find $\theta^{i,*}$ for $i \in [1, k]$ that minimizes*

$$\left\| \mathbb{E}_{\widehat{\Gamma}^i, \Gamma^i_{\theta^i} \sim \widehat{\mathcal{D}}, \mathcal{D}_{\theta^i}} \Psi^i\left(\Gamma^i_{\theta^i}, \widehat{\Gamma}^i\right) \right\| \tag{2}$$

Note that we slightly abuse notation and ignore the terminal cost at $t = T$.

IV. Multi-agent MaxEnt IRL for Unstructued Pedestrian Crowds

We visually summarize the algorithm in Figure 1. This section begins with a brief motivation for Multi-agent MaxEnt IRL, followed by describing the algorithm in detail, which includes the Tractability-Rationality trade-off trick which is depicted in the yellow box in Figure 1.

In MaxEnt IRL [26], the goal is to learn a cost function that applies either to a single agent or is designed to collectively model a group of individuals. However, individuals in groups behave differently according to their nature. For instance, some individuals might be more risk averse than others, and therefore, are more likely to have a higher reward for collision avoidance. Others might place greater importance on reaching their goals faster and place greater weight for the cost of distance to goal. In [36], Mehr et al.

introduced the following multi-agent formulation of MaxEnt IRL for unstructured pedestrian crowds.

Given a feature function $\Phi^i\left(\Gamma^i_{\theta^i}\right)$, that computes the features for agent i with respect to weights θ^i and costs $\mathcal{J}^i\left(x_t, \mathbf{u}_t\right)$, solving Equation (2) reduces to finding $\theta^{i,*}$ for every i such that,

$$\theta^{i,*} = \arg\min_{\theta^i} \left\| \mathbb{E}_{\Gamma^i_{\theta^i} \sim \mathcal{D}_{\theta^i}} \Phi^i\left(\Gamma^i_{\theta^i}\right) - \mathbb{E}_{\widehat{\Gamma}^i \sim \widehat{\mathcal{D}}} \Phi^i\left(\widehat{\Gamma}^i\right) \right\| \quad (3)$$

Equation (3) can be efficiently solved via block coordinate descent [65] where we treat θ as a concatenation of each θ^i, where i then refers to a coordinate. An update to the cost parameters at the current iteration, θ^i, of agent i is of the following form,

$$\theta^i_{\text{new}} \leftarrow \theta^i - \beta \left(\mathbb{E}_{\widehat{\Gamma}^i, \Gamma^i_{\theta^i} \sim \widehat{\mathcal{D}}, \mathcal{D}_{\theta^i}} \Phi^i\left(\Gamma^i_{\theta^i}\right) - \Phi^i\left(\widehat{\Gamma}^i\right) \right) \quad (4)$$

where we compute the features for trajectories generated from the new set of cost parameters $\left(\theta^i_{\text{new}}\right)$ and proceed to update the cost parameters for the next agent. We iterate over each agent i and repeat this process until convergence. At each iteration of the update process, each agent i uses a dynamic game solver [66] as its parameterized stochastic policy $\mathcal{F}^i_{\theta^i}\left(x_t\right)$ to roll out M trajectories $\{\Gamma^i\}_{1:M}$ corresponding to the updated cost parameters $\left(\theta^i_{\text{new}}\right)$. In the case of multi-agent IRL, the policy $\mathcal{F}^i_{\theta^i}$ must be obtained at each iteration using the current parameters.

Recently, Mehr et al. [36] showed that when the cost function for an agent, $\mathcal{J}^i$, assumes a quadratic form with linear system dynamics, then the set of policies, $\mathcal{F}^i_{\theta^i}\left(x_t\right)$, for k boundedly rational agents can be obtained via backwards recursion through a set of Ricatti equations [33]. In unstructured crowds, however, the pedestrian dynamics are unknown. Recent IRL-based studies found that constant velocity [67] and constant acceleration [26] motion models achieve a good enough approximation for pedestrian trajectory prediction benchmarks. Inspired by this result, we approximate the unknown systems dynamics with constant velocity dynamics,

$$\begin{bmatrix} \dot{p}^{i,x} \\ \dot{p}^{i,y} \\ \dot{\psi}^i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v^{i,x} \\ v^{i,y} \\ 0 \end{bmatrix}, \quad (5)$$

where $x^i_t = [p^{i,x}, p^{i,y}, \psi^i]^\top$ represents the 2D position and heading angle of agent i. Furthermore, as we do not know the cost functions $\mathcal{J}^i$, we generate a quadratic approximation via Taylor series expansion around the cumulative joint state value $\bar{x}_t = x_{1:t}$ to approximate $\mathcal{J}^i$ as

