

# Socially Aware Robot Crowd Navigation with Interaction Graphs and Human Trajectory Prediction

Shuijing Liu\*, Peixin Chang\*, Zhe Huang, Neeloy Chakraborty, Weihang Liang, Junyi Geng, and Katherine Driggs-Campbell

**Abstract**—We study the problem of safe and socially aware robot navigation in dense and interactive human crowds. We redefine the personal zones of walking pedestrians with their future trajectories. The predicted personal zones are incorporated into a reinforcement learning framework to prevent the robot from intruding into the personal zones. To learn socially aware navigation policies, we propose a novel recurrent graph neural network with attention mechanisms to capture the interactions among agents through space and time. We demonstrate that our method enables the robot to achieve good navigation performance and non-invasiveness in challenging crowd navigation scenarios in simulation and real world.

## I. INTRODUCTION

As robots are increasingly used in human-centric environments, social navigation is an important yet challenging problem. In public spaces, people have clear norms about personal spaces [1], [2]. A robot that breaks into people’s personal space can make them feel uncomfortable and may result in accidents [3]. To navigate in a safe and socially aware manner, the robot must reason about the interactions in the crowd and avoid intrusions into the personal spaces of its surrounding pedestrians.

Robot navigation in dynamic environments has received much attention for many years [4]–[8]. Most works define the humans’ personal zones as *simplistic circles* to represent the non-traversable regions for the robot, as shown in Fig. 1a [6], [8]–[13]. However, the circular personal zones ignore the past motions and future intentions of walking humans. In addition, these methods only consider the interactions between the robot and humans, while ignoring the human-human interactions. As a result, the navigation performance and the social awareness of the robot deteriorate in denser and more complicated human crowds.

To incorporate future intentions of pedestrians into planning, trajectory-based crowd navigation methods first use trajectory predictors to predict other agents’ trajectories for one step. Then, the predicted trajectories are used to learn state transition probabilities and plan a feasible path for the robot [14]–[17]. However, these methods may still lead to impolite behaviors in Fig. 1a, since (1) the one-step predictions do not capture the long-term intent of each

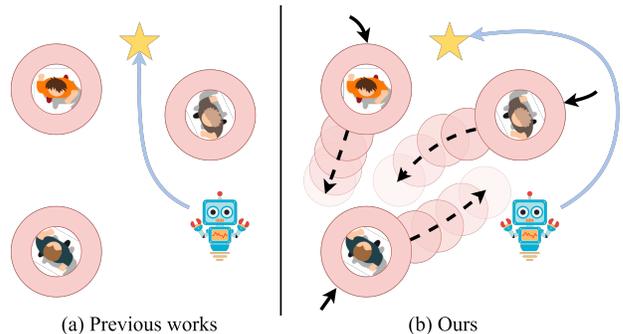


Fig. 1: **The simplistic personal zones and our prediction-based personal zones.** (a) A simplistic social zone of a walking pedestrian is a circle centered at the pedestrian’s position, which may result in unsafe or unsocial robot behaviors. (b) Our prediction-based social zone of a pedestrian is a set of circles centered at its future positions, which improves the performance and social awareness of the robot.

human, and (2) the planner does not penalize the robot if it intrudes into the predicted paths.

To address these problems, we redefine the personal zones of walking humans as a set of circles centered at their future positions, as shown in Fig. 1b, and approximate the personal zones with a pretrained pedestrian trajectory predictor. The *prediction-based personal zones* capture the interactions and future intentions of people in a crowd more accurately than the *simplistic circles*. To learn a robot policy, we incorporate the predicted personal zones into a model-free reinforcement learning (RL) framework, which effectively prevents the robot from intruding into humans’ personal zones. Although the predicted trajectories can make more spaces untraversable and lead to the freezing robot problem [18], our method is less prone to such problems since RL allows the robot to explore the environment and learn from its past experience [13].

We model the crowd navigation scenario as a spatio-temporal (st) interaction graph to capture the interactions among agents through both space and time. Then, we convert the st-interaction graph to a novel end-to-end neural network. In the network, we use multi-head attention to model the human-human and robot-human interactions. With the predicted personal zones and the interaction graph neural network, the robot is proactive, resilient, and non-invasive while navigating through dense and interactive crowds.

