Learning Relation in Crowd Using Gated Graph Convolutional Networks for DRL-Based Robot Navigation

Haoge Jiang, Niraj Bhujel, Zhuoyi Lin, Kong-Wah Wan, Jun Li, Senthilnath Jayavelu, *Senior Member, IEEE*, and Xudong Jiang, *Fellow, IEEE*

***Abstract*— Deep reinforcement learning (DRL) frameworks have shown their remarkable effectiveness in learning navigation policy for the mobile robot navigating in a human crowded environment. Moreover, attention mechanisms coupled with DRL allows the robot to identify neighbors with different level of influence and incorporate them into the robot's decision. However, as the crowd density increases, attention mechanisms may fail to identify critical neighbors which can lead to significant drops in navigation efficiency. In this work, we aim to address this limitation by encoding both human-human and human-robot interaction using a special class of Graph Convolutional Networks (GCN) known as Message-Passing GCN (MP-GCN). In contrast to existing methods, where attention between robot and humans are encoded uniformly, the proposed approach named MP-GatedGCN-RL encodes asymmetric interactions using the combination of novel message-passing function and edge-wise gating mechanisms. We evaluate our approach on the simulated environments of ETH/UCY pedestrians datasets consisting of different scenarios like collision avoidance, group forming, diverging, crossing, and so on. Experimental results demonstrate that our proposed method outperforms the conventional benchmark dynamic avoidance method ORCA with a 20.6% increase in success rate and a 9.1% reduction in navigation time. Moreover, we also achieve a 5.5% enhancement in success rate compared to other state-of-the-art DRL-based methods without any additional labeled expert data nor prior supervised learning.**

***Index Terms*— Deep reinforcement learning, graph convolutional network, collision avoidance, motion and path planning, real-time autonomous navigation, autonomous unmanned vehicles.**

I. Introduction

MOBILE robot application environment will expand from static in-door to outdoor which is often crowded with humans. When a mobile robot has to share an environment with rational humans, safe and effective collision avoidance with socially-acceptable behavior becomes a crucial challenging aspect [1], [2]. As such, collision avoidance in a human crowded environment attracts lots of attention in the robot navigation community [3], [4], [5], [6]. Traditional approaches like DWA [3] and Reciprocal Velocity Obstacle (RVO) methods [4], [7], [8] consider all humans in the environments as static obstacles or make only one-step prediction to find the next optimal action of the robot. Another line of works [9], [10], [11], [12], [13] focus on forecasting pedestrian trajectories prior to planning the robot's movements. Unfortunately, as the crowd density increases, the possible movable space of the robot could be fully occupied by the expected pedestrian paths. Meanwhile, these methods are based on mathematical models, that may not always yield optimal solutions or even "freezing" when the number of humans in motion is very large, which may lead the agent to a dangerous situation [14].

To alleviate the above problems, recent works suggest modeling the robot and human states and their relations (interactions). As such, they applied a sequence to encode the interactions between an agent and humans, which only partially models the interactions or lacks spatial relation reasoning, i.e., they only consider explicitly modeling the robot-human interactions or using a sequence model to store the crowds' state [5], [15], [16].

Most recently, graph-based learning methods MP-RGL [17] and G-GCNRL [6] use Graph Convolutional Networks (GCN) [18] to explicitly model all interactions and spatial relationships among agents. These approaches, however, ignore the asymmetric interactions hidden in pedestrian trajectories and make an assumption that interactions between neighboring pedestrians are uniform i.e., two neighboring pedestrians have equal influence in both directions. Typically, the interactions between the agent and adjacent pedestrian are likely to have asymmetric relationships. For instance, an agent or pedestrian in the front will have a higher influence than the agent or pedestrian in the back and vice-versa [19]. Modeling such information is critical for identifying important neighbors in the crowd. Moreover, these methods are built on the Deep V-learning RL method [5], which necessitates the discretization of the robot's action space and traverses a discrete action space to find the action with the maximum value from its

Manuscript received 10 October 2022; revised 25 March 2023 and 23 September 2023; accepted 27 November 2023. This work was supported in part by the National Research Foundation, Singapore, under the National Research Foundation (NRF) Medium Sized Centre Scheme (CARTIN); and in part by Agency for Science, Technology and Research (A*STAR) from the National Robotics Programme (NRP), Singapore, under Grant 192 25 00049. The Associate Editor for this article was C. Sundarabalan. *(Corresponding author: Xudong Jiang.)*

Haoge Jiang and Xudong Jiang are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: haoge001@e.ntu.edu.sg; exdjiang@ntu.edu.sg).

Niraj Bhujel, Zhuoyi Lin, Kong-Wah Wan, Jun Li, and Senthilnath Jayavelu are with the Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: bhuj0001@e.ntu.edu.sg; zhuoyi.lin@outlook.com; kongwah@i2r.a-star.edu.sg; jli@i2r.a-star.edu.sg; J_Senthilnath@i2r.a-star.edu.sg).

Digital Object Identifier 10.1109/TITS.2023.3343923

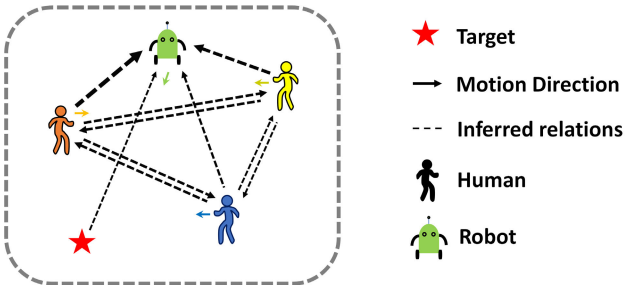


Fig. 1. An overview of the interaction model used in our approach. Varying levels of interactions between agents are encoded using directed edges between the nodes.

value network. As a result, the size of the discrete action space affects the final action's optimality, hence affecting the performance of the navigation policy.

