Structured Imitation Learning of Interactive Policies through Inverse Games

Max M. Sun, Todd Murphey
Center for Robotics and Biosystems, Northwestern University, Evanston, IL 60208
Email: msun@u.northwestern.edu
Project website: https://murpheylab.github.io/inverse-mixed-strategy

***Abstract*—Generative model-based imitation learning methods have recently achieved strong results in learning high-complexity motor skills from human demonstrations. However, imitation learning of interactive policies that coordinate with humans in shared spaces without explicit communication remains challenging, due to the significantly higher behavioral complexity in multi-agent interactions compared to non-interactive tasks. In this work, we introduce a structured imitation learning framework for interactive policies by combining generative single-agent policy learning with a flexible yet expressive game-theoretic structure. Our method explicitly separates learning into two steps: first, we learn individual behavioral patterns from multi-agent demonstrations using standard imitation learning; then, we structurally learn inter-agent dependencies by solving an inverse game problem. Preliminary results in a synthetic 5-agent social navigation task show that our method significantly improves non-interactive policies and performs comparably to the ground truth interactive policy using only 50 demonstrations. These results highlight the potential of structured imitation learning in interactive settings.**

I. Introduction

Advances in generative models have significantly increased the capabilities of imitation learning methods [4, 11]—motor skill learning paradigms that generate action policies for robots by capturing the statistical behavioral patterns in human demonstrations—enabling high-complexity tasks that are challenging for conventional model-based methods, such as dexterous manipulation [19, 21], autonomous driving [15, 20], and agile locomotion [1, 10]. However, most existing generative model-based imitation learning methods focus on tasks in non-interactive environments, while real-world deployment requires robots to coordinate actions with humans in shared spaces without explicit communication, such as avoiding collisions during navigation [16, 18], coordinating manipulation on the same object [5, 6], and expressing emotional behaviors on robotic characters [7]. Since each agent's action influences all others, such environments require learning interactive policies that not only plan for the robot but also anticipate other agents' intents and actions. Imitation learning of such policies from multi-agent demonstrations remains an open challenge.

Compared to single-agent policies in non-interactive environments, multi-agent interactions exhibit greater behavioral complexity, making learning difficult [2]. This complexity arises not just from the number of agents, but also the intertwined nature of decision-making. Each agent's action simultaneously influences all others, introducing extra inter-agent dependencies. Furthermore, each agent's actions are governed by both individual intents (e.g., reaching a goal) and collective intents shared with the group (e.g., avoiding collisions). Yet, the influence of these intents in demonstrations is subtle, making it especially difficult for learning methods to distinguish and capture behaviors driven by different intents.

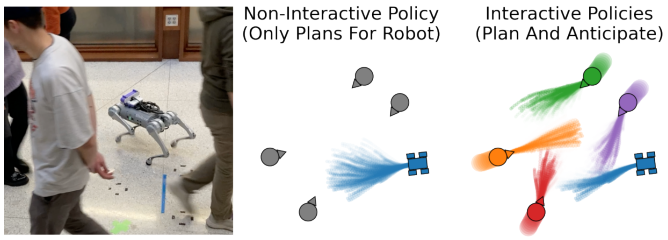

Fig. 1: For example, social navigation requires the robot to not only plan for itself but also anticipate the actions of surrounding humans to effectively coordinate with them.

Instead of relying solely on increasing dataset size to address this complexity, we propose combining generative model-based imitation learning with structured interaction models compatible with generative paradigms. Specifically, our method separates the learning of interactive policies into two stages. First, we leverage generative models to capture individual, non-interactive behavioral patterns from multi-agent data as a standard single-agent imitation learning problem. Then, we model inter-agent dependencies as a game-theoretic optimization problem, whose solution updates the non-interactive policies to incorporate interactions. Importantly, the game structure can be learned from the demonstration data as a neural network, preserving the behavioral model's expressiveness while improving its ability to capture interactive patterns. We show preliminary results in a 5-agent synthetic social navigation benchmark, where our method significantly improves the non-interactive policy and performs comparably with the ground truth using only 50 demonstrations. These results highlight the potential of structured imitation learning frameworks in interactive environments.