$$\widetilde{\mathcal{J}}^i\left(\bar{x} + \delta_{x_t}\right) \approx \widetilde{\mathcal{J}}^i\left(\bar{x}\right) + \frac{1}{2}\delta^\top_{x_t} H^i_t \delta_{x_t} + {l^i_t}^\top \delta_{x_t} \quad (6)$$

where $\delta_{x_t} = \bar{x}_t - x_t$ is the difference between the cumulative joint state and the joint state at the current time step. Note that this formulation only considers nonlinear costs on state variables and the dependence on agents' actions is quadratic. A similar approximation may be derived for a more general case where the cost function is nonlinear in control as well. The matrix H^i_t and vector l^i_t correspond to the Hessian and gradient of the approximate cost function with respect to x_t. Each policy, $\mathcal{F}^i_{\theta^i}\left(x_t\right)$, is a Gaussian distribution where the mean and covariance parameters are functions of H^i_t:

$$\begin{aligned} \mathcal{F}^i_{\theta^i}\left(x_t\right) &= \mathcal{N}\left(\mu^i_t, \Sigma^i_t\right) \\ \mu^i_t, \Sigma^i_t &\leftarrow f\left(H^i_t\right)^{-1} \end{aligned} \quad (7)$$

where $f(\cdot)$ is a linear matrix transformation function (c.f. [36] for more details of $f(\cdot)$).

Intractability of Equation (6)***:*** Our assumption of constant velocity dynamics in Equation (5), combined with the fact that humans typically walk along linear paths at near identical speeds on average, results in a linear approximation of the cost function in Equation (6). Under constant velocity dynamics and linear motion, the state evolution is linear. The deviations δ_{x_t} from the trajectory $\bar{x}_t$ remain small and linear. Consequently, the second-order derivatives i.e., the entries of $\left(H^i_t\right)$, become very small since there is minimal curvature in the cost function along linear trajectories. Mathematically, the curvature of the cost function is captured by the eigenvalues of the Hessian H^i_t. If the cost function has little curvature (as it does under linear motion), these eigenvalues are close to zero, resulting in a non positive semi-definite (PSD) matrix.

Tractability-Rationality Trade-off Trick*:* To achieve computational tractability, we relax the assumption of bounded rationality, which is typically represented by the stochastic nature of the agent's policy $\mathcal{F}^i_{\theta^i}(x_t)$, characterized by the covariance matrix Σ^i_t. The bounded rationality assumption implies that agents exhibit some randomness in their decision-making, often modeled as a probabilistic distribution over actions.

However, performing computations directly with Σ^i_t can lead to intractable solutions due to its dependence on H^i_t which may be non-PSD. To address this, we modify the covariance matrix by adding a function of its diagonal elements:

$$\widetilde{\Sigma}^i_t = \Sigma^i_t + g\left(\text{diag}(\Sigma^i_t)\right) I, \quad (8)$$

where $g\left(\text{diag}(\Sigma^i_t)\right)$ produces a vector that adjusts the diagonal elements of Σ^i_t. This operation ensures that $\widetilde{\Sigma}^i_t$ is a positive semi-definite (PSD) matrix, thus preserving the stochastic nature of the policy while making the problem more tractable. The diagonal elements of Σ^i_t, denoted as $\sigma^{ij}, i = j$, quantify the uncertainty or randomness in the action that agent i takes toward agent j. By applying the function g to these diagonal elements, we essentially decrease the variance (uncertainty) of the agent's actions. This modification shifts towards a more tractable model at the cost of potentially capturing a less accurate representation of agent behaviors, as seen in our evaluation.

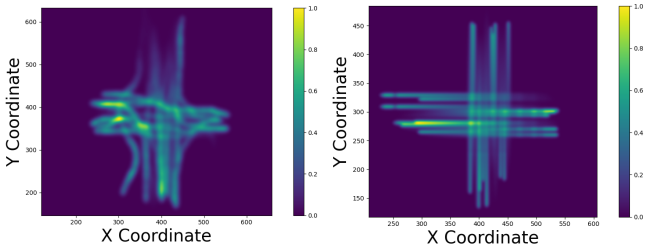
Remark: There are several new open problems in this research direction. First, in this work, we heuristically select $g(\cdot)$ from multiple viable candidates $\left(g\left(\text{diag}\left(\Sigma^i_t\right)\right) I = I\right)$ is a trivial choice which enforces tractability but eliminates rationality). Establishing a suitable class of functions for $g(\cdot)$ remains a topic for future work. Another interesting open problem is to establish a bound on the change in the agents' behavior due to the adjustment of the covariance matrix, $\widetilde{\Sigma}^i_t$, and its subsequent impact on the mixed-Nash equilibrium.