To this end, we propose a bidirectional graph based Deep Reinforcement Learning (DRL) approach via an actor-critic framework for robot navigation in continuous action space in a human crowded environment. In particular, the environment state is represented by a graph where information about humans and robot are stored as node attributes. Then, the relation between the robot and humans is represented by constructing bidirectional edges on the graphs and stored as edge attributes. Fig. 1 illustrates how we explicitly represent the human-human and human-robot relationships and their asymmetric influence on each other. The thickness of the edges indicates the degree of influence between the agent and humans. In contrast to [6] and [17], we utilize a Message-Passing based GatedGCN (MP-GatedGCN) to learn the influence functions between humans and robot, where the asymmetric interactions are learned by an edge-wise gating mechanism [20] between the nodes of the graph. We propose a novel message passing function for updating the states of humans and robot. The message passing function for humans learns the neighbor influences, whereas for the robot, it learns the influences from the human neighbors and the robot's goal simultaneously. During message passing, the neighboring nodes will exchange their hidden states with each other which are then aggregated using a permutation invariant reduce function (i.e. max-pooling) to generate the final hidden state of the nodes during the reduction step. The node representation is subsequently leveraged by the DRL algorithm to generate continuous actions of each agent. More specifically, the learnable weights of MP-GatedGCN are shared among the agents which allows the robot node to generate a human-like navigation strategy by considering its own experience and its neighbors' states in the dense crowd.

The main contribution of this research work is outlined below:

- In this work, we investigate the importance of asymmetric influence between robot and dynamic humans on robot navigation by creating a novel human-human and human-robot interaction model using the Message-Passing Graph Convolutional Network.
- A novel navigation learning structure is proposed, which is optimized in continuous action space. This optimizes the robot navigation policy by searching for finer optimal action in the wider range of action space.
- The proposed method is evaluated on real-world crowd datasets consisting of various situations like collision avoidance, group avoidance, walking parallel, and so on. We demonstrate how the proposed edge-wise gating mechanism enables the robot to reach its destination while paying attention to its neighboring humans in a crowded environment.

II. Related Works

A. Crowd Navigation

The reaction-based method like RVO [4] and ORCA [7] solve collision avoidance problems by modeling other agents as dynamic obstacles to find optimal collision-free velocities under reciprocal assumptions. While these models perform adequately where all agents perform the same policy, they are suboptimal in real-world crowd scenes since the assumption of reciprocity could be violated and the pedestrians' motion is unpredictable and can be uncooperative. The performance can be significantly affected due to the invisible setting of the robot and the increased size of the crowd.

Another approach is using trajectory-based methods, to plan a feasible route for the robot before the motion by predicting other agents' intended paths [9], [10], [11], [12], [21]. Long-term decisions can be made by using the trajectory prediction results of the robot planner. The main problem is that searching for a feasible path online after estimating trajectory sequences can be time-consumed and computationally expensive so the real-time performance is a concern [22]. Even though it can be completely computed, all of the robot's available space could be taken up by predicted pedestrian paths, resulting in the robot being overly cautious [23].

Learning-based approaches have been presented recently as a result of the development of deep learning. Especially Deep Reinforcement Learning (DRL) has been found great success in robot continuous control and many other fields [24], [25], [26], [27], [28], [29], [30]. CADRL [15] initially described robot navigation as a sequential decision-making problem constrained by robot kinematics, and then used reinforcement learning to learn the joint state value function. However, it evaluates only the pairwise restrictions between robots and pedestrians and disregards the interactions between pedestrians. This type of end-to-end RL robot navigation strategy has also been utilized in [31], [32], and [33]. To navigate in situations of dynamically changing the number of pedestrians, [16] proposes utilizing LSTM to model the pedestrians' state as a sequence with fixed length. However, LSTM-RL assigns attention to surrounding humans according to distance, which is not always reasonable and consistent with human decision-making. A typical case pointed out in the related works [6] is that the humans within the perception range should bring higher attention than those out of the perception range even though the former is further away. Several self-attention models have been developed in recent years to model the interactions between pedestrians and robots, including LM-SARL [5] and SARL* [34]. However, these approaches only

model pedestrian interactions on a local map, which is just a partial representation of crowd interactions.

B. Graph Representation Learning

Graph convolutional networks (GCN) has shown great promise in learning hidden structure in complex data such as traffic networks [35], [36], social networks [37], physics [38], and medical diagnosis [39]. Graph neural networks learn the node representations by applying message passing mechanism [40] between graph nodes that aggregate information of local neighborhood [41], [42]. The fundamental structural characteristics of GCN makes it feasible to model spatial interaction between pedestrians and robots in a crowd [6], [17]. Pedestrians and the robot are modeled as graph nodes and their edges represent influences between them. The influence function is then learned through a learnable adjacency matrix [17] or through learnable edges [19]. Other factors like the human gaze can be used to learn the influence function between the nodes [6]. These methods, however, use the high-level graph representation of the crowd-interactions instead of fine-grained node-level representations for generating the action of the agents (nodes).

III. Proposed Approach

We first introduce how the agent decision-making in crowd is formulated as a reinforcement learning (RL) problem and then we will model the environment state as a bidirectional graph representation, which leads to our proposed network architecture.