II. Methodology

A. Problem formulation

We denote a multi-agent dataset as $\mathcal{D}_N = \{d_1, \ldots, d_N\}$ containing N expert demonstrations of multi-agent interactions. A demonstration $d_i = \{\tau_{i,1}, \ldots, \tau_{i,M_i}\}$ contains state-

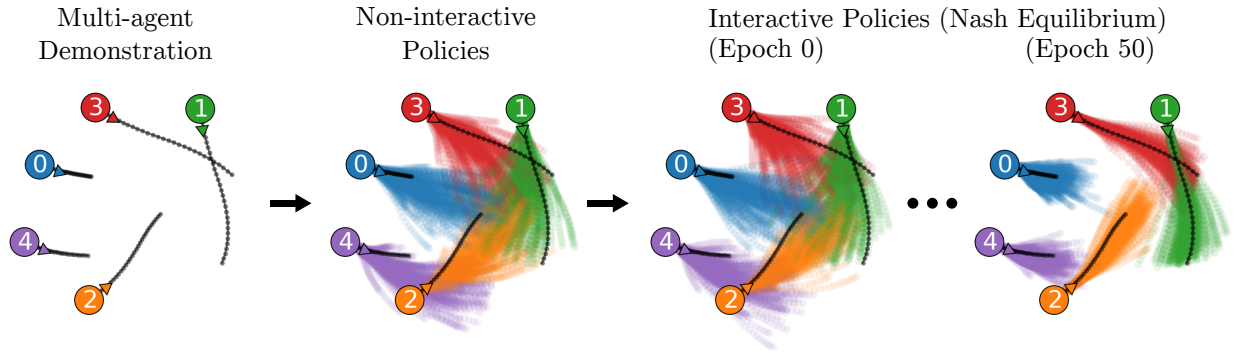

Fig. 2: Overview of the structured imitation learning framework. Given a multi-agent demonstration dataset (dark lines indicate demonstrated actions), we first learn the non-interactive policies using standard single-agent imitation learning methods based on generative models. The interactive policies are the Nash equilibrium of a game-theoretic optimization problem based on the non-interactive policies. The cost function of the game-theoretic problem is modeled as a neural network and optimized based on the MLE formula (7).

action sequences from M_i expert agents. A state-action sequence $\tau_{i,j} = (s_{i,j,1}, a_{i,j,1}), \ldots, (s_{i,j,T_i}, a_{i,j,T_i})$ contains T_i state-actions pairs from a single expert agent. We assume the expert agents are homogeneous and are subject to the same transition dynamics $p(s'|s,a)$. Note that the number of expert agents M_i and the number of time steps T_i contained in one demonstration d_i could vary between demonstrations.

Definition 1 (Interactive policy)**.** *The interactive policy for agent j, denoted as $\pi_\theta^{(j)}(a_j|s_{1\ldots M})$, is a distribution of the agent's actions conditioned on the joint observation of all agents' states.*

We formulate the problem of imitation learning of interactive policies as the following maximum likelihood estimation (MLE) problem:

$$\theta^* = \arg\max_\theta \sum_{i=1}^{N} \sum_{j=1}^{M_i} \sum_{t=1}^{T_i} \log \pi_\theta^{(j)}(a_{i,j,t}|s_{i,1\ldots M_i,t}). \quad (1)$$

The MLE formula (1) simultaneously optimizes the interactive policies for all the agents within a demonstration to capture the influences between the agents during decision-making. At the runtime after training, the robot can infer the joint actions of all agents through:

$$\pi_\theta(a_{1\ldots M}|s_{1\ldots M}) = \prod_{j=1}^{M} \pi_\theta^{(j)}(a_j|s_{1\ldots M}), \quad (2)$$

which enables the robot to simultaneously plan its own actions while anticipating other agents' actions (see Fig. 1 for an example in social navigation).

However, learning the interactive policies from multi-agent demonstrations is challenging. Given the same joint observation of all agents' states in a multi-agent demonstration, the interactive policies must produce actions for each agent in a decentralized manner, while remaining close to the joint actions demonstrated by all agents in the dataset. Our goal in this paper is to introduce a structured interaction model into the interactive policy formula and the MLE problem (1) to simplify the learning problem without compromising the effectiveness of the learned policies.