Method	ETH	Hotel	Univ	Zara1	Zara2	Speedway	Average
Social-GAN [14]	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84	-	0.58/1.18
SoPhie [15]	0.70/1.43	0.76/1.67	0.54/1.24	0.30/0.63	0.38/0.78	-	0.54/1.15
CGNS [68]	0.62/1.40	0.70/0.93	0.48/1.22	0.32/0.59	0.35/0.71	-	0.49/0.97
S-BiGAT* [16]	0.69/1.29	0.49/1.01	0.55/1.32	0.30/0.62	0.36/0.75	-	0.48/1.00
MATF [69]	1.01/1.75	0.43/0.80	0.44/0.91	0.26/0.45	0.26/0.57	-	0.48/0.90
Next* [70]	0.73/1.65	0.30/0.59	0.60/1.27	0.38/0.81	0.31/0.68	-	0.46/1.00
SR-LSTM [71]	0.63/1.25	0.37/0.74	0.51/1.10	0.41/0.90	0.32/0.70	-	0.45/0.94
STGAT [72]	0.65/1.12	0.35/0.66	0.52/1.10	0.34/0.69	0.29/0.60	-	0.43/0.83
GOAL-GAN* [73]	0.59/1.18	0.19/0.35	0.60/1.19	0.43/0.87	0.32/0.65	-	0.43/0.85
SGCN [74]	0.63/1.03	0.32/0.55	0.37/0.70	0.29/0.53	0.25/0.45	-	0.37/0.65
MANTRA [75]	0.48/0.88	0.17/0.33	0.37/0.81	0.27/0.58	0.30/0.67	-	0.32/0.65
Transformer [17]	0.61/1.12	0.18/0.30	0.35/0.65	0.22/0.38	0.17/0.32	-	0.31/0.55
SMEMO [18]	0.39/0.59	0.14/0.20	0.23/0.41	0.19/0.32	0.15/0.26	-	0.22/0.35
Introvert [76]	0.42/0.70	0.11/0.17	0.20/0.32	0.16/0.27	0.16/0.25	-	0.21/0.34
PCCSNet [19]	0.28/0.54	0.11/0.19	0.29/0.60	0.21/0.44	0.15/0.34	-	0.21/0.42
MaxEnt IRL [26]	0.76 / 1.48	0.68 / 1.74	0.51 / 1.21	0.37 / 0.69	0.29 / 0.75	0.40/0.80	0.50/1.11
PECNet [77]	0.54/0.87	0.18/0.24	0.35/0.60	0.22/0.39	0.17/0.30	0.35/0.75	0.30/0.51
SceneTransformer [20]	0.50 / 0.76	0.14 / 0.20	0.29 / 0.42	0.22 / 0.36	0.16 / 0.27	0.33/0.72	0.27/0.46
AgentFormer [21]	0.45/0.75	0.14/0.22	0.25/0.45	0.18/0.30	0.14/0.24	0.31/0.76	0.24/0.44
HST [22]	0.41 / 0.73	0.10 / 0.14	0.24 / 0.44	0.17 / 0.30	0.14 / 0.24	0.35/0.68	0.24/0.42
LB-EBM [78]	0.30/0.52	0.13/0.20	0.27/0.52	0.20/0.37	0.15/0.29	0.32/0.81	0.23/0.45
Trajectron++ [23]	0.39/0.83	0.12/0.19	0.22/0.43	0.17/0.32	0.12/0.25	0.29/0.75	0.22/0.47
Expert-Goals [24]	0.37/0.65	0.11/0.15	0.20/0.44	0.15/0.31	0.12/0.25	0.29/0.69	**0.21/0.41**
MAIRL	0.39 / 0.71	0.13 / 0.18	0.23 / 0.32	0.19 / 0.29	0.15 / 0.28	**0.26/0.60**	0.23 / 0.39

TABLE I: Quantitative results (ADE/FDE) of our proposed method compared to state-of-the-art baselines on five benchmark datasets. All results are in meters. Lower is better

Dataset	PecNET [79]	SMEMO [18]	OpenTraj [80]	JRDB_Traj [81]	MaxEnt IRL [26]	Social-pose [82]	HST [22]	MAIRL
JRDB [83]	3.939	2.671	3.498	2.646	4.249	2.558	**2.506**	2.640
SCAND [6]	0.55/1.20	0.50/1.10	0.44/0.90	0.36/0.81	0.42/0.85	-	-	**0.23/0.67**

TABLE II: EFE (End-to-end Forecasting Error) results on the JRDB and SCAND dataset. All results are in meters/ Lower is better.



(a) Our dataset (b) Waymo/INTERACTION

Fig. 2: Visualizing and comparing the entropy of trajectories in the Speedway dataset (1.061 bits) with the entropy of trajectories in the Waymo and INTERACTION datasets (0.336 and 0.475 bits, respectively). We observe that trajectories in the Speedway dataset are denser, more unstructured, and have a higher entropy.