A. Overview

The navigation problem in this work is for a mobile robot to navigate across an area with dynamic objects such as human crowded environments and complete its task to reach a given target. The state of the environment is given as $s_t = \{s_t^r, s_t^n\}, n \in 1 \dots N$, where s_t^r is the state of the robot and $s_t^n, n \in 1 \dots N$ are the states of N humans. At each step t, the state s_t of the environment is represented as a set of robot and human node state $x_t = \{x_t^r, x_t^n\}, n \in 1 \dots N$. In addition, an edge states $x_{i,j}^e$ is created that store relational information between agent i and agent j. Two linear embedding layers are used to map the robot node state and human nodes state to a fixed dimensional embedding $X = \{E_t^r, E_t^n\}$. Next, we create a bidirectional spatial graph G_t with node attributes X and edge attributes X_e on which message-passing is performed to extract hidden states $h_t = \{h_t^r, h_t^n\}$ of each agent. Finally, an MLP layer is used to generate the next action of each agent $a_t = \{a_t^r, a_t^n\}$ from h_t. The actions will transit the environment state into the next state s_{t+1} and the process is repeated for each episode. A reward R_t will be computed for each agent by the reward function described in Section III-A.3 based on the action a_t they executed. The experienced state transition of the environment will be stored in the prioritized experience reply buffer as a tuple T_r which includes graph representations, action and reward (G_t, a_t, G_{t+1}, R_t), which is used to train the actor-critic network. The episode will be terminated when the robot successfully finishes the navigation task, or the robot makes collisions or the maximum number of episode steps T is reached.

1) Action Space: Two motion types of action a_t are considered for holonomic robots and non-holonomic robots, one uses angular and linear velocities $[v, \omega]$ and the other uses linear velocities $[v_x, v_y]$. In this work, we select the holonomic model where $a_t = [v_x, v_y]$ for our wheel platform. In contrast to the existing V-learning based methods [5], [6], [15], [16], [17] that are optimized for discrete action space, the action a_t in our methods is optimized in a continuous action space within the range of $[-V_{max}, V_{max}]$.

2) State Graph Representation: The state of the environment with N humans and a robot is represented by a directed graph $G_t(V, E, X, X_e)$ with nodes $V = \{i \mid i \in \{1, \dots, N+1\}\}$ and edges $E = \{e_{i,j} \mid i, j \in \{1, \dots, N+1\}, i \neq j\}$. A node $i \in V$ is connected to node $j \in V$ via an undirected edge $e_{i,j} \in E$ if $\|p_i - p_j\|_2^2 \leq 5$, where p_i and p_j is the position of node i and node j respectively. The nodes attribute $X \in \mathbb{R}^6$ store state information of the particular node whereas the edge attributes $X_e \in \mathbb{R}^2$ store relational information between the nodes. Specifically, each edge attribute $x_{i,j}^e \in X_e$ is the difference between the real-world position between the neighbor nodes j and i.

Each element of the node attributes $x^i \in \mathbb{R}^4, i = 1 \dots N+1$ represents the position $\{p_x, p_y\}$ and velocity $\{v_x, v_y\}$. Since the action of a robot depends on its goal position, we added the robot's goal position $\{g_x, g_y\}$ to the attributes of the robot node $x^r \in \mathbb{R}^6$. Next, we utilize two linear projection layers, one for human nodes and the other for the robot node, to embed the node attributes to a fixed dimensional vector D. Additionally, we model the attractive influence from the robot's goal by adding a goal node to G_t with node attributes $x^g = \{0, 0, 0, 0, g_x, g_y\}$ and edge attributes $x_{r,g}^e = \{g_x - p_x, g_y - p_y\}$, where $\{p_x, p_y\}$ is the position of robot. The intuition is to facilitate the directional information going through the message passing to the robot's action during the node updates.

3) Reward function: To obtain consistent results, all the algorithms were given the same settings and parameters for the reward functions R_t. The reward function is composed of four components according to the configuration of [6]:

$$r_t = \begin{cases} r_{\text{fail}} & \text{if crashes or timeout} \\ r_{\text{reach}} & \text{if reaches the target} \\ r_{\text{danger}} & \text{if is too close to humans} \end{cases}$$

where r_{reach}, r_{fail}, and r_{danger} are defined as follows:

- r_{reach}= 1 is the positive reward for reaching the goal without any collision, otherwise 0.
- r_{fail}= -0.25 is the negative penalty for the robot if it collides with an obstacle, otherwise 0.
- r_{danger} is set to make the robot keep an appropriate safe distance from the detected obstacle(s). It is defined as:

$$r_{\text{danger}} = -0.1 + \frac{d_t}{2} \quad \text{if } (d_t < r_i) \tag{1}$$