B. Game-theoretic structure for interactive policies

Definition 2 (Non-interactive policy)**.** *A non-interactive policy of agent j, denoted as $\overline{\pi}_\phi^{(j)}(a|s)$, is an individual decision-making policy that describes the decision-making of agent j without considering other agents.*

Learning a non-interactive policy is equivalent to a standard single-agent imitation learning problem, where the multi-agent dataset is viewed as a collection of single-agent state-action sequences:

$$\phi^* = \arg\max_\phi \sum_{i=1}^{N} \sum_{j=1}^{M_i} \sum_{t=1}^{T_i} \log \overline{\pi}_\phi^{(j)}(a_{i,j,t}|s_{i,j,t}). \quad (3)$$

To simplify notation, we denote the non-interactive policy $\overline{\pi}_\phi^{(j)}(a|s)$ with the optimal parameter ϕ^* as $\overline{\pi}^{(j)}(a|s)$.

Learning the non-interactive policies in (3) is significantly easier compared to learning interactive policies in (1) due to the significantly simplified conditional variable—the non-interactive policy only depends on a single agent's state instead of the joint states of all agents—and this is a well-studied problem with various well-performing methods, such as conditional variational autoencoders [8], diffusion models [4] and flow-based models [5].

Definition 3 (Interaction game)**.** *Each agent in the interaction game optimizes an individual policy $\pi^{(j)}(a|s)$ with respect to an individual objective that depends on other agents' policies:*

$$J^{(j)}(\pi^{(1)}, \ldots, \pi^{(M)}) = \sum_{k \neq j}^{M} \mathbb{E}_{\pi^{(j)}, \pi^{(k)}}[l_\gamma] + D_{KL}(\pi^{(j)} \| \overline{\pi}^{(j)}), \quad (4)$$

where $l_\gamma(s, a, s', a')$ is a parameterized joint loss function for the state-action pairs from two policies $\pi^{(j)}(a|s)$ and $\pi^{(k)}(a'|s')$, and D_{KL} is the KL-divergence.