V. Evaluation

We compare our multi-agent MaxEnt IRL approach with single-agent MaxEnt IRL and state-of-the-art trajectory planning and prediction approaches and aim to understand how our method compares to these baselines on popular autonomous driving and pedestrian crowd datasets.

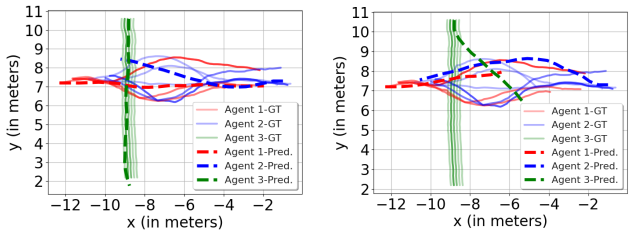
A. Data Pre-processing

We collected an unstructured real world pedestrian crowd dataset at one of the busiest intersections on a University campus using a BlueCity[2] Velodyne LiDAR mounted at the intersection. Our dataset consists of more tightly coupled agent interactions compared to existing motion planning datasets [6], [83], [25].

The BlueCity LiDAR is equipped with in-built object detection and tracking modules [84]. The LiDAR continuously streams data as frames where each frame consists of the timestamp and a sequence of frame objects corresponding to agents observed by the LiDAR. Each frame object consists of a unique id that is persistent in all the frames in which the agent appears, location of the agent in meter with respect to its distance to the sensor, width and length of the bounding box detected for the agent, the angle of the detected bounding box in radian, the class type of the agent such as 'pedestrian' or 'car', the speed of the agents, and the accuracy of the detection. We recorded trajectories consisting of the id of the agents, their locations, speed, and orientation during busy moments (such as in between classes).

We processed the data by removing agents that were either at a standstill or appeared at the edges of the LiDAR's field of view. Furthermore, we clipped each trajectory within a uniform spatio-temporal range ($[-20m, +20m]$ in the x axis and $[-10m, 15m]$ in the y axis). After processing the raw trajectories, we ended up with 20 trajectories of pedestrians with 5 trajectories each going from east to west (denoted as 'W'), west to east (denoted as 'E'), north to south (denoted as 'S'), south to north (denoted as 'N'). We created 4 different categories, 'W-E-S','W-E-N','S-N-W','S-N-E', with $5^3 = 125$ trajectories each for a total of 500 trajectories. Each trajectory is finally processed in the form of a $T \times 4k$ array where T corresponds to the number of frames in that interaction and $4k$ corresponds to

[2]https://ouster.com/products/software/blue-city



(a) Multi-agent Max-Ent IRL (b) Max-Ent IRL [27]

Fig. 3: **Qualitative comparison**–Each color represents a pedestrian. Solid faded lines represent demonstration trajectories and dashed lines represent trajectories generated from the learned policies. The directions of movement for the pedestrians are north to south (↓), east to west (←) and west to east (→). 'GT' and 'Pred' refer to ground truth and predicted trajectories. We inspect how closely the predicted trajectories align with the ground truth distribution.

the dimension of the joint state, that is, $x_t = \left[p^i, v^i, \phi^i\right]^\top$ for $i \in [1, k], p^i \in \mathbb{R}^2, v^i \in \mathbb{R}^1, \phi^i \in \mathbb{S}^1$. The final processed trajectories in the speedway dataset form the demonstration trajectories, denoted as $\widehat{\mathcal{D}}$. Features are computed based on distance to the goal position, distance to other agents, and control effort.

B. Baselines and Evaluation Metric

We compare our approach, which we call MAIRL (multi-agent IRL), with state-of-the-art trajectory prediction methods that have consistently performed well on leading benchmarks including the JRDB and ETH/UCY. To compare MAIRL with these methods[3] on the Speedway dataset, we directly use the opensource software for these baselines and train them on our dataset to present a direct evaluation. We set the learning rate to 3×10^{-4} and perform mini-batch training with a batch size of 64. A $60 - 40$ cross-validation split is used for evaluation. To prevent overfitting, we apply a dropout rate of 0.2. Training utilizes the AdamW optimizer with the Mean Squared Error (MSE) loss function. All training was performed on an NVIDIA GeForce RTX 2080 GPU machine and the average time to train the baselines was in the order of minutes.

Following standard evaluation procedures followed by the literature [26], [36], we measure the average displacement error (ADE) and the final displacement error (FDE) between the demonstration trajectories and the trajectories generated via the policy $\mathcal{F}^i_{\theta^i}$.