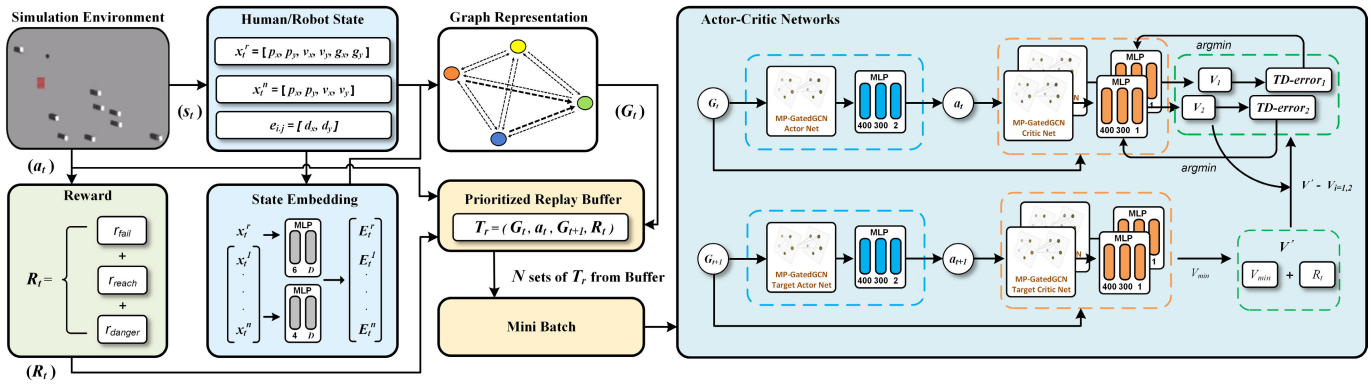


Fig. 2. An overview of our proposed approach. The agent's environment is converted into a graph representation on which message passing is performed to extract node-level state representation. The extracted representations are then fed into the Actor-Critic Networks (right box) to generate action action-value pairs a_t and V for each agent in the environment.

otherwise 0, where r_i is the danger layer, d_t is the instantaneous distance between the closet obstacle and the agent. When the agent moves into the danger layer of an obstacle, it will get a penalty, the closer the agent is to the pedestrian(s), the bigger penalty will be given. In this paper, the $r_i = 0.2$.

The total reward of the reward function R_t is the summation of all r_t cases as the below equation:

$$R_t = r_{\text{fail}} + r_{\text{reach}} + r_{\text{danger}} \tag{2}$$

4) Actor-Critic configuration: We use MP-GatedGCN in both actor and critic to model dynamic interactions between robot and humans. Specifically, the message passing mechanism, described in III-B, is used to encode both human-human and human-robot interactions. The hidden features of a node after l layers of message-passing have relevant information about the states of the neighboring nodes. The underlying spatial relationships between agents and their influences are encoded by the edge features. An aggregation operation is performed on the node and edge features which are used to generate action-value pairs for the robot. Specifically, the node and edge features are utilized for next layer node feature representation h_i^{l+1}, which is then fed to the actor network to generate the action a_t following its learnt policy $\pi(a_t|s_t)$. Afterward, the action a_t will be passed to the critic network to generate output $V(s_t)$ that can be regarded as the metric of evaluation. The actor-critic networks are trained to maximize the expected value function $V(s_t) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}[R(s_t, a_t)]$, where γ is a discount factor that reflects the preference of instant rewards over future rewards. In the training procedure, the DRL algorithm generates two copies of the neural network (online and target network) to make training more stable for the actor network (parametrized by θ^{π}, $\theta^{\pi'}$) and critic network (parametrized by θ^{V}, $\theta^{V'}$), noted that the target network will be updated by soft update after several steps update of the online network. Besides, a double Q learning technique is applied to prevent overestimation of the Q-value. More specifically, based on the online and target network framework, the DRL algorithm initializes two critic networks $V^{\theta_1}, V^{\theta_2}$ and an actor network π with their corresponding parameters $\theta^{V_1}, \theta^{V_2}, \theta^{\pi}$ respectively. And when calculating the Q-value from the critic network, the smaller Q-value between the two critic networks is utilized. The Q-value will be estimated more precisely in this way. A step-by-step training process is shown in Algorithm 1.

Algorithm 1 Algorithm to Train Our Method

Initialize actor network π and critic networks V^{θ_1}, V^{θ_2} with random parameters θ^{π}, θ^{V_1}, θ^{V_2}.
Initialize target networks $\theta^{V_1'} \leftarrow \theta^{V_1}$, $\theta^{V_2'} \leftarrow \theta^{V_2}$, $\theta^{\pi'} \leftarrow \theta^{\pi}$
Initialize replay buffer B with warm up steps M
Set mini-batch N
for t = 1 ... M **do**
  Build state graph G_t from observation s_t.
  Sample action $a_t \sim \pi(G_t) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma)$
  Get reward R_t and next state graph representation G_{t+1}
  Store transition tuple $(G_t, a_t, R_t, G_{t+1}, d)$ in B
  if $t >$ M **then**
    Sample N set of transitions $(G_t, a_t, R_t, G_{t+1}, d)$ from B
    Sample action $a_{t+1} \leftarrow \pi'(G_{t+1}) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma)$
    Estimate $V' = r + \gamma \min_{i=1,2,} V^{\theta_i'}(G_{t+1}, a_{t+1})$
    Update critic networks:
    $\theta^{V_i} \leftarrow \text{argmin}_{\theta^{V_i}} N^{-1} \sum \left(V' - V^{\theta_i}(G_t, a_t)\right)^2$
    if t mod d **then**
      Update θ^{π} by the deterministic policy gradient:
      $\nabla_{\theta^{\pi}} L \approx \mathbb{E}\left[\nabla_a V(G_t, a|\theta^{V})|_{a=\pi(G_t)} \nabla_{\theta^{\pi}} \pi(G_t)\right]$
      Update target networks:
      $\theta^{V_i'} \leftarrow \tau\theta^{V_i} + (1-\tau)\theta^{Vi'}$
      $\theta^{\pi'} \leftarrow \tau\theta^{\pi} + (1-\tau)\theta^{\pi'}$
    end if
  end if
end for

B. Message Passing GCN With Edge-Wise Gating Mechanism

The vanilla GCN considers the influence between neighboring nodes uniformly. This may not be optimal in the real world. Assuming a situation in which the robot is following the former pedestrian and a latter pedestrian is trailing the robot, the former pedestrian will have more influence on the robot than the latter pedestrian. The robot should find optimal action to avoid collision by paying more attention

to the former pedestrian. Hence, the weights from former pedestrians to the robot must be up-weighted, and the weights of the latter pedestrians need to be down-weighted due to its effect being less than the former one. Similarly, the weights of pedestrians in the opposite moving direction of the robot must be up-weighted, and the pedestrians that are in the same direction as the robot are down-weighted; up-weighted weights of pedestrians that are near the robot and down-weighted for those that are far away. In this way, the weights between human-human, and human-robot will be reasonably distinguished. Hence the crowd feature extraction ability is enhanced and we can direct the robot's attention to the more important pedestrian.