The main contributions of this paper are as follows. (1) We redefine the personal zones for walking human crowds with their future motions. We propose a novel method to incorporate the predicted personal zones into a model-free RL framework for robot crowd navigation. (2) We propose

\* denotes equal contribution.

S. Liu, P. Chang, Z. Huang, W. Liang, N. Chakraborty and K. Driggs-Campbell are with the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. emails: {sliu105,pchang17,zheh4,neeloy2,weihang2,krcd}@illinois.edu

J. Geng is with the Robotics Institute at Carnegie Mellon University. email: junyigen@andrew.cmu.edu

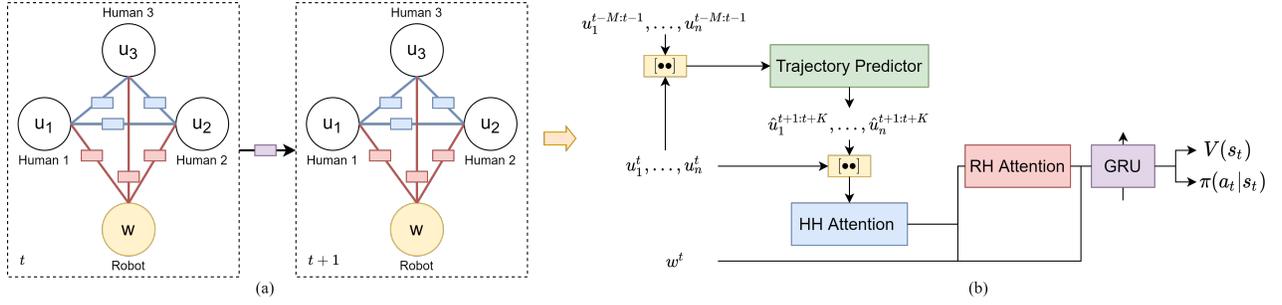


Fig. 2: **The spatial-temporal interaction graph and the network architecture.** (a) Graph representation of crowd navigation. The robot node is denoted by  $w$  and the  $i$ -th human node is denoted by  $u_i$ . HH edges and HH interaction functions are in blue, while RH edges and RH interaction functions are in red. Temporal function that connects the graphs at adjacent timesteps is in purple. (b) Our network. A trajectory predictor is used to predict personal zones. Two attention mechanisms are used to model the human-human interactions and robot-human interactions. We use a GRU as the temporal function.

a novel graph neural network that uses attention mechanism to effectively capture the spatial and temporal interactions among heterogeneous agents. (3) The experiments demonstrate that our method outperforms previous works in terms of navigation performance and social awareness.

## II. METHODOLOGY

In this section, we first formulate the crowd navigation as an RL problem and introduce the social reward function. Then, we present our approach to model the crowd navigation scenario as a spatio-temporal interaction graph, which leads to the derivation of our network architecture.

### A. Preliminaries

1) *MDP formulation:* Consider a robot navigating and interacting with humans in a 2D Euclidean space. We model the scenario as a Markov Decision Process (MDP), defined by the tuple  $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma, \mathcal{S}_0 \rangle$ . Let  $w^t$  be the robot state which consists of the robot's position  $(p_x, p_y)$ , goal position  $(g_x, g_y)$ , maximum speed  $v_{max}$ , heading angle  $\theta$ , and radius of the robot base  $\rho$ . Let  $u_i^t$  be the current state of the  $i$ -th human at time  $t$ , which consists of the human's position  $(p_x^i, p_y^i)$ . Then, the  $K$  future and the  $M$  previous positions of the  $i$ -th human are denoted as  $\hat{u}_i^{t+1:t+K}$  and  $u_i^{t-M:t-1}$ , respectively. We define the state  $s_t \in \mathcal{S}$  of the MDP to be  $s_t = [w^t, u_1^t, \hat{u}_1^{t+1:t+K}, \dots, u_n^t, \hat{u}_n^{t+1:t+K}]$  if a total number of  $n$  humans are observed at the timestep  $t$ , where  $n$  may change within a range in different timesteps.

At each timestep  $t$ , the robot takes an action  $a_t \in \mathcal{A}$  according to its policy  $\pi(a_t|s_t)$ . In return, the robot receives a reward  $r_t$  and transits to the next state  $s_{t+1}$ . Meanwhile, all other humans also take actions according to their policies and move to the next states. The process continues until the robot reaches its goal,  $t$  exceeds the maximum episode length  $T$ , or the robot collides with any humans.