C. Results and Discussion

Table I compares (in order of better performance) the ADE/FDE (lower is better) of MAIRL with state-of-the-art trajectory prediction and planning methods on the well-known ETH/UCY datasets as well as on Speedway, in meters. The horizontal rule in the middle of the table divides the methods between ones that could be tested on the Speedway dataset versus ones that could not[3]. The results show that our MAIRL outperforms the top methods on the Speedway dataset, and is comparable on the ETH/UCY datasets, only performing worse than Expert-Goals [24] and Trajectron++ [23], two well-adopted methods for deep learning-based trajectory prediction. Table II additionally

[3]Note that for comparison on the Speedway dataset, baselines were selected by identifying state-of-the-art methods that (i) were compatible with our input data format and (ii) had open source implementations.

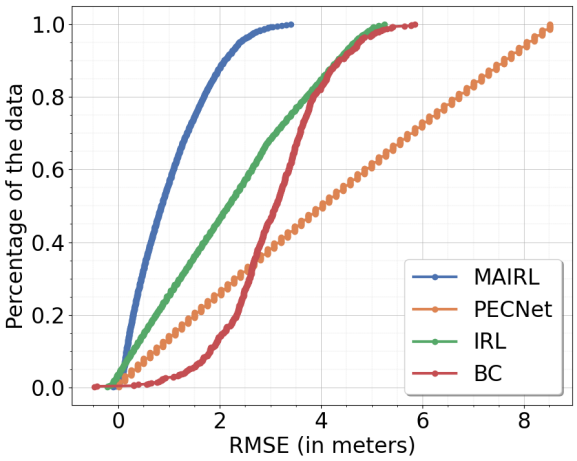


Fig. 4: Cumulative RMSE distribution demonstrates the percentage of trajectories below a certain RMSE threshold; steeper curves indicate a more effective learner.

compares MAIRL on the JRDB and SCAND datasets based on the End-to-End Forecasting Error (EFE), a specific metric used by the JRDB benchmakr, but is similar to the ADE/FDE metrics, again, in meters. MAIRL performs the best on SCAND, and comes in 3rd on JRDB, behind Social-Pose [82] and HST [22], two methods that additionally take into account the skeleton pose of the pedestrians. In both Tables I and II, note that we also compare MAIRL with single agent MaxEnt IRL, which performs considerably worse.

Figure 3 visually illustrates the comparison between multi- and singe-agent MaxEnt IRL, where solid faded lines represent the ground truth demonstration trajectories and dashed lines represent the predicted trajectories. The visualization corresponds to the intersection data comprising of 3 agents, each representing a different color. The directions of movement for the pedestrians are north to south (↓), east to west (←) and west to east (→). We inspect how closely the predicted trajectories are aligned with the ground truth distribution. Due to the single-agent nature of MaxEnt IRL, it aligns the predicted trajectories with the ground truth demonstrations for at most 1 agent (the blue agent), and fails for the green (extreme deviation) and the red (not reached goal) agents. Note that even though MAIRL outperforms all three baselines in terms of aligning with the ground truth demonstrations, it still remains imperfect in its own alignment due to the reduction in accuracy resulting from the Tractability-Rationality Trade-off, as discussed in Section III.

In Figure 4, we plot a cumulative distribution frequency (CDF) that measures the percentage of trajectories that fall below a certain error threshold; steeper curves imply a more effective algorithm as it indicates that that the algorithm was able to generate more low-ADE trajectories. The CDF plot is useful because the ADE/FDE metrics alone may not convey the full picture. For example, a baseline could yield a poor ADE, and be declared as weak algorithm, but a low ADE could result from one or two outlier trajectories with a high ADE, skewing the average against its favor.

VI. Conclusion and Open Questions

We show that multi-agent MaxEnt inverse reinforcement learning provides accurate predictions of pedestrians' motion

in real world unstructured pedestrian crowds with only few training trajectories. Our key insight was a mathematical trick that enabled tractability of multi-agent MaxEnt IRL at the cost of a slight reduction in accuracy. We used a dynamic game theoretic solver to compute optimal policies that maximize the reward functions learned via our approach. We evaluated this new approach using a dataset collected at a busy intersection on a University campus and showed that despite the accuracy trade-off, our approach outperforms state-of-the-art imitation learning baselines such as single-agent MaxEnt IRL, behavior cloning, and generative modeling. Our findings highlighted that our approach yields the lowest error between the policy-generated trajectories. Future work includes finding ways to solve Equation (7) in unstructured pedestrian crowds without relaxing bounded rationality. Another open problem is to compute an exact bound for the tractability-rationality tradeoff trick (Section IV). In addition, incorporating external factors like social contexts and the environment in modeling pedestrian behaviors can yield holistic and precise predictions, providing a comprehensive view of motion dynamics in various environments.