To this end, we adopt GatedGCN [20] that control the flow of information between the nodes through an edge-wise gating mechanism. GatedGCN has been found particularly suitable for modeling asymmetric interactions among pedestrians [19]. Other complex methods such as self-attention networks GAT [42] can also be used. But in our experiments, we didn't observe significant differences between GAT and GatedGCN [20] while the latter being faster. Since the GCN architecture of [19] is not specifically designed for robot behavior, we extend their architecture to take into account of the robot's goal during the update operation of the robot node. The node update operation at layer l for human and robot nodes is as follows where the robot node is specified with subscript i equals r:

$$h_i^{l+1} = h_i^l + \mathrm{ReLU}\left(U^l h_i^l + \sum_{j \in N_i} \eta_{ij}^l \odot V^l h_j^l\right) \tag{3}$$

$$h_r^{l+1} = h_r^l + \mathrm{ReLU}\left(U^l h_r^l + \sum_{j \in \{N_r, D_r\}} \eta_{rj}^l \odot V^l h_j^l\right) \tag{4}$$

where U^l, $V^l \in \mathbb{R}^{6\times 256}$ are the learnable matrices shared between humans and robot nodes, $\odot$ denotes the Hadamard product, ReLU denotes the non-linear activation function and D_r denote the node representing the destination of the robot. The term η_{ij}^l is a gating function that controls how much information from neighbor node j should be used for updating the hidden state of node i:

$$\eta_{ij}^l = \frac{\sigma\left(e_{ij}^l\right)}{\sum_{j' \in \mathcal{N}_i} \sigma\left(e_{ij'}^l\right) + \epsilon}, \quad \eta_{rj}^l = \frac{\sigma\left(e_{rj}^l\right)}{\sum_{j' \in \{\mathcal{N}_r, D_r\}} \sigma\left(e_{rj'}^l\right) + \epsilon} \tag{5}$$

where, σ is the standard sigmoid activation function, ϵ is a small fixed constant for numerical instability, and e_{ij}^l is the edge feature which is obtained by;

$$e_{ij}^l = e_{ij}^{l-1} + \mathrm{ReLU}\left(A^l h_i^{l-1} + B^l h_j^{l-1} + C^l e_{ij}^{l-1}\right) \tag{6}$$

$$e_{rj}^l = e_{rj}^{l-1} + \mathrm{ReLU}\left(A^l h_r^{l-1} + B^l h_r^{l-1} + C^l e_{rj}^{l-1}\right) \tag{7}$$

where $A^l, B^l \in \mathbb{R}^{D\times D}$, $C^l \in \mathbb{R}^{D\times D}$ are the learnable matrices. The relational edge features combined with the gating mechanism facilitate the learning of asymmetric influences among nodes with different levels of field of view (FOV).

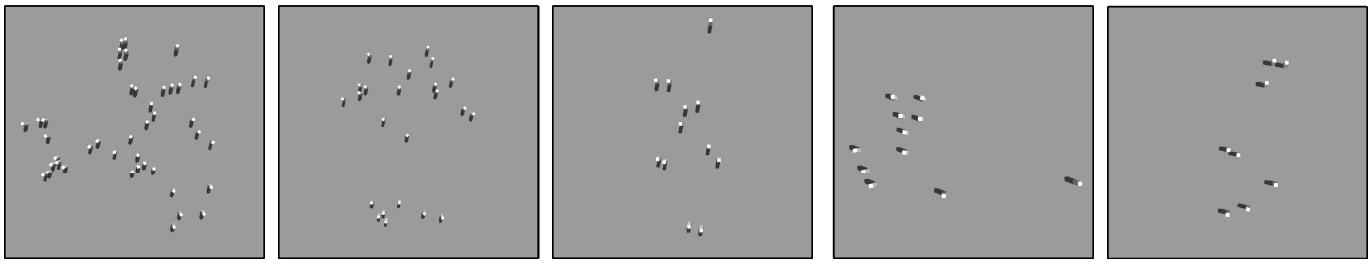
(a) Real-world scene (b) Pedestrian simulator

Fig. 3. A visualization of the pedestrian environment on our Gazebo simulator. Black, red and white color represents robot, robot's goal and pedestrians respectively.

IV. Simulation Environment

To train the policy and evaluate the performance of the approaches in the crowd scene, we build a pedestrian simulator base on the Gazebo platform [43] and utilize the Turtlebot3 robot as our evaluation wheel platform. The Gazebo pedestrian simulator with real-world pedestrian trajectories of datasets imported is shown in Fig.4. These figures demonstrate the density level in the simulator. More specifically, according to the pedestrian data from datasets, we let human models (square cylinders) be respawned and deleted on the ground plane to simulate the real-world scene (e.g. a shopping mall center or a shopping mall street) of human walking into a scene and leaving out from the scene. An example is shown in Fig.3(b). A video demo of our simulator can be viewed in the supplementary video materials (https://youtu.be/CsaVqC4YS-E). Every person with its specified ID has a full trajectory with its appearing time interval and time stamps. We load these data from datasets, spawn and delete cylinders that represent the humans frame by frame according to their time stamp. We will then re-load human dataset frames from scratch after rendering all data frames. The robot is updating at a fixed frequency where we used 0.25 seconds in the paper. As the keyframes' time interval of the original dataset is not the same as the robot, we use interpolate function and insert consecutive frames between every keyframe in their interval to make the trajectory smoother and be able to synchronize the executive time step of human and robot. The inserted frame node information is based on the last frame node information, where $[p_x, p_y]$ is calculated from interpolate function and $[v_x, v_y]$ is along the direction of the connected line from the last keyframe to the next keyframe.

UCY and ETH datasets consist of tracked video segments of pedestrian crowds recorded from an elevated bird-eye camera in five different environments. An example is shown in Fig.3(a). The environment constitutes different crowd levels and varying levels of interaction scenarios such as collision avoidance, crossing, merging, group formation, and so on. UCY dataset [44] includes four scenes: STUDENTS001 ($\sim$ 40 pedestrians/frame), STUDENTS003 ($\sim$ 26 pedestrians/frame), ZARA01 ($\sim$ 26 pedestrians/frame), and ZARA2



(a) Students001 (∼40 /frame) (b) Students003 (∼26 /frame) (c) Zara02 (∼7 /frame) (d) ETH (∼6 /frame) (e) Hotel (∼6 /frame)

Fig. 4. Visualization of crowd density on the simulated environment of ETH/UCY datasets at a particular frame.

TABLE I
HYPER PARAMETERS

Hyper-parameters	Value
Mini-batch size (N)	100
Buffer size (B)	100,000
Max steps/episode	1000
Max training steps (T)	1000,000
Warm up steps (M)	2000
Discount factor (γ)	0.99
Activation functions	ReLU
Number of GCN layers (L)	2
Dimension of hidden layers (D)	256
Learning rate	0.001

TABLE II
SIMULATION PARAMETERS

Simulation parameters	Value
Simulation platform	Gazebo
Robot model	Turtlebot3
Human model	Square Cylinder
Goal model	Red Square
Human model size	0.2×0.2 (m^2)
Goal model size	0.4×0.4 (m^2)
Robot model radius	0.2 (m)
Collision detection range	0.25 (m)
Datasets	UCY, ETH
Time step	0.25 (s)

(∼ 7 pedestrians/frame). ETH dataset [45] includes the HOTEL (∼ 6 pedestrians/frame) and ETH dataset (∼ 6 pedestrians/frame). The trajectories of all pedestrians are manually extracted every 0.4s i.e. 2.5Hz. We smoothen the trajectory by interpolating at 10Hz before feeding to the simulation environment.

The simulation is running on a server with Intel Core i9-9900k and NVIDIA GeForce RTX2080Ti. The training parameters and simulation parameters are shown in Table I and Table II respectively.

V. SIMULATION RESULTS AND ANALYSIS

This section compares the proposed approach to other methods. The generated pedestrian information is from the aforementioned real-world human trajectory datasets, the STUDENTS001 datasets will be used for policy training, and the remained datasets are used for evaluation. In the training process, the starting point is randomly generated on a 4 m radius circle, with the circle's center corresponding to the distribution center of pedestrians. The goal and starting point are symmetrical concerning the center of the circle. So the total length of navigation will be 8 m. Each policy is trained for 1×10^6 timesteps in the training phase. In the evaluation process, the reward function with the same setting for all evaluated approaches will be used to maintain consistency. The start point and goal point in evaluation follow the rule described in training, a total length of 8 m with a fixed start and goal point is set in each scene. To fully show the method's efficacy, the robot in the simulated situation is made invisible to pedestrians. So the robot will have no effect on the pedestrian's behavior which means the pedestrians will not perceive the robot and hence do not avoid it. Besides, this can alleviate the problem that the robot makes an aggressive invade policy, which is the situation where the robot invades human social space and tries to force pedestrians to avoid it. To do the statistics, each dataset is evaluated more than 250 times.

A. Comparison With ORCA

The conventional dynamic avoidance method ORCA based on the optimization [7] and under the reciprocity assumption, performs well in multi-agent navigation scenarios in which all the agents execute the same ORCA policy. However, it requires hand-tuned parameters and an optimization function. In addition, ORCA presupposes that agents would cooperate for half of the necessary distance for avoidance while pedestrian policy in real crowd scenes can be affected by various unknown reasons, and hence the reciprocity assumption may violate, resulting in a bad performance. We evaluate the ORCA and RL models by success rate and navigation time in the datasets with different densities. As indicated in Table III, RL approaches have a great improvement in the success rate and navigation time as compared to ORCA. Particularly in the scenario of the STUDENTS003 dataset, the navigation success rate reaches around 0.8 thanks to the attention weight extracted by our MP-GatedGCN. In addition, the prolonged navigation time demonstrates that ORCA is too cautious in the thick crowd.