The first term in the objective function (4) represents the

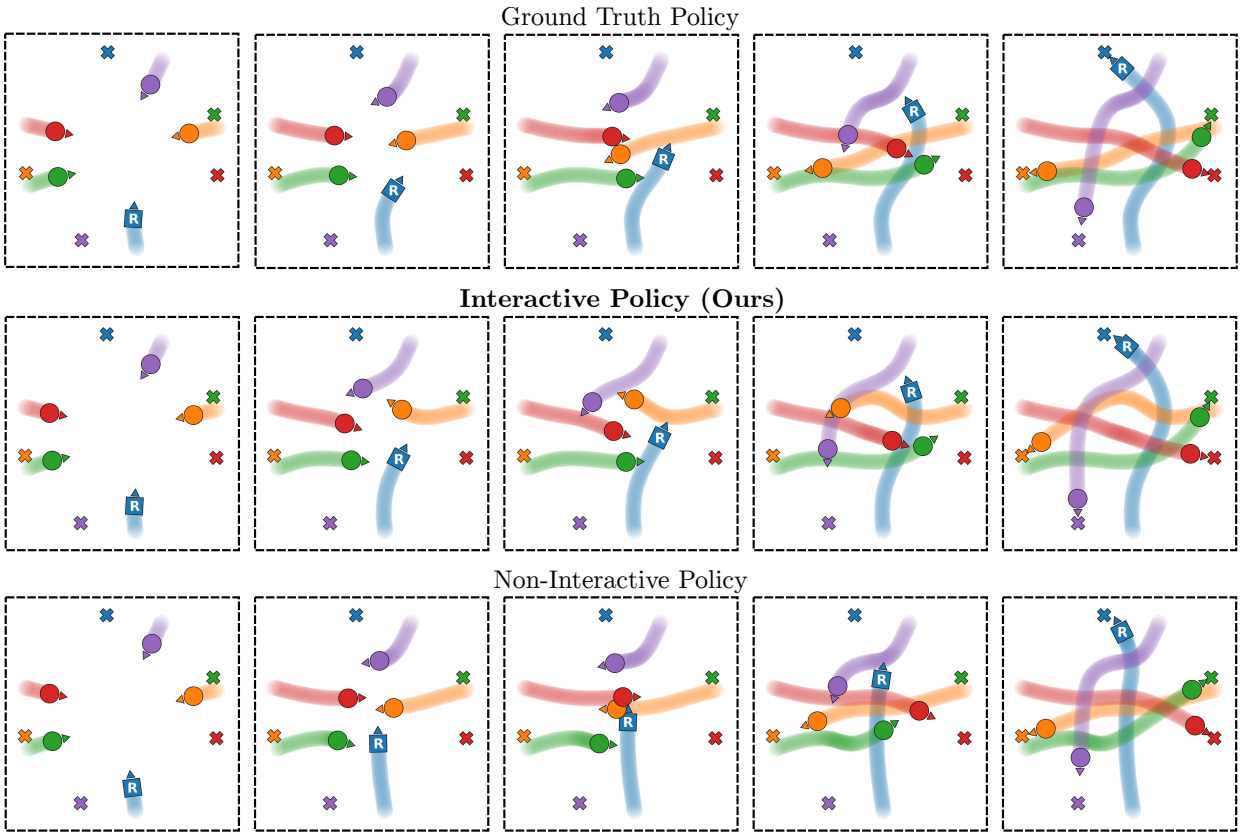

Fig. 3: Qualitative results from the social navigation benchmark, where the letter "R" indicates the robot and the cross indicates the navigation goal of an agent. Learning from only 50 demonstrations, the proposed interactive policy significantly improves the safety performance of the non-interactive policy without compromising the efficiency, while performing comparably to the ground-truth policy.

collective intent shared among all the agents, such as avoiding collisions with others in navigation tasks. The second term in (4) represents the individual intent of the agent, where the KL-divergence regulates the agent's current policy from deviating away from the agent's non-interactive policy.

Definition 4 (Nash equilibrium)**.** *A set of policies form a Nash equilibrium, denoted as* $(\pi^{(1)^*}, \dots, \pi^{(M)^*})$*, if and only if the following holds for all agents [14]:*

$$\pi^{(j)^*} = \arg\min_{\pi^{(j)}} J^{(j)}(\pi^{(1)^*}, \dots, \pi^{(j)}, \dots, \pi^{(M)^*}), \ \forall j. \quad (5)$$

The intuition behind Nash equilibrium is that it describes the scenario where all agents are simultaneously satisfied with its current policy given other agents' current policies, in which case no rational agent is willing to unilaterally change the policy.

Assumption 1. *We assume the interactive policies in (1) form a Nash equilibrium of the game formula (4).*

This assumption integrates a game-theoretic structure into the formulation of the interactive policies in the MLE problem (1), where we formulate the decision-making of each expert agent in the dataset as an explicit game-theoretic optimization problem defined in (4). Importantly, if the joint cost function l_γ is known in (4), an iterative optimization algorithm is proposed in [13] to efficiently solve for the Nash equilibrium (5) with guaranteed convergence. Therefore, given the non-interactive policies, the interactive policies as Nash equilibrium are parameterized by the parameter γ in the joint loss function (4):

$$\begin{aligned} &\pi_\gamma^{(1)}(a_1|s_{1\dots M}), \dots, \pi_\gamma^{(M)}(a_M|s_{1\dots M}) \\ &= NE(l_\gamma, \overline{\pi}^{(1)}, \dots, \overline{\pi}^{(M)}, s_{1\dots M}), \end{aligned} \quad (6)$$

where NE denotes the algorithm from [13] solving for the Nash equilibrium.

C. Learning interactive policies as inverse games

Fully specifying the interactive policies as the Nash equilibrium of the game formula (4) requires specifying the joint loss function l_γ, which is unknown a priori. In [17], it is shown that the interactive policies (6) under Assumption 1 are differentiable with respect to the joint loss function parameter γ. Therefore, we can formulate the MLE problem for interactive policies (1) as the following MLE problem for learning the joint loss function:

$$\gamma^* = \arg\max_\gamma \sum_{i=1}^{N} \sum_{j=1}^{M_i} \sum_{t=1}^{T_i} \log \pi_\gamma^{(j)}(a_{i,j,t}|s_{i,1\dots M_i,t}), \quad (7)$$

$$\text{s.t. } \pi_\gamma^{(1)}, \dots, \pi_\gamma^{(M_i)} = NE(l_\gamma, \overline{\pi}^{(1)}, \dots, \overline{\pi}^{(M)}, s_{i,1\dots M_i,t}), \ \forall i.$$