REFERENCES

[1] R. Chandra, V. Zinage, E. Bakolas, P. Stone, and J. Biswas, "Deadlock-free, safe, and decentralized multi-robot navigation in social mini-games via discrete-time control barrier functions," 2024.

[2] R. Chandra, M. Wang, M. Schwager, and D. Manocha, "Game-theoretic planning for autonomous driving among risk-aware human drivers," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2876–2883, 2022.

[3] R. Chandra and D. Manocha, "Gameplan: Game-theoretic multi-agent planning with human drivers at intersections, roundabouts, and merging," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2676–2683, 2022.

[4] N. Suriyarachchi, R. Chandra, J. S. Baras, and D. Manocha, "Gameopt: Optimal real-time multi-agent planning and control at dynamic intersections," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2599–2606, IEEE doi - 10.1109/ITSC55140.2022.9921968, 2022.

[5] A. Mavrogiannis, R. Chandra, and D. Manocha, "B-gap: Behavior-rich simulation and navigation for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4718–4725, 2022.

[6] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone, "Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11807–11814, 2022.

[7] A. H. Raj, Z. Hu, H. Karnan, R. Chandra, A. Payandeh, L. Mao, P. Stone, J. Biswas, and X. Xiao, "Targeted learning: A hybrid approach to social robot navigation," 2023.

[8] Z. Sprague, R. Chandra, J. Holtz, and J. Biswas, "Socialgym 2.0: Simulator for multi-agent social robot navigation in shared human spaces," *arXiv preprint arXiv:2303.05584*, 2023.

[9] R. Chandra, R. Maligi, A. Anantula, and J. Biswas, "Socialmapf: Optimal and efficient multi-agent path finding with strategic agents for social navigation," *IEEE Robotics and Automation Letters*, 2023.

[10] R. Chandra, R. Menon, Z. Sprague, A. Anantula, and J. Biswas, "Decentralized social navigation with non-cooperative robots via bi-level optimization," *arXiv preprint arXiv:2306.08815*, 2023.

[11] R. Chandra, V. Zinage, E. Bakolas, J. Biswas, and P. Stone, "Decentralized multi-robot social navigation in constrained environments via game-theoretic control barrier functions," *arXiv preprint arXiv:2308.10966*, 2023.

[12] S. Poddar, C. Mavrogiannis, and S. S. Srinivasa, "From crowd motion prediction to robot navigation in crowds," 2023.

[13] A. Francis, C. Pérez-D'Arpino, C. Li, F. Xia, A. Alahi, R. Alami, A. Bera, A. Biswas, J. Biswas, R. Chandra, H.-T. L. Chiang, M. Everett, S. Ha, J. Hart, J. P. How, H. Karnan, T.-W. E. Lee, L. J. Manso, R. Mirksy, S. Pirk, P. T. Singamaneni, P. Stone, A. V. Taylor, P. Trautman, N. Tsoi, M. Vázquez, X. Xiao, P. Xu, N. Yokoyama, A. Toshev, and R. Martín-Martín, "Principles and guidelines for evaluating social robot navigation algorithms," 2023.

[14] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018.

[15] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1349–1358, 2019.

[16] V. Kosaraju, A. Sadeghian, R. Martin-Martin, I. Reid, H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.

[17] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," *arXiv preprint arXiv:2003.08111*, 2020.

[18] F. Marchetti, F. Becattini, L. Seidenari, and A. Del Bimbo, "Smemo: social memory for trajectory forecasting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[19] J. Sun, Y. Li, H.-S. Fang, and C. Lu, "Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis," *arXiv preprint arXiv:2103.07854*, 2021.

[20] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, *et al.*, "Scene transformer: A unified architecture for predicting multiple agent trajectories," *arXiv preprint arXiv:2106.08417*, 2021.

[21] Y. Yuan, X. Weng, Y. Ou, and K. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," *arXiv preprint arXiv:2103.14023*, 2021.

[22] T. Salzmann, H.-T. L. Chiang, M. Ryll, D. Sadigh, C. Parada, and A. Bewley, "Robots that can see: Leveraging human pose for trajectory prediction," *IEEE Robotics and Automation Letters*, 2023.

[23] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control," *arXiv preprint arXiv:2001.03093*, 2020.

[24] Z. He and R. P. Wildes, "Where are you heading? dynamic trajectory prediction with expert goal examples," in *Proceedings of the International Conference on Computer Vision (ICCV)*, Oct. 2021.

[25] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9710–9719, 2021.

[26] D. Gonon and A. Billard, "Inverse reinforcement learning of pedestrian–robot coordination," *IEEE Robotics and Automation Letters*, 2023.

[27] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, "Maximum entropy inverse reinforcement learning.," in *Aaai*, vol. 8, pp. 1433–1438, Chicago, IL, USA, 2008.