B. Comparison With the State-of-the-Art Methods

We first compare our proposed model with the State-of-the-Art RL method G-GCNRL [6]. Note that the result of

TABLE III
QUANTITATIVE COMPARISON OF SUCCESS RATE AND NAVIGATION TIME WITH THE STATE-OF-THE-ART METHOD

Methods	Success Rate (%)					Navigation Time (s)				
	STUDENTS003	ZARA2	HOTEL	ETH	**AVG**	STUDENTS003	ZARA2	HOTEL	ETH	**AVG**
ORCA	0.566	0.623	0.618	0.702	0.627	12.9	12.2	12.3	11.5	12.2
G-GCNRL [6]	0.753	0.936	0.703	0.831	0.806	11.2	10.9	11.1	11.1	11.1
U-GCNRL [6]	0.671	0.928	0.687	0.827	0.778	11.2	**10.1**	**9.3**	**10.2**	**10.2**
MP-GatedGCN-RL	**0.788**	**0.941**	**0.723**	**0.878**	**0.833**	**11.1**	11.0	10.9	11.1	11.0

TABLE IV
COMPARISON OF THE PROPOSED METHOD AGAINST OUR BASELINE MODELS

Methods	Success Rate (%)					Navigation Time (s)				
	STUDENTS003	ZARA2	HOTEL	ETH	**AVG**	STUDENTS003	ZARA2	HOTEL	ETH	**AVG**
DNN-RL	0.611	0.803	0.624	0.692	0.683	12.8	12.4	12.2	12.9	12.6
GCN-RL	0.697	0.917	0.696	0.829	0.785	11.9	**10.4**	**10.2**	**10.3**	**10.7**
MP-GatedGCN-RL	**0.788**	**0.941**	**0.723**	**0.878**	**0.833**	**11.1**	11.0	10.9	11.1	11.0

G-GCNRL is from the original paper since the source code is not released. We attempt to reproduce the experimental environment similar to that of the original paper as much as possible in our gazebo simulator (e.g. trajectory datasets, robot prefer velocity, robot state representation, etc.) In contrast to G-GCNRL which uses all pedestrians in the environment, we use only pedestrians that are within a radius of 5 meters from the robot position, as the environment in the real world is rarely able to be fully observed. The models are compared using two criteria: success rate and navigation time. Table III shows the experimental results for four datasets. The success rate of our approach is greater than G-GCNRL and higher than 6% of the model U-GCNRL.

In the STUDENTS003 dataset, our MP-GatedGCN model outperforms the U-GCNRL and G-GCNRL in terms of success rates and navigation time. However, the U-GCNRL model performs better in navigation time in the ZARA2, HOTEL, and ETH datasets, and also gets a satisfactory result in success rate. We can notice that these datasets have a common characteristic that is with low density-level (under 10 pedestrians per frame). The possible explanation is that when crowds' density is sparse (lower than 8 pedestrians per frame), the specific attention mechanism is less important. It shortens navigation time by showing less focus on the crowd's attention weights. As for crowd navigation in the STUDENTS003 dataset (density of over 26 pedestrians per frame), our method has the best success rate and the shortest navigation time among all methods. Even though the improvement is slight, there is one thing worth noting that we do not need additional information such as human gaze expert data. The G-GCNRL paper describes that the human gaze expert data is extracted by the Tobii Pro X60 remote eye tracker. And the way expert provides the human gaze data is similar to work that collects gaze data while human play the Atari games [46]. So it means that the human gaze data collection requires an expert to have a full observation view of the environment (a top view of the fully observed environment), which is very difficult to build in the real world only by local sensors in the robot, so this work is hard to extend to real-world experiment. What's more, comparing the G-GCNRL with U-GCNRL in table III, we can also notice that the improvement of G-GCNRL is mainly on the dataset STUDENTS003, the most density environment, which indicates the importance of distributing correct attention to the proper subject in the crowd environment. In contrast, our method uses a gate mechanism and considers the asymmetric influence of nodes. The attention is obtained from end-to-end training, with no need for additional information. This demonstrates that the attention extracted by our data-driven method, MP-GatedGCN with gate mechanism, is more promising than the method using the supervised learning method G-GCNRL, which supervised learns the attention by expert human gaze data.

C. Ablation Studies

We generate an ablation model that modeled the state as a DNN 1-D sequence representation and named this model DNN-RL. This model concatenates the pedestrian state after the robot state. The number of surrounding pedestrians will be fixed at 40, so the sequence will be a vector with a fixed length of (1+40) x 6. We also implemented a GCN-RL model in our environment that uses graph representation to model the state. Besides, uniform attention is applied to all sensed pedestrian nodes (5 meters) in this model, so the structure and function of the GCN-RL are similar to the U-GCNRL. From the result in Tables III and IV, we can see that the performance of U-GCNRL and GCN-RL only have a slight difference. The navigation time in the different kinds of density presents a similar trend and character as well. However, U-GCNRL gets a slightly lower success rate. The reason is that the U-GCNRL work is highly based on Deep V-learning code from [5] and the Deep V-learning method uses the discrete action space which divides action space into 5 samples. We use the continuous action space thus providing a wider range for searching for the optimal action, resulting in better navigation performance.

The DNN-RL model performs poorly in our evaluation as it is unable to capture spatial relationships in state modeling. Since the state vector has a fixed length in a sparse

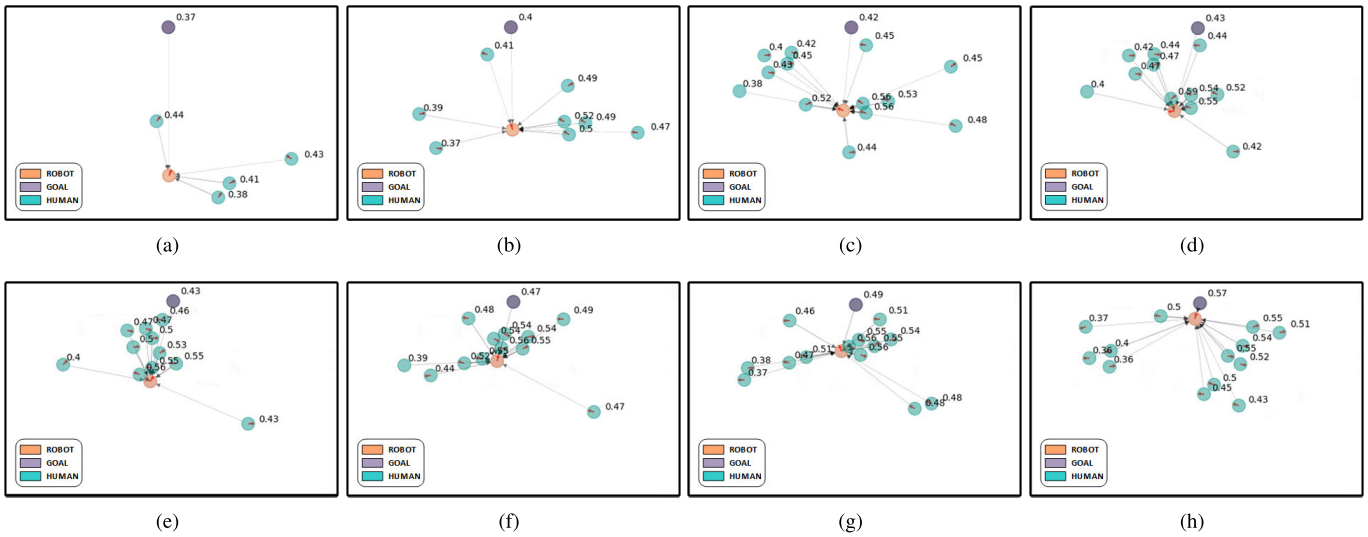


Fig. 5. Visualization of attention weights. The direction of influences from the neighboring pedestrians on the robot is shown by the black arrows. The number represents the degree of influence by each pedestrian on the robot i.e. attention of the robot to the corresponding pedestrians. Short red arrows represent the direction of the corresponding agent.

TABLE V
DANGERS/EPISODE METRIC IN ABLATION STUDY

Methods	Dangers / Episode				
	STUDENTS003	ZARA2	HOTEL	ETH	**AVG**
DNN-RL	1.536	0.788	1.688	1.884	1.474
GCN-RL	1.372	0.472	1.336	0.992	1.043
MP-GatedGCN-RL	**0.808**	**0.368**	**0.948**	**0.776**	**0.725**

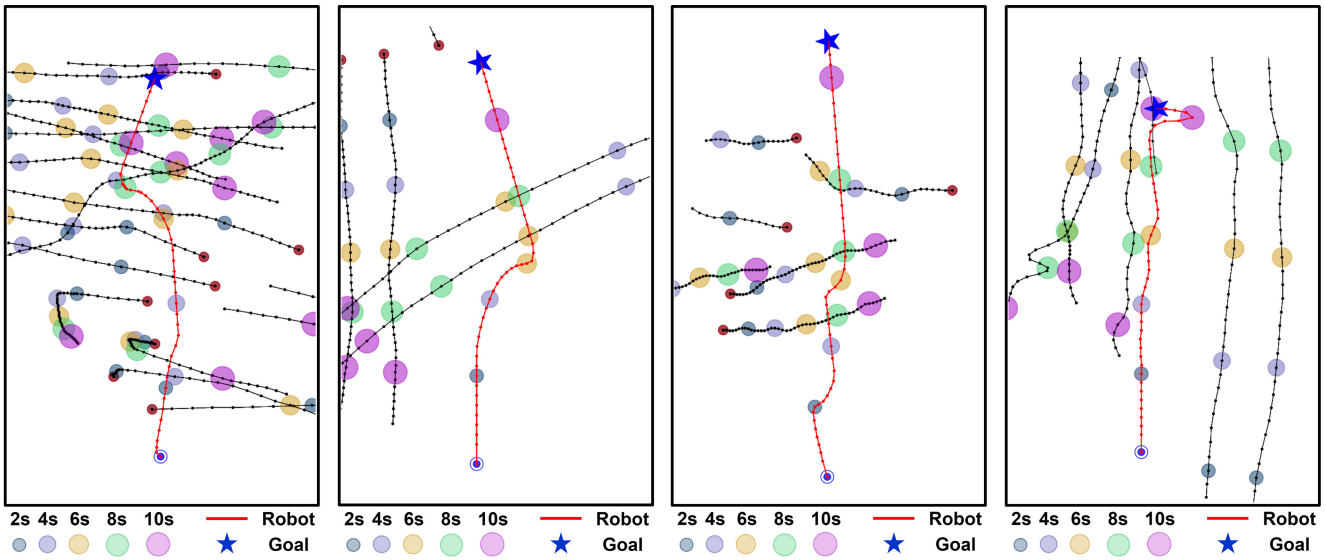
environment, the pedestrians' information will be none or 0 in the sequence. As a result, its performance is not stable across datasets. In addition, we observe a significant decrease in DNN-RL performance in the extreme crowd scene (STUDENTS003 dataset). This is because the robot is unable to find the most appropriate action to perform collision avoidance. After all, it does not generate proper attention distribution for the surrounding pedestrians.

We also introduce a new metric named dangers per episode to evaluate the effectiveness of our method that considers the asymmetric influence of humans to the robot. Once the robot moves close to a human with a distance lower than 0.2 meters, it will be counted as one instance of danger. The metric is computed by the total number of danger counts divided by the total number of episodes. It evaluates whether the robot learns safe and efficient behavior to avoid obstacles while moving to the target. From Table V we can see that it is helpful for the robot to learn the optimized actions by constructing the spatial relationship of the crowd. Compared to the DNN-RL model and the GCN-RL model, the proposed MP-GatedGCN-RL model performs