2) *Reward function:* To discourage the robot from intruding into the predicted personal zones of humans, we use a social reward  $r_{social}$  to penalize such intrusions:

$$\begin{aligned} r_{social}^i(s_t) &= \min_{k=1, \dots, K} \left( \mathbb{1}_i^{t+k} \frac{r_c}{2^k} \right) \\ r_{social}(s_t) &= \min_{i=1, \dots, n} r_{social}^i(s_t) \end{aligned} \quad (1)$$

where  $\mathbb{1}_i^{t+k}$  indicates whether the robot collides with the predicted position of the human  $i$  at time  $t+k$  and  $r_c = -20$

is the penalty for collision. We assign different weights to the intrusions at different prediction timesteps, and thus the robot gets less penalty if it intrudes into the predicted social zone further in the future.

In addition, we add a potential-based reward  $r_{pot} = 2(-d_{goal}^t + d_{goal}^{t-1})$  to guide the robot to approach the goal, where  $d_{goal}^t$  is the  $L2$  distance between the robot position and goal position at time  $t$ . Let  $S_{goal}$  be the set of goal states, where the robot successfully reaches the goal, and  $S_{fail}$  be the set of failure states, where the robot collides with any human. Then, the whole reward function is defined as

$$r(s_t, a_t) = \begin{cases} 10, & \text{if } s_t \in S_{goal} \\ r_c, & \text{if } s_t \in S_{fail} \\ r_{pot}(s_t) + r_{social}(s_t), & \text{otherwise.} \end{cases} \quad (2)$$

Intuitively, the robot gets a high reward when it approaches the goal and avoids intruding into the current and future positions of all humans.

### B. Spatio-Temporal Interaction Graph

We formulate the crowd navigation scenario as a spatio-temporal (st) interaction graph. As shown in Fig. 2a, at each timestep  $t$ , our st-interaction graph  $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$  consists of a set of nodes  $\mathcal{V}_t$  and a set of edges  $\mathcal{E}_t$ . The nodes represent the visible agents. The edges connect two different visible agents and represent the spatial interactions between the agents at the same timestep. We divide all edges  $\mathcal{E}_t$  into the human-human (HH) edges that connect two humans and the robot-human (RH) edges that connect the robot and a human. The two types of edges allow us to factorize the spatial interactions into HH interaction function and RH interaction function. Compared with the previous works that ignore HH edges [6], [8], [10], [13], our method considers the pair-wise interactions among all visible agents and thus scales better in dense and highly interactive crowds.

Since the movements of all agents cause the visibility of each human to change dynamically, the set of nodes  $\mathcal{V}_t$  and edges  $\mathcal{E}_t$  and the parameters of the interaction functions may change correspondingly. To this end, we integrate the temporal correlations of the graph  $\mathcal{G}_t$  at different timesteps using another function denoted by the purple box in Fig. 2a. The temporal function connects the graphs at adjacent timesteps, which overcomes the short-sightedness of reactive methods and enables long-term decision-making of the robot.

To reduce the number of parameters, the same type of edges share the same function parameters. This parameter sharing is important for the scalability of our st interaction graph because the number of parameters is kept constant with an increasing number of humans [19].

### C. Network Architecture

As shown in Fig. 2b, we derive our network architecture from the st-interaction graph. In our network, a pretrained trajectory predictor predicts the personal zones of humans. We represent the HH and RH functions as feedforward networks with attention mechanisms, referred to as HH attn and RH attn respectively. We represent the temporal function as a gated recurrent unit (GRU).

1) *Trajectory predictor*: Since the robot has a limited field of view and the tracking of humans is imperfect, we use a Gumbel Social Transformer (GST), which provides unbiased modeling of partially detected humans, to predict the personal zones of humans [20]. As shown in Fig. 2b, the trajectory predictor takes the trajectories of observed humans from time  $t - M$  to  $t$  as input and predicts their future trajectories from time  $t + 1$  to  $t + K$ :

$$\hat{\mathbf{u}}_i^{t+1:t+K} = GST(\mathbf{u}_i^{t-M:t}), \quad i \in \{1, \dots, n\} \quad (3)$$

We concatenate the current states and the predicted future states of humans as one of the input observations of the robot policy network. To compute the intrusions in Eq. 1 for rewards, we add a circle centered at each predicted position to approximate personal zones as shown in Fig. 1b.