[28] S. Natarajan, G. Kunapuli, K. Judah, P. Tadepalli, K. Kersting, and J. Shavlik, "Multi-agent inverse reinforcement learning," in *2010 ninth international conference on machine learning and applications*, pp. 395–400, IEEE, 2010.

[29] X. Lin, S. C. Adams, and P. A. Beling, "Multi-agent inverse reinforcement learning for certain general-sum stochastic games," *Journal of Artificial Intelligence Research*, vol. 66, pp. 473–502, 2019.

[30] L. Yu, J. Song, and S. Ermon, "Multi-agent adversarial inverse reinforcement learning," in *International Conference on Machine Learning*, pp. 7194–7201, PMLR, 2019.

[31] J. Song, H. Ren, D. Sadigh, and S. Ermon, "Multi-agent generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 31, 2018.

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

[33] S. Lecleac'h, M. Schwager, and Z. Manchester, "Lucidgames: Online unscented inverse dynamic games for adaptive trajectory prediction and planning," *IEEE Robot. Automat. Lett.*, vol. 6, pp. 5485–5492, Jul. 2021.

[34] S. Rothfuß, J. Inga, F. Köpf, M. Flad, and S. Hohmann, "Inverse optimal control for identification in non-cooperative differential games," in *IFAC-PapersOnLine*, vol. 50, pp. 14909–14915, 2017.

[35] F. Köpf, J. Inga, S. Rothfuß, M. Flad, and S. Hohmann, "Inverse reinforcement learning for identification in linear-quadratic dynamic games," in *IFAC-PapersOnLine*, vol. 50, pp. 14902–14908, 2017.

[36] N. Mehr, M. Wang, M. Bhatt, and M. Schwager, "Maximum-entropy multi-agent dynamic games: Forward and inverse solutions," *IEEE Transactions on Robotics*, 2023.

[37] R. E. Kalman, "When is a linear control system optimal?," 1964.

[38] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, p. 1, 2004.

[39] A. Y. Ng, S. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *Icml*, vol. 1, p. 2, 2000.