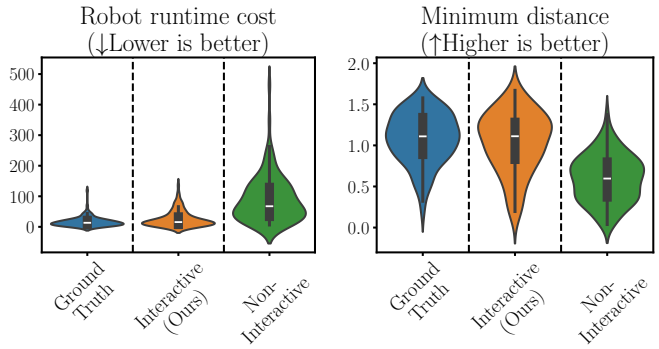

Fig. 4: Quantitative results of the social navigation benchmark (median, quartiles, and distribution of the metrics). The proposed interactive policy has comparable performance with the ground-truth policy and outperforms the non-interactive policy.

Since the calculation of the Nash equilibrium is differentiable, we model the joint cost function l_γ as a neural network (e.g., a multi-layer perceptron) and solve the MLE problem (7) through backpropagation.

This problem of learning the cost function of a game formula is known as *inverse games*, which is the multi-agent equivalent of the inverse optimal control (IOC) or inverse reinforcement learning (IRL) problem.

III. Experiment

A. Experiment design

We design a social navigation task with 100 randomized trials, where a group of 5 agents (one of them being the robot during tests) coordinate collision avoidance while reaching their individual goals. We simulate the agents using the iLQGames algorithm [9], a commonly used dynamic game solver. We model each agent as a circular disk under the Dubins car dynamics. The individual runtime cost function of each agent for iLQGames is specified by a navigation goal, a preferred longitudinal velocity, and a straight line reference trajectory from the current position to the goal with the preferred longitudinal velocity.

B. Implementation details

We implement both the iLQGames algorithm and our method in JAX [3] and Flax [12]. We implement the non-interactive policy as a conditional variational autoencoder (CVAE). We implement the joint cost function as a multi-layer perceptron (MLP). We collect 50 navigation trials as the training data, where all 5 agents are simulated using iLQGames. In each trial, we uniformly sample the initial position of each agent on a circle and uniformly sample the parameters for the individual runtime cost function of each agent.

C. Experiment results

We compare the proposed interactive policy with the ground truth iLQGames policy and the non-interactive policy that the Nash equilibrium is calculated based on. Qualitative results from one representative navigation trial are shown in Fig. 3, where we show the actions of the robot controlled by different policies with the same condition. In each trial, the non-robot iLQGames agents operate under the assumption that the robot is an iLQGames agent with a presumed runtime cost function. To quantitatively evaluate how closely each policy behaves compared to the presumed iLQGames policy, we evaluate the state-action trajectories produced by each policy under the corresponding iLQGames runtime cost function, with the results shown in Fig. 4 (left). Since navigation is a safety-critical task, we further evaluate the minimum distance between the robot and other agents in each trial, shown in Fig. 4 (right).

From the qualitative and the quantitative results, we can see that the proposed interactive policy significantly outperforms the corresponding non-interactive policy. In particular, the interactive policy improves the safety performance (minimum distance) without compromising the navigation efficiency (runtime cost). Furthermore, we can also see that the proposed interactive policy performs comparably to the ground-truth iLQGames policy, closely imitating its behavior from only 50 demonstrations. These preliminary results demonstrate the potential of structured imitation learning methods in interactive environments with a limited number of demonstrations.

IV. Conclusion and Discussion

This work addresses the problem of imitation learning of interactive policies from multi-agent demonstrations. Despite the inherent high-complexity of the problem, we propose a structured imitation learning framework that combines single-agent generative model-based imitation learning with a game-theoretic interaction model. Through a social navigation benchmark, we show that leveraging explicit structure for modeling multi-agent interaction significantly improves the data efficiency, where the proposed interactive policy can perform comparably to the ground truth policy from only 50 demonstrations.

The proposed method is compatible with any generative model-based single-agent imitation learning method, since the game-theoretic optimization problem (4) can be solved using arbitrary non-interactive policies. Ongoing work includes integrating the method with a wider range of imitation learning methods and expanding the task beyond navigation to domains such as cooperative manipulation. Lastly, we plan to investigate the game formula (4) further, aiming to improve the computational efficiency of the inverse game process, enabling rapid learning and adaptation of interactive policies with online observations.

Acknowledgments

This work is supported by the National Science Foundation grant CNS-2237576. The views expressed are the authors' and not necessarily those of the funders.

References

[1] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning Agile

Robotic Locomotion Skills by Imitating Animals. In *Robotics: Science and Systems XVI*. 2020.

[2] Andreea Bobu, Andi Peng, Pulkit Agrawal, Julie A Shah, and Anca D. Dragan. Aligning Human and Robot Representations. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '24, pages 42–54, New York, NY, USA, 2024. Association for Computing Machinery.

[3] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[4] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[5] Eugenio Chisari, Nick Heppert, Max Argus, Tim Welschehold, Thomas Brox, and Abhinav Valada. Learning Robotic Manipulation Policies from Point Clouds with Conditional Flow Matching. In *8th Annual Conference on Robot Learning*, 2024.

[6] Sammy Christen, Wei Yang, Claudia Pérez-D'Arpino, Otmar Hilliges, Dieter Fox, and Yu-Wei Chao. Learning Human-to-Robot Handovers from Point Clouds. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9654–9664, Vancouver, BC, Canada, 2023. IEEE.

[7] Sammy Christen, David Müller, Agon Serifi, Ruben Grandia, Georg Wiedebach, Michael A. Hopkins, Espen Knoop, and Moritz Bächer. Autonomous Human-Robot Interaction via Operator Imitation, 2025. arXiv:2504.02724 [cs].

[8] John Co-Reyes, YuXuan Liu, Abhishek Gupta, Benjamin Eysenbach, Pieter Abbeel, and Sergey Levine. Self-Consistent Trajectory Autoencoder: Hierarchical Reinforcement Learning with Trajectory Embeddings. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1009–1018. 2018.

[9] David Fridovich-Keil, Ellis Ratner, Lasse Peters, Anca D. Dragan, and Claire J. Tomlin. Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1475–1481, 2020.

[10] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. HumanPlus: Humanoid Shadowing and Imitation from Humans. In *Conference on Robot Learning*, 2024.

[11] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile ALOHA: Learning Bimanual Mobile Manipulation using Low-Cost Whole-Body Teleoperation. In *Conference on Robot Learning*, 2024.

[12] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2024.

[13] Max Muchen Sun, Francesca Baldini, Katie Hughes, Peter Trautman, and Todd Murphey. Mixed strategy Nash equilibrium for crowd navigation. *The International Journal of Robotics Research*, page 02783649241302342, 2024.

[14] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36 (1):48–49, 1950.

[15] Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Agile Autonomous Driving using End-to-End Deep Imitation Learning. In *Robotics: Science and Systems XIV*. 2018.

[16] Claudia Pérez-D'Arpino and Julie A. Shah. Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6175–6182, 2015.

[17] Max Muchen Sun, Pete Trautman, and Todd Murphey. Inverse Mixed Strategy Games with Generative Trajectory Models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[18] Pete Trautman, Jeremy Ma, Richard M. Murray, and Andreas Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. *The International Journal of Robotics Research*, 34(3):335–356, 2015.

[19] Yanwei Wang, Nadia Figueroa, Shen Li, Ankit Shah, and Julie Shah. Temporal Logic Imitation: Learning Plan-Satisficing Motion Policies from Demonstrations. In *Proceedings of The 6th Conference on Robot Learning*, pages 94–105. 2023.

[20] Catherine Weaver, Chen Tang, Ce Hao, Kenta Kawamoto, Masayoshi Tomizuka, and Wei Zhan. BeTAIL: Behavior Transformer Adversarial Imitation Learning From Human Racing Gameplay. *IEEE Robotics and Automation Letters*, 9(8):7302–7309, 2024.

[21] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations. In *Robotics: Science and Systems XX*. 2024.