2) *Attention mechanisms*: The HH and RH attention functions are similar to the scaled dot-product attention in [21], which computes attention score using a query  $Q$  and a key  $K$ , and apply the normalized score to a value  $V$ .

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V \quad (4)$$

where  $d$  is the dimension of the queries and keys.

In HH attention, the current states and the predicted future states of humans are concatenated and passed through linear layers to obtain  $Q_{HH}^t, K_{HH}^t, V_{HH}^t \in \mathbb{R}^{n \times d_{HH}}$ , where  $d_{HH}$  is the attention size for the HH attention. We obtain the human embeddings  $v_{HH}^t \in \mathbb{R}^{n \times d_{HH}}$  using a multi-head scaled dot-product attention, and the number of attention heads is 8.

In RH attention,  $K_{RH}^t \in \mathbb{R}^{1 \times d_{RH}}$  is the linear embedding of the robot states  $\mathbf{w}^t$  and  $Q_{RH}^t, V_{RH}^t \in \mathbb{R}^{n \times d_{RH}}$  are linear embeddings of the weighted human features from HH attention  $v_{HH}^t$ . We compute the attention score from  $Q_{RH}^t$  and  $K_{RH}^t$  as in Eq. 4, transpose the score, and apply the score to  $V_{RH}^t$  to obtain the twice weighted human features  $v_{RH}^t \in \mathbb{R}^{1 \times d_{RH}}$  with a single head.

In HH and RH attention networks, we use binary masks that indicate the visibility of each human to prevent attention to invisible humans. Unlike DS-RNN that fills the invisible humans with dummy values [13], the masks provide unbiased gradients to the attention networks, which stabilizes and accelerates the training.

3) *GRU*: We embed the robot states  $\mathbf{w}^t$  with linear layers  $f_R$  to obtain  $v_R^t$ , which are concatenated with the twice weighted human features  $v_{RH}^t$  and fed into the GRU. Finally, the output of the GRU is input to a fully connected layer to obtain the value  $V(s_t)$  and the policy  $\pi(a_t|s_t)$ .

4) *Training*: We train the trajectory predictor with a dataset of human trajectories collected from our simulator. In RL training, we freeze the trainable parameters of the trajectory predictor and use Proximal Policy Optimization (PPO) for policy and value function learning [22]. We train trajectory predictor and RL policy separately because the two tasks have different objectives, resulting in unstable and less efficient joint training.

## III. EXPERIMENTS AND RESULTS

In this section, we present our simulation environment, experiment setup, and experimental results.

### A. Simulation environment

As shown in Fig. 3, in our 2D environment, the robot has a limited sensor range of  $5m$ . The maximum number of humans can reach up to 20. In each episode, the starting and goal positions of the robot and the humans are randomly chosen. To simulate a continuous human flow, humans will move to new random goals immediately after they arrive at their original goals. All humans are controlled by ORCA and react only to other humans but not to the robot. We use holonomic kinematics for each agent, whose action at time  $t$  consists of the desired velocity along the  $x$  and  $y$  axis,  $a_t = [v_x, v_y]$ . Our environment mimics complex real-world crowds with the following randomizations. First, all humans occasionally change their goal positions within an episode. Second, each human has random maximum speed and radius.

### B. Experiment setup

1) *Baselines and Ablation Models*: We denote our method as (GST, HH attn). We choose ORCA [6] and DS-RNN [13] as baselines<sup>1</sup>. In another baseline denoted as (Const vel, HH attn), we replace the GST predictor with a constant velocity predictor, which predicts the future trajectories by the latest velocity of the agent.

2) *Training and Evaluation*: For all RL-based methods, we use the same reward as defined in Eq. 2 and train them for  $2 \times 10^7$  timesteps. We test all methods with 500 random unseen test cases. Our metrics include navigation metrics and social metrics. The navigation metrics measure the quality of the navigation and include the percentage success rate (SR), average navigation time (NT) in seconds, and path length (PL) in meters of the successful episodes. The social metrics measure the social awareness of the agent, which include intrusion time ratio (ITR) and social distance during intrusions (SD) in meters. The intrusion time ratio per episode is defined as  $c/C$ , where  $c$  is the number of timesteps that the robot is in any human's ground truth social zone and  $C$  is the length of that episode. The ITR is the average

<sup>1</sup>Comparison to more baselines can be found in [23]

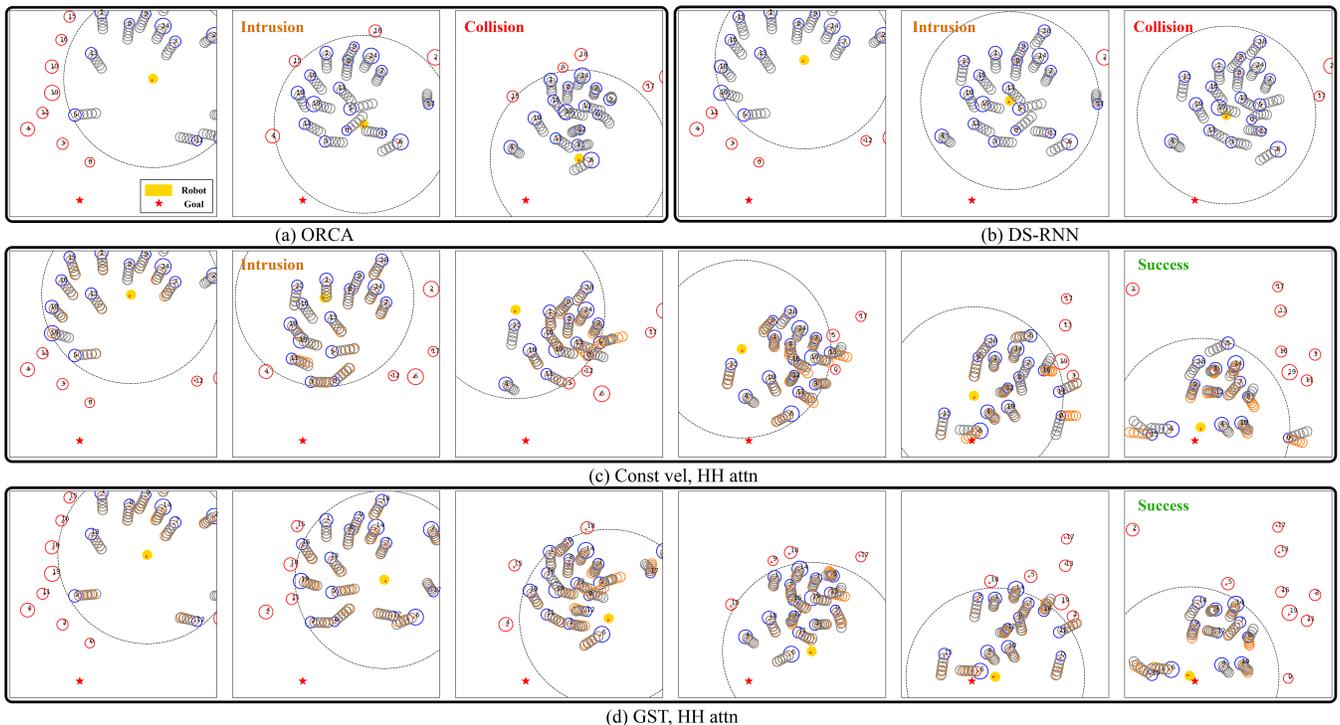


Fig. 3: Comparison of different methods in the same testing episode with randomized humans. The orientation of an agent is indicated by a red arrow, the robot is the yellow disk, and the robot’s goal is the red star. We outline the borders of the robot sensor range with dashed lines. Represented as empty circles, the humans in the robot’s field of view are blue and those outside are red. The ground truth future trajectories and personal zones are in gray and are only used to visualize intrusions, and the predicted trajectories are in orange.

ratio of all testing episodes. We define SD as the average distance between the robot and its closest human when an intrusion occurs. To ensure a fair comparison, all intrusions are calculated by ground truth future positions of humans.

TABLE I: Testing results

Method	SR $\uparrow$	NT $\downarrow$	PL $\downarrow$	ITR $\downarrow$	SD $\uparrow$
ORCA	69.0	14.77	17.67	19.61	0.38
DS-RNN	64.0	16.31	19.63	23.91	0.34
Ours (Const vel, HH attn)	87.0	14.03	20.14	7.00	0.42
Ours (GST, HH attn)	<b>89.0</b>	15.03	21.31	<b>4.18</b>	<b>0.44</b>

### C. Results

1) *Navigation performance*: From Table I, our method (GST, HH attn) outperforms ORCA and DS-RNN by a large margin in terms of success rate. Fig 3a, b, and d provide an example episode where (GST, HH attn) succeeds but ORCA and DS-RNN end up with collisions. Since the above two baselines only model RH interactions but ignore HH interactions, these results suggest that the human-human interactions are essential for dense crowd navigation and are successfully captured by our HH attention mechanism.

However, we also notice that in both scenarios, compared with constant velocity predictor, GST predictor only improves the success rate with small margins. The reason is that the humans in our simulator are controlled by ORCA, which prefers linear motion if no other agents are nearby.

2) *Social awareness*: From Table I, we observe that compared with ORCA and DS-RNN, the models aided by the trajectory predictors have significantly lower intrusion time ratio (ITR) and higher social distance during intrusions

(SD). Our method (GST, HH attn) has the lowest ITR and SD and therefore, the robot is the least invasive and most polite. For example, the robot in Fig 3d always keeps a good social distance from the personal zones of all humans, while it occasionally intrudes the personal zones in Fig 3c. In contrast, without predicted personal zones, the robots in Fig 3a and b are notably more aggressive and impolite. These results indicate that the proposed prediction-based personal zones accurately capture the personal space of walking pedestrians. Moreover, a better trajectory predictor approximates the personal zones more accurately, which leads to better policies.

3) *Real-world experiments*: We deploy the model trained in the simulator to a TurtleBot 2i. A video demonstration of both simulation and real world results can be found at [https://www.youtube.com/watch?v=p\\_asv42K18Q](https://www.youtube.com/watch?v=p_asv42K18Q).

## IV. CONCLUSION AND FUTURE WORK

We propose a novel learning framework for socially-aware crowd navigation. We redefine the personal zones of the walking humans with their long-term future trajectories, which are incorporated into an RL framework to prevent the intrusions of the robot. We capture the spatial interactions in the crowd with self-attention mechanisms and propose a novel graph neural network to learn navigation policies. Our method shows promising results. Possible directions to explore in future work include (1) using datasets collected from real pedestrians to train our method, and (2) performing user studies to evaluate the social awareness of our model.

## REFERENCES

- [1] J. K. Burgoon and S. B. Jones, "Toward a theory of personal space expectations and their violations," *Human Communication Research*, vol. 2, no. 2, pp. 131–146, 1976.
- [2] A. Bera, T. Randhavane, R. Prinja, and D. Manocha, "Sociosense: Robot navigation amongst pedestrians with social and psychological constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 7018–7025.
- [3] L. Takayama and C. Pantofaru, "Influences on proxemic behaviors in human-robot interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 5495–5502.
- [4] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [5] 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.
- [6] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer, 2011, pp. 3–19.
- [7] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.
- [8] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6015–6022.
- [9] J. Van Den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 1928–1935.
- [10] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.
- [11] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 285–292.
- [12] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3052–3059.
- [13] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, "Decentralized structural-rnn for robot crowd navigation with deep reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3517–3524.
- [14] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, "Relational graph learning for crowd navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10007 – 10013.
- [15] S. Eiffert, H. Kong, N. Pirmarzashti, and S. Sukkarieh, "Path planning in dynamic environments using generative rnns and monte carlo tree search," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10263–10269.
- [16] K. Li, M. Shan, K. Narula, S. Worrall, and E. Nebot, "Socially aware crowd navigation with multimodal pedestrian trajectory prediction for autonomous vehicles," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8.
- [17] A. J. Sathiamoorthy, J. Liang, U. Patel, T. Guan, R. Chandra, and D. Manocha, "Densecavoid: Real-time navigation in dense crowds using anticipatory behaviors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 11345–11352.
- [18] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 797–803.
- [19] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5308–5317.
- [20] Z. Huang, R. Li, K. Shin, and K. Driggs-Campbell, "Learning sparse interaction graphs of partially detected pedestrians for trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1198–1205, 2022.
- [21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.
- [22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [23] S. Liu, P. Chang, Z. Huang, N. Chakraborty, W. Liang, J. Geng, and K. Driggs-Campbell, "Socially aware robot crowd navigation with interaction graphs and human trajectory prediction," *arXiv:2203.01821*, 2022.