[40] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded

spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.
[41] R. Chandra, U. Bhattacharya, C. Roncal, A. Bera, and D. Manocha, "Robusttp: End-to-end trajectory prediction for heterogeneous road-agents in dense traffic with noisy sensor inputs," in *Proceedings of the 3rd ACM Computer Science in Cars Symposium*, pp. 1–9, 2019.
[42] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8483–8492, 2019.
[43] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, C. Li, and D. Anguelov, "Tnt: Target-driven trajectory prediction," in *Conference on Robot Learning*, 2020.
[44] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2165–2174, 2017.
[45] N. Rhinehart, R. Mcallister, K. Kitani, and S. Levine, "Precog: Prediction conditioned on goals in visual multi-agent settings," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Los Alamitos, CA, USA), pp. 2821–2830, IEEE Computer Society, nov 2019.
[46] S.-W. Yoo, C. Kim, J. Choi, S.-W. Kim, and S.-W. Seo, "Gin: Graph-based interaction-aware constraint policy optimization for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 464–471, 2022.
[47] Z. Cao, E. Biyik, G. Rosman, and D. Sadigh, "Leveraging smooth attention prior for multi-agent trajectory prediction," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 10723–10730, IEEE, 2022.
[48] S. Qi and S.-C. Zhu, "Intent-aware multi-agent reinforcement learning," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 7533–7540, IEEE, 2018.
[49] Y. Wang, F. Zhong, J. Xu, and Y. Wang, "Tom2c: Target-oriented multi-agent communication and cooperation with theory of mind," *arXiv preprint arXiv:2111.09189*, 2021.
[50] N. Rabinowitz, F. Perbet, F. Song, C. Zhang, S. A. Eslami, and M. Botvinick, "Machine theory of mind," in *International conference on machine learning*, pp. 4218–4227, PMLR, 2018.
[51] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *2018 IEEE international Conference on Robotics and Automation (ICRA)*, pp. 4601–4607, IEEE, 2018.
[52] Z. Dai, T. Zhou, K. Shao, D. H. Mguni, B. Wang, and H. Jianye, "Socially-attentive policy optimization in multi-agent self-driving system," in *Conference on Robot Learning*, pp. 946–955, PMLR, 2023.
[53] H. Wu, P. Sequeira, and D. V. Pynadath, "Multiagent inverse reinforcement learning via theory of mind reasoning," *arXiv preprint arXiv:2302.10238*, 2023.
[54] H. He, J. Boyd-Graber, K. Kwok, and H. Daumé III, "Opponent modeling in deep reinforcement learning," in *International conference on machine learning*, pp. 1804–1813, PMLR, 2016.
[55] Z. Zhu, E. Bıyık, and D. Sadigh, "Multi-agent safe planning with gaussian processes," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6260–6267, IEEE, 2020.
[56] G. Papoudakis, F. Christianos, and S. Albrecht, "Agent modelling under partial observability for deep reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 19210–19222, 2021.
[57] D. P. Losey, M. Li, J. Bohg, and D. Sadigh, "Learning from my partner's actions: Roles in decentralized robot teams," in *Conference on robot learning*, pp. 752–765, PMLR, 2020.
[58] S. Parekh, S. Habibian, and D. P. Losey, "Rili: Robustly influencing latent intent," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 01–08, IEEE, 2022.
[59] F. B. Von Der Osten, M. Kirley, and T. Miller, "The minds of many: Opponent modeling in a stochastic game.," in *IJCAI*, pp. 3845–3851, 2017.
[60] K. K. Ndousse, D. Eck, S. Levine, and N. Jaques, "Emergent social learning via multi-agent reinforcement learning," in *International Conference on Machine Learning*, pp. 7991–8004, PMLR, 2021.
[61] A. Xie, D. Losey, R. Tolsma, C. Finn, and D. Sadigh, "Learning latent representations to influence multi-agent interaction," in *Conference on robot learning*, pp. 575–588, PMLR, 2021.
[62] W. Z. Wang, A. Shih, A. Xie, and D. Sadigh, "Influencing towards stable multi-agent interactions," in *Conference on robot learning*, pp. 1132–1143, PMLR, 2022.
[63] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. Ortega, D. Strouse, J. Z. Leibo, and N. De Freitas, "Social influence as intrinsic motivation for multi-agent deep reinforcement learning," in *International conference on machine learning*, pp. 3040–3049, PMLR, 2019.
[64] D.-K. Kim, M. Riemer, M. Liu, J. Foerster, M. Everett, C. Sun, G. Tesauro, and J. P. How, "Influencing long-term behavior in multiagent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 18808–18821, 2022.
[65] S. J. Wright, "Coordinate descent algorithms," *Mathematical programming*, vol. 151, no. 1, pp. 3–34, 2015.
[66] S. Le Cleac'h, M. Schwager, and Z. Manchester, "Algames: a fast augmented lagrangian solver for constrained dynamic games," *Autonomous Robots*, vol. 46, no. 1, pp. 201–215, 2022.
[67] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020.
[68] J. Li, H. Ma, and M. Tomizuka, "Conditional generative neural system for probabilistic trajectory prediction," pp. 6150–6156, 11 2019.
[69] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12126–12134, 2019.
[70] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei, "Peeking into the future: Predicting future person activities and locations in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5725–5734, 2019.
[71] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12085–12094, 2019.
[72] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "Stgat: Modeling spatial-temporal interactions for human trajectory prediction," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6271–6280, 2019.
[73] P. Dendorfer, A. Osep, and L. Leal-Taixe, "Goal-gan: Multimodal trajectory prediction based on goal position estimation," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.
[74] L. Shi, L. Wang, C. Long, S. Zhou, M. Zhou, Z. Niu, and G. Hua, "Sgcn: Sparse graph convolution network for pedestrian trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8994–9003, 2021.
[75] F. Marchetti, F. Becattini, L. Seidenari, and A. Del Bimbo, "MANTRA: Memory augmented networks for multiple trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
[76] N. Shafiee, T. Padir, and E. Elhamifar, "Introvert: Human trajectory prediction via conditional 3d attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16815–16825, 2021.
[77] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," *arXiv preprint arXiv:2004.02025*, 2020.
[78] B. Pang, T. Zhao, X. Xie, and Y. N. Wu, "Trajectory prediction with latent belief energy-based model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11814–11824, 2021.
[79] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 759–776, Springer, 2020.
[80] J. Amirian, B. Zhang, F. V. Castro, J. J. Baldelomar, J.-B. Hayet, and J. Pettré, "Opentraj: Assessing prediction complexity in human trajectories datasets," in *Proceedings of the asian conference on computer vision*, 2020.
[81] S. Saadatnejad, Y. Gao, H. Rezatofighi, and A. Alahi, "Jrdb-traj: A dataset and benchmark for trajectory forecasting in crowds," *arXiv preprint arXiv:2311.02736*, 2023.
[82] Y. Gao, "Social-pose: Human trajectory prediction using input pose," 2022.
[83] R. Martin-Martin, M. Patel, H. Rezatofighi, A. Shenoi, J. Gwak, E. Frankel, A. Sadeghian, and S. Savarese, "Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 6, pp. 6748–6765, 2021.
[84] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, "Real-time vehicle detection based on improved yolo v5," *Sustainability*, vol. 14, no. 19, p. 12274, 2022.