2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

SocialGAIL: Faithful Crowd Simulation for Social Robot Navigation

Bo Ling[1,†], Yan Lyu[1,†], Dongxiao Li[1], Guanyu Gao[2], Yi Shi[1], Xueyong Xu[3,*], Weiwei Wu[1,*]

***Abstract*— Navigation through crowded human environments is challenging for social robots. While reinforcement learning has been adopted for its capacity to capture complex interactions, the training process often relies on simulators to replicate realistic crowd behaviors, ensuring cost-efficiency. Existing crowd simulation methods typically rely on either handcrafted rules, which may lead to overly aggressive navigation, or learning from human trajectory demonstrations, which can be challenging to generalize effectively. In this paper, we introduce a data-driven crowd simulation method called SocialGAIL, which leverages Generative Adversarial Imitation Learning (GAIL) to emulate real pedestrian navigation in crowded environments. SocialGAIL utilizes an attention-based graph neural network to encode observations and employs a generator-discriminator architecture to closely mimic pedestrian behavior. We propose a set of metrics to evaluate the faithfulness of crowd simulation. Experimental results demonstrate that SocialGAIL outperforms baseline methods in terms of goal-reaching, intermediate state faithfulness, trajectory faithfulness, and adherence to global trajectory patterns. The code of our approach is available at https://github.com/William-island/SocialGAIL.**

I. Introduction

Mobile robots are increasingly being deployed in public places such as stations, hotels and shopping malls to provide services like visitor guidance, goods delivery, and security surveillance [1]. Among the prominent challenges faced by these robots is navigating through crowded human environments [2]. While conventional robot navigation methods focus on path planning for efficiency and safety with handcrafted rules or optimization techniques [3], [4], more advanced approaches have incorporated reinforcement learning (RL) to train robots interact with dynamic environments, as RL is particularly capable in capturing complex interactions and cooperation among agents in crowded spaces [5].

Given the potential inefficiency and costliness of direct training with the robots in real-world human environments, the RL based techniques often utilize simulators to mimic crowd interactions with the robot agent. In the pursuit of training an effective navigation policy for real-world deployment, it becomes paramount to make the crowd simulation as realistic as possible, i.e., simulated crowds should be able to navigate themselves like real humans [6].

Unfortunately, most of existing crowd simulation techniques adopted handcrafted rules to control human agents, such as humans are attracted by their goals [7] and there will be a repulsive force to avoid obstacles [4] and collisions [8]. While these rules provide the abilities of goal reaching and local collision avoidance, training social robots with these agents may lead to the development of an overly aggressive navigation policy [9]. To be more real, another approach is to directly replay real-world human trajectories in the simulator, however, the human agents can not be reactive, which may lead to the freezing robot problem [3].

Recent efforts have been focused on data-driven approaches to learn navigation policies from actual human trajectory data, including searching similar navigation actions for demonstration [10]–[13], supervising the human agents by real human trajectories with behavior cloning [14]–[16], and assessing navigation interactions in deep reinforcement learning based on their approximation to real trajectories to guide human agents [17], [18]. These approaches, however, are critically dependent on the trajectory dataset, necessitating the inclusion of diverse and comprehensive scenarios, which is quite challenging to ensure the generalizability of the learned navigation policies.

In this paper, we propose a data driven crowd simulation method SocialGAIL to imitate navigation policy of real pedestrians in a crowded environment with Generative Adversarial Imitation Learning (GAIL). To determine a navigation action (i.e., direction and speed in next time step) for a human agent, SocialGAIL observes state of the agent and encodes the motions and social relations between the target agent and its neighboring agents with an attention-based graph neural network (GNN). It employs a generator to receive the embedded state and output the action to interact with the simulator. Discriminator is used to distinguish between the state-action pairs from the generator and those from human trajectory demonstrations, in order to provide rewards. By iteratively interacting with the simulator and comparing with the real pedestrian demonstrations, SocialGAIL learns a navigation policy that closely resembles human-like behavior. In addition, we propose a set of metrics to assess the faithfulness of crowd simulation, encompassing the perspectives of goal-reaching, intermediate navigation states, individual trajectories, and adherence of global patterns. In summary, our contributions are

- A data-driven crowd simulation technique SocialGAIL to imitate navigation policy of real pedestrians by interacting with crowded environments.
- A motion graph social attention module to encode dynamic state of an agent by graph and social interaction by attention mechanism.
- A set of evaluation metrics to assess faithfulness of crowd simulation.

[†]These authors contributed equally. *Corresponding Author.
[1]The authors are with Southeast University, Nanjing, China. {boling, lvyanly, dongxiaoli, shiyi, weiweiwu}@seu.edu.cn
[2]The author is with Nanjing University of Science and Technology, Nanjing, China. gygao@njust.edu.cn
[3]The author is with North Information Control Research Academy Group Co., Ltd., Nanjing, China. xxyyeah@163.com

979-8-3503-8457-4/24/$31.00 ©2024 IEEE

- Experiment results show that our SocialGAIL outperforms baseline methods in terms of goal-reaching, intermediate state faithfulness, trajectory faithfulness, and adherence to global trajectory patterns.

II. Related Works

In this section, we discuss related crowd simulation approaches and corresponding evaluation metrics.

A. Crowd Simulation for Robot Navigation

Many simulators have been developed for robot social navigation [19]. Yet only a few simulators focus on simulating social behaviors for crowds [20], [21], with behavior graphs [22], [23] or behavior trees [6] for more realistic individual reactions. The movements of crowds in those simulators are mostly controlled by conventional rule-based approaches including Social Force [7], Optimal Reciprocal Collision Avoidance (ORCA) [4] and Pedestrian Velocity Obstacle (PedVO) [8]. These methods share a golden rule to avoid collision as much as possible. While they are effective in simulating crowds with realistic pedestrian velocities and crowd densities [4], [7], utilizing them for training social robots may lead to the development of an overly aggressive navigation policy, increasing the risk of collisions.

For more realistic crowd simulations, real pedestrian trajectories have been incorporated. A common method involves replaying these trajectories in the simulator [19]. However, this approach lacks adaptability since human agents cannot dynamically respond to the presence of robots. Robots often face constraints or become "frozen" due to limited available space [3]. More advanced techniques are:

Database searching methods use a database to store local pedestrian states and their corresponding actions. During simulations, human agents compare their state to the database, selecting a similar entry and adopting its actions. These methods prioritize either efficient searching [12] or precise state similarity measurement [13]. However, their generalizability is limited by database coverage.

Behavior cloning learns navigation policies from human trajectories. Recent work focuses on improved state encoding [15], [16] and adaptability to diverse scenarios [24]. However, it suffers from the compounding error problem [25], [26], where small errors in the learned network lead to significant deviations over time. Instead, we use Generative Adversarial Imitation Learning (GAIL) for better robustness.

Reinforcement learning in crowd simulation was used for collision avoidance and goal-reaching [27]–[30]. Some work pursues faithfulness with data-driven rewards, such as crowd profile-based rewards [17] and state distribution similarity rewards [18]. However, handcrafted rewards may fall short in evaluating action outcomes. In contrast, our SocialGAIL uses a discriminator to provide more accurate rewards based on actual state distributions.

B. Evaluation Metric for Crowd Simulation

Crowd simulation has been evaluated by a range of statistical metrics including crowd density [24], velocity [24],

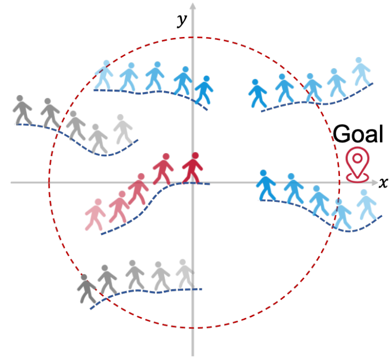

Fig. 1. State of a human agent, including motions of the target human agent (in red) and motions of its neighboring human agents (in blue and grey) within a certain radius. We utilize a target agent-centric coordinate frame, wherein the x-axis is oriented towards the agent's intended goal.

average changes in velocity and acceleration and collision rate [31]. These statistics were used to evaluate the risks of over-crowdedness or capacity of an indoor space [8]. Other metrics focused on local similarities of trajectories including state-action similarity [32], outlier-detection of trajectory [33] and trajectory distribution entropy [34]. Global movement patterns of crowds are also evaluated by clustering the paths [35], [36]. While existing metrics serve distinct purposes, a notable gap remains in the availability of comprehensive measures for evaluating *faithfulness* of crowd simulation. In this paper, we introduce a set of metrics to assess the faithfulness of crowd simulation, encompassing the perspectives of goal-reaching, intermediate navigation states, individual trajectories, and adherence of global patterns.

III. Problem Statement

We discretize time into equal-length time steps and record the locations of each individual *human agent* at each time step in our simulator. The sequence of locations over time constitutes the *trajectory* of a human agent. To determine navigation actions for each agent individually, we introduce a distinction between the **target human agent**, currently needed to determine a navigation action, and the other agents who have either executed their actions or are awaiting their turn to do so. Next, we formally define *motion*, *state* and *action* of a human agent, and the problem to determine the action that replicates the behavior of real pedestrians, ensuring the realism of crowd simulation.

Motion of a human agent i at time t, denoted by m_t^i, is the movement from location (x_{t-1}^i, y_{t-1}^i) at time $t-1$ to location (x_t^i, y_t^i) at time t. We utilize a target agent-centric coordinate frame, wherein the x-axis is oriented towards the agent's intended goal (see Fig. 1). That means the current location of the target agent would always be $(0,0)$ at time t. A binary indicator η is employed to indicate position relation between target agent and its neighboring agents. When agent i is a target agent, $\eta = 0$, but its neighboring agents have their own indicators determined by whether the neighboring agents is in front ($\eta = 1$) or behind ($\eta = 0$) the target agent. For consistency, we denote the motion as $m_t^i = (x_{t-1}^i, y_{t-1}^i, x_t^i, y_t^i, \eta)$.

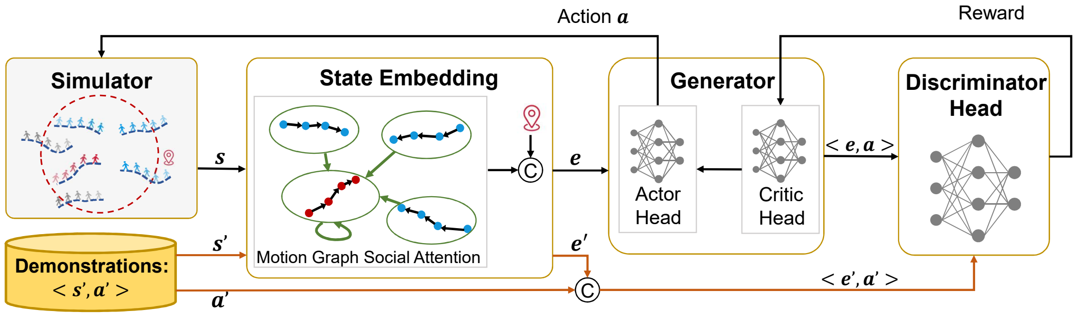

Fig. 2. Overview of SocialGAIL. SocialGAIL observes state of a target agent in Simulator and encodes the motions and social relations between the target agent and its neighboring agents with an attention-based graph neural network (GNN). Generator receives the embedded state as input and outputs the action to interact with the Simulator. Discriminator distinguishes between the state-action pair from the generator and the corresponding pair from human trajectory demonstrations to provide feedback (reward) to the generator.

State of a target agent (see Fig. 1), denoted by s, includes 1) the historical motions of the agent, denoted by $M = \{m_t, m_{t-1}, ..., m_{t-\tau}\}$ in past τ times steps, and 2) historical motions of each neighboring agent within a radius r, denoted by $M^i = \{m^i_t, m^i_{t-1}, ..., m^i_{t-\tau}\}$, $1 \leq i \leq N$, where N denotes the total number of neighboring agents.

Action of a target agent is a speed vector, denoted by $a = (a_x, a_y)$, that navigates the agent to the subsequent location when multiplied by the duration of one time step.

Problem Statement: We aim to learn a navigation policy of each human agent for crowd simulation. This policy determines the speed action a when observing a state s to optimize the faithfulness of the resulting trajectory.

IV. Method

In this section, we present SocialGAIL that imitates navigation policy of real pedestrians in a crowded environment with Generative Adversarial Imitation Learning (GAIL).

A. System Overview

SocialGAIL (as illustrated in Fig. 2) consists of three key components: 1) State Embedding with Motion Graph Social Attention to encode dynamic movements of the target agent and its neighboring agents, and automatically learn their social relations; 2) Generator with Actor-Critic architecture that takes the embedded state e as input and outputs the action a to interact with the simulator; and 3) Discriminator to distinguish between the embedded state-action pair $\langle e, a \rangle$ from the generator and the pair extracted from actual pedestrian trajectory demonstrations to provide more accurate rewards to the critic in the generator. We also design a simulator to simulate human agents interacting with the generator by providing the state and executing the navigation action in subsequent time steps. By iteratively interacting with the simulator and comparing with the real pedestrian demonstrations, SocialGAIL learns a navigation policy that closely resembles human-like behavior. Next, we will delve into the technical details of each component and outline the training procedure for SocialGAIL.

B. State Embedding by Motion Graph Social Attention

Existing data-driven crowd simulation techniques represent a state of a human agent by motion vectors of either

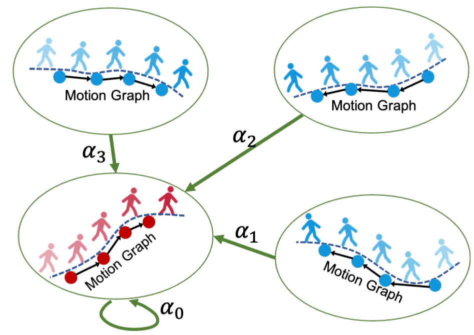

Fig. 3. State Embedding by Motion Graph Social Attention

k nearest neighbours [14], [24] or by the nearest neighbour within radial regions [15], [16]. However, they fail to capture historical motions and social interactions which affect how humans navigate in the crowded environment. Therefore, we are motivated by [37] to utilize graph neural network (GNN) to encode social interactions with an attention mechanism. We also build a subgraph for the agent, called *motion graph*, in which each node corresponds to a motion from a past time step, and there is a edge connecting two successive motions.

Motion Graph. For each agent i, we build a directed subgraph to represent its motion histories. A node represents a motion m^i_k at time k, there will be a directed edge from the motion at the previous time step to the motion of the current time step (see Fig. 3). Then the motion graph M consists of nodes $\{m_0, m_1, ..., m_\tau\}$ in τ time steps and edges between successive motions.

To encode the motion graph with GNN, we define a single layer of motion graph propagation operation as:

$$m_k^{(l+1)} = f_{\text{enc}}(m_k^{(l)}) \oplus \varphi_{\text{agg}}\left(\left\{f_{\text{enc}}(m_j^{(l)})\right\}_{j \in \mathcal{N}(k)}\right) \quad (1)$$

where $m_k^{(l)}$ is the feature of motion m_k in l-th layer. $f_{\text{enc}}(\cdot)$ is a layer for node feature encoding. Features of neighboring nodes ($j \in \mathcal{N}(k)$) are aggregated by maxpooling function $\varphi_{\text{agg}}(\cdot)$, which are further concatenated ($\oplus$) with the k-th node features for higher presentation. We aggregate node features in the last layer by maxpooling to get the final motion embedding g of the motion graph.

Social Attention. To capture social interactions, we further integrate the motion subgraphs into a bigger social graph, in which, each node represents an embedded motion subgraph of an agent, edges connect the target agent to its neighboring agents within a specified radius r. We employ an attention technique to autonomously learn the social weights, allowing us to weigh the significance of interactions between agents within the social graph. Specifically, we take the embedded motion subgraph of the target agent, denoted by g_0, as a query with a linear layer, denoted by ψ_{q}, i.e.,

$$q_0 = \psi_{\mathrm{q}}(g_0). \tag{2}$$

For target agent and its neighboring agent i, we calculated key k_i and value v_i with linear layers ψ_{k} and ψ_{v}, i.e.,

$$k_i = \psi_k(g_i), 0 \leq i \leq N, \tag{3}$$
$$v_i = \psi_v(g_i), 0 \leq i \leq N. \tag{4}$$

Attention score between the target agent and its neighbor i, denoted by α_i, can be calculated by cosine similarity between query and key, i.e.,

$$\alpha_i = \cos(q_0, k_i), 0 \leq i \leq N. \tag{5}$$

The final state embedding, denoted by e, is a concatenation of goal location l_g and the linear combination of all values weighted by the attention scores, i.e.,

$$e = \left(\sum_{i=0}^{N} \mathrm{softmax}(\alpha_i) v_i\right) \oplus l_g. \tag{6}$$

This state embedding module will be implemented in the actor, critic and discriminator networks, respectively.

C. Training Procedure with GAIL

We employ Generative Adversarial Imitation Learning (GAIL) [38] to mimic pedestrian navigation policies. GAIL consists of a generator to interact with the simulator and a discriminator to distinguish between generated and human-demonstrated state-action pairs.

The generator employs an Actor-Critic architecture for action generation. The actor directly observes the state of a target agent in the simulator. After state embedding, both actor and critic heads take the input of e to produce the action a and state value V, respectively, as follows:

$$a = f_a(e; W_a), \tag{7}$$
$$V = f_c(e; W_c), \tag{8}$$

where f_a and f_c represent fully connected (FC) networks in the actor and critic heads respectively, and W_a and W_c denote parameters to be learned in these networks.

The discriminator also encodes states with our Motion Graph Social Attention network. Its head takes either the embedded e and the action a generated by the actor, or the corresponding embedded state e' and action a' from actual pedestrian trajectory demonstration, as input. The discriminator classifies whether the input state-action pair is generated or from human demonstrations with a FC network f_d and outputs a score D, i.e.,

$$D = f_d([e, a]; W_d), \tag{9}$$

where W_d is network parameters to be learned in discriminator. The score D subsequently serves as the reward signal to provide feedback to the critic network in the generator.

The generator and discriminator are trained iteratively in a minimax game to achieve Nash equilibrium. The discriminator aims to discern whether $\langle e, a \rangle$ originates from the learning policy π or the expert policy π_E; while generator is optimized to imitate expert policy π_E by the discriminator's assessment on generated $\langle e, a \rangle$.

To train SocialGAIL, we implement a sequential training strategy for individual human agents. For each human agent, we randomly select a trajectory from the training dataset to set its starting and goal positions. Meanwhile, the other agents continue following their original trajectories from the dataset. This sequential training approach tends to converge to a good policy more effectively compared to training multiple agents simultaneously. We adopt replay buffer to update both the generator and the discriminator. The generator is updated with Proximal Policy Optimization (PPO) [39], while the discriminator is updated through binary classification, with sampled state-action pairs from the buffer labeled 1 and sampled actual demonstrations labeled 0.

V. Crowd Simulation Evaluation Metric

We propose a set of metrics to evaluate faithfulness of crowd simulations from aspects of goal-reaching, states, individual trajectories and global patterns adherence.

Goal Reaching. We evaluate the ability of a human agent to reach its intended goal using the **Final Displacement Error (FDE)**, which measures the Euclidean distance between the simulated agent's final position and its actual goal.

Faithfulness of Individual Trajectory. The shape similarity between a simulated trajectory L and the corresponding true trajectory L' generated by a pedestrian can be measured by **Fréchet Distance** [40], denoted by $F(L, L')$, that considers the location and ordering of the points along the curves.

Faithfulness of Intermediate States. For effective training, robots should ideally find themselves in realistic states throughout their journey. This necessitates a focus on ensuring that intermediate states closely mirror those in real pedestrian demonstrations. We employ the **Hausdorff distance** [41] to measure the similarity of intermediate states with their corresponding ground truth for each agent. We restructure the state representation by treating each historical motion as data points, forming sets. Let S denote the set of motion points from simulation, and S' denote the set of motion points from demonstration. The Hausdorff distance quantifies the maximum discrepancy between S and S', i.e.,

$$H(S, S') = \max\{\max_{m \in S} \min_{m' \in S'} d(m, m'), \max_{m' \in S'} \min_{m \in S} d(m, m')\}, \tag{10}$$

where d is Euclidean distance, which can be directly computed as we have unified axis centering on the target agent and orienting towards its intended goal.

Faithfulness of Global Trajectory Patterns. Fréchet Distance can be used for assessing the similarity of individual trajectories, but it is sensitive to random factors due to

TABLE I
COMPARISON RESULTS OF SOCIALGAIL WITH BASELINE METHODS AND SOCIALGAIL VARIANTS.

Method	Fréchet Distance	FDE	Hausdorff Distance
Database Searching [10]	14.87	13.40	497.91
ORCA [4]	12.47	3.29	410.38
RadarRNN [16]	6.03	4.41	380.52
DRL [43]	6.66	1.31	373.46
SocialBC (ours)	4.50	2.01	343.14
RadialGAIL (ours)	4.31	1.41	341.05
SocialGAIL (ours)	**3.80**	**0.44**	**285.48**

pedestrian movements' inherent randomness. To alleviate the influence of such randomness, we aim to measure the extent to which simulated human agents navigate similarly to the majority of pedestrians with close starting and ending points [36]. To achieve this, we employ Kernel Density Estimation (KDE) to evaluate how closely a simulated trajectory aligns with a collection of real trajectories that share similar origins and destinations. Given a real trajectory set $\{L_1', L_2', \cdots, L_n'\}$ with close starting and ending locations, the similarity of a simulated trajectory L to the set is measured by the kernel density as

$$P(L) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{F(L, L_i')}{h}\right), \tag{11}$$

where $F(L, L_i')$ is the Fréchet distance between L and L_i'. K denotes Gaussian Kernel [42] with a bandwidth $h = 5$. This allows us to assess the extent to which the simulated agent adheres to the prevalent patterns exhibited by the majority of pedestrians in similar situations. Higher $P(L)$ indicates a stronger adherence to the global pattern. We transform $P(L)$ into $-\log(P(L))$ with a negative sign to maintain consistency with distance metrics, where smaller values signify higher similarity.

VI. EXPERIMENT

In this section, we evaluate performance of SocialGAIL using the set of metrics we have proposed. We also perform a case study to demonstrate the effectiveness of SocialGAIL.

A. Experiment Settings and Implementation

1) Data Set: The New York Grand Central (GC) Dataset consists of 12,600 pedestrian trajectories. The busiest scene involves around 290 pedestrians. We divided the dataset into a 70/30 split for training and testing.

2) Baseline Techniques: We compare with the following state-of-the-art crowd simulation methods:

- **ORCA** is a rule-based simulation technique that has been commonly used for social robot navigation.
- **Database Searching** selects the most similar state from the pedestrian trajectory dataset and adopts the associated actions for navigation [10]. Here we measure the similarity by Euclidean distance between state vectors.
- **RadarRNN** considers the nearest neighbors in each radial region in state and learns the navigation policy using a supervised recursive neural network [16].

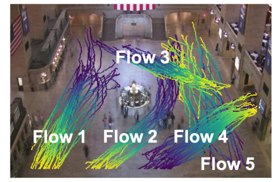

Fig. 4. Identified five trajectory flows (from purple to yellow) with similar origin destination locations in GC dataset.

- **DRL** employs deep reinforcement learning for collisions avoidance and goal reaching [43].

We also compare with SocialGAIL variants to evaluate effectiveness of key components in SocialGAIL.

- **SocialBC** uses our Motion Graph Social Attention module for state embedding but adopts fully connected network for behavior cloning.
- **RadialGAIL** adopts the GAIL framework to train navigation policies but uses motion vectors of agents in each radial region as state representation [16].

3) Implementation: We consider historical motions within 5 time steps ($\tau = 5$) and neighboring agents within a radius of 6 meters ($r = 6$). In SocialGAIL, the actor, critic and discriminator networks all have two layers of graph network (32 units), one layer for social attention (32 units), and three layers of fully connected network (32 units per layer, ReLu activations) for heads. The batch size is 2048 for the PPO and 64 for training discriminator. Adam optimizer [44] is used. The simulator is build with python and models are trained on NVIDIA GeForce RTX 3090, Intel Core i9-10940K CPU @3.30GHz with 32GB memory.

B. Experimental Results

We summarize the performance of SocialGAIL, compared with the baseline methods and SocialGAIL variants from the perspective of goal reaching, state faithfulness, trajectory faithfulness, and global pattern adherence.

1) Faithfulness of goal reaching: FDE measures how close an agent gets to its goal. As shown in Table I, Database Searching fails to reach goal effectively. ORCA, focusing on goal optimization in velocity space, achieves lower FDE compared to RadarRNN using behavior cloning. DRL outperforms all baselines due to its goal-centric reward design. SocialGAIL ranks highest in goal-reaching, followed by DRL and RadialGAIL, demonstrating the benefits of learning goal-reaching policies through environmental interaction.

2) Faithfulness of individual trajectory. : Fréchet Distance measures the similarity between simulated and actual trajectories. In Table I, data-driven methods outperform rule-based approaches (ORCA, Database Searching, DRL) in generating more accurate trajectories. SocialBC outperforms RadarRNN, highlighting the effectiveness of our Motion Graph Social Attention module. SocialGAIL and RadialGAIL produce more faithful trajectories due to their generalizability, even with a limited set of demonstrations.

3) Faithfulness of intermediate states: Hausdorff distance measures the similarity between a simulated state and its corresponding actual state of a pedestrian. Database Searching

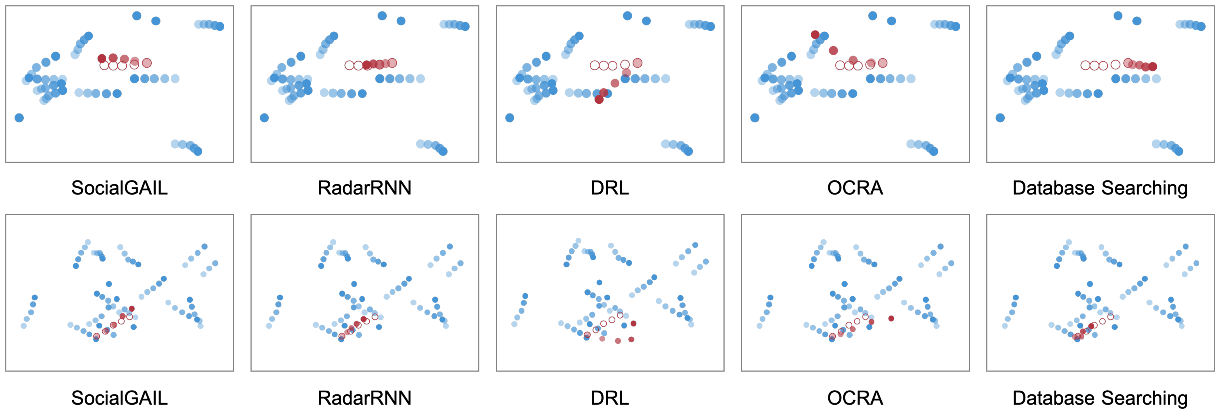

Fig. 5. Case study to understand how an agent reacts to nearby pedestrian movements. Sequence of colored solid dots represents a trajectory of an agent. Red dots represent target agent determining motions while blue dots represent nearby agents replaying actual pedestrian trajectories from the dataset. Darker colors indicate later time steps, and red circles represent the ground truth trajectories of the target agent.

TABLE II
SIMILARITY BETWEEN SIMULATED TRAJECTORIES WITH IDENTIFIED FIVE TRAJECTORY FLOWS.

Method	Flow 1	Flow 2	Flow 3	Flow 4	Flow 5
Database Searching [10]	356.00	916.32	885.58	266.05	393.04
ORCA [4]	259.30	316.75	277.09	167.72	231.50
RadarRNN [16]	222.59	189.65	193.86	144.72	170.51
DRL [43]	166.21	179.03	204.97	144.09	194.99
SocialBC (ours)	161.77	148.36	**148.88**	140.52	151.47
RadialGAIL (ours)	156.08	154.21	163.12	137.41	150.16
SocialGAIL (ours)	**153.52**	**144.98**	154.62	**136.97**	**147.47**

and ORCA yield the least faithful states, while data-driven methods improve faithfulness through human demonstration learning. SocialGAIL produces the most faithful states. Notably, DRL achieves a slightly smaller Hausdorff distance than supervised RadarRNN, despite focusing on collision avoidance rather than faithfulness.

4) Faithfulness of global trajectory pattern: Kernel Density Estimation (KDE) measures the adherence of a simulated trajectory to a flow of human trajectories with similar origins and destinations. We identified 5 popular trajectory flows [36] in the dataset (see Fig. 4). Table II summarizes the average adherence of simulated trajectories to each of the five trajectory flows. We can see that Database Searching performs poorly, failing to align with patterns. SocialBC and SocialGAIL improve faithfulness to global patterns. GAIL-based methods (SocialGAIL and RadialGAIL) outperform DRL and supervised learning (SocialBC and RadarRNN).

In summary, SocialGAIL excels across various aspects of crowd simulation, including goal-reaching, state faithfulness, trajectory faithfulness, and global pattern adherence.

C. Case Study

We conducted a case study to qualitatively evaluate how a human agent reacts to nearby pedestrian movements and how faithful these reactions are. Fig. 5 plots the simulated motions of two target agents (in red) and the nearby human agents that replay actual pedestrian trajectories from the dataset. Darker colors indicate later time steps, and red circles represent the ground truth trajectories.

In the first case (first row in Fig. 5), we can see SocialGAIL reacts most faithfully when faced with front agents approaching. In contrast, RadarRNN becomes conservative and slows down, while DRL chooses a different direction to avoid potential collisions. ORCA tries to accelerate to bypass the crowd, lacking anticipation of future interactions. Database Searching takes a completely incorrect direction. In the second case (second row in Fig. 5), most nearby agents move alongside or away from the target agent, except one passing in front during the first two time steps. SocialGAIL accurately predicts the motion of neighboring agents and chooses the correct direction while slightly accelerating in empty space later. In contrast, RadarRNN and Database Searching opt for a slower speed, while DRL and ORCA select different directions to avoid potential collisions.

VII. CONCLUSIONS

In this paper, we present SocialGAIL, a data-driven crowd simulation method that learns navigation policies from real pedestrian interactions using Generative Adversarial Imitation Learning (GAIL). We proposed an attention-based graph neural network to capture agent movements and social interactions. The generator uses this information to produce actions via an actor network, while the discriminator distinguishes between generated and human trajectory data, providing feedback to the generator. We also propose novel metrics for evaluating simulation faithfulness. Experimental results show that SocialGAIL excels in terms of goal reaching, intermediate state faithfulness, trajectory faithfulness, and global patterns adherence. Future work includes multi-agent GAIL, and improving generalibility in different scenes.

VIII. ACKNOWLEGEMENT

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102200; in part by the Natural Science Foundation of China under Grant 62102082, 62232004, 61972086; in part by the Jiangsu Natural Science Foundation of China under Grant BK20210203, BK20230024.

REFERENCES

[1] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torre-Ferrero, and J. Perez-Oria, "Working together: A review on safe human-robot collaboration in industrial environments," *Ieee Access*, vol. 5, pp. 26 754–26 773, 2017.

[2] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[3] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 797–803.

[4] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research: The 14th International Symposium ISRR*. Springer, 2011, pp. 3–19.

[5] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6015–6022.

[6] N. Pérez-Higueras, R. Otero, F. Caballero, and L. Merino, "Hunavsim: A ros 2 human navigation simulator for benchmarking human-aware robot navigation," *arXiv preprint arXiv:2305.01303*, 2023.

[7] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[8] S. Curtis and D. Manocha, "Pedestrian simulation using geometric reasoning in velocity space," in *Pedestrian and evacuation dynamics 2012*. Springer, 2014, pp. 875–890.

[9] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, "Decentralized structural-rnn for robot crowd navigation with deep reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3517–3524.

[10] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer graphics forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 655–664.

[11] K. H. Lee, M. G. Choi, Q. Hong, and J. Lee, "Group behavior from video: a data-driven approach to crowd simulation," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2007, pp. 109–118.

[12] P. Charalambous and Y. Chrysanthou, "The pag crowd: A graph based approach for efficient data-driven crowd simulation," in *Computer Graphics Forum*, vol. 33, no. 8. Wiley Online Library, 2014, pp. 95–108.

[13] M. Zhao, S. J. Turner, and W. Cai, "A data-driven crowd simulation model based on clustering and classification," in *2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*. IEEE, 2013, pp. 125–134.

[14] Y. Ma, E. W. M. Lee, and R. K. K. Yuen, "An artificial intelligence-based approach for simulating pedestrian movement," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3159–3170, 2016.

[15] X. Wei, W. Lu, L. Zhu, and W. Xing, "Learning motion rules from real data: Neural network for crowd simulation," *Neurocomputing*, vol. 310, pp. 125–134, 2018.

[16] X. Zhao, J. Zhang, and W. Song, "A radar-nearest-neighbor based data-driven approach for crowd simulation," *Transportation Research Part C: Emerging Technologies*, vol. 129, p. 103260, 2021.

[17] A. Panayiotou, T. Kyriakou, M. Lemonari, Y. Chrysanthou, and P. Charalambous, "Ccp: Configurable crowd profiles," in *ACM SIGGRAPH 2022 conference proceedings*, 2022, pp. 1–10.

[18] P. Charalambous, J. Pettre, V. Vassiliades, Y. Chrysanthou, and N. Pelechano, "Greil-crowds: Crowd simulation with deep reinforcement learning and examples," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–15, 2023.

[19] A. Biswas, A. Wang, G. Silvera, A. Steinfeld, and H. Admoni, "Socnavbench: A grounded simulation testing framework for evaluating social navigation," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 11, no. 3, pp. 1–24, 2022.

[20] A. Favier, P.-T. Singamaneni, and R. Alami, "An intelligent human simulation (inhus) for developing and experimenting human-aware and interactive robot abilities," 2021.

[21] ——, "Simulating intelligent human agents for intricate social robot navigation," in *RSS Workshop on Social Robot Navigation 2021*, 2021.

[22] N. Tsoi, M. Hussein, J. Espinoza, X. Ruiz, and M. V'azquez, "Sean: Social environment for autonomous navigation," *Proceedings of the 8th International Conference on Human-Agent Interaction*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:221556248

[23] N. Tsoi, A. Xiang, P. Yu, S. S. Sohn, G. Schwartz, S. Ramesh, M. Hussein, A. W. Gupta, M. Kapadia, and M. Vázquez, "Sean 2.0: Formalizing and generating social situations for robot navigation," *IEEE Robotics and Automation Letters*, vol. 7, pp. 11 047–11 054, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251430109

[24] X. Song, D. Han, J. Sun, and Z. Zhang, "A data-driven neural network approach to simulate pedestrian movement," *Physica A: Statistical Mechanics and its Applications*, vol. 509, pp. 827–844, 2018.

[25] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 661–668.

[26] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[27] F. Martinez-Gil, M. Lozano, and F. Fernández, "Multi-agent reinforcement learning for simulating pedestrian navigation," in *Adaptive and Learning Agents: International Workshop, ALA 2011, Held at AAMAS 2011, Taipei, Taiwan, May 2, 2011, Revised Selected Papers*. Springer, 2012, pp. 54–69.

[28] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 285–292.

[29] L. Casadiego and N. Pelechano, "From one to many: Simulating groups of agents with reinforcement learning controllers," in *Intelligent Virtual Agents: 15th International Conference, IVA 2015, Delft, The Netherlands, August 26-28, 2015, Proceedings 15*. Springer, 2015, pp. 119–123.

[30] J. Lee, J. Won, and J. Lee, "Crowd simulation by deep reinforcement learning," in *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*, 2018, pp. 1–7.

[31] S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman, "Steerbench: a benchmark suite for evaluating steering behaviors," *Computer Animation and Virtual Worlds*, vol. 20, no. 5-6, pp. 533–548, 2009.

[32] A. Lerner, Y. Chrysanthou, A. Shamir, and D. Cohen-Or, "Data driven evaluation of crowds," in *Motion in Games: Second International Workshop, MIG 2009, Zeist, The Netherlands, November 21-24, 2009. Proceedings 2*. Springer, 2009, pp. 75–83.

[33] P. Charalambous, I. Karamouzas, S. J. Guy, and Y. Chrysanthou, "A data-driven framework for visual crowd analysis," in *Computer Graphics Forum*, vol. 33, no. 7. Wiley Online Library, 2014, pp. 41–50.

[34] S. J. Guy, J. Van Den Berg, W. Liu, R. Lau, M. C. Lin, and D. Manocha, "A statistical similarity measure for aggregate crowd dynamics," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, pp. 1–11, 2012.

[35] H. Wang, J. Ondřej, and C. O'Sullivan, "Trending paths: A new semantic-level metric for comparing simulated and real crowd data," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 5, pp. 1454–1464, 2016.

[36] F. He, Y. Xiang, X. Zhao, and H. Wang, "Informative scene decomposition for crowd analysis, comparison and simulation guidance," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 50–1, 2020.

[37] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 525–11 533.

[38] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[40] H. Alt and M. Godau, "Computing the fréchet distance between two polygonal curves," *International Journal of Computational Geometry & Applications*, vol. 5, no. 01n02, pp. 75–91, 1995.

[41] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[42] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *The Annals of Statistics*, pp. 1236–1265, 1992.

[43] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 6252–6259.
[44] Z. Liu, J. Li, and K. Wu, "Context-aware taxi dispatching at city-scale using deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, 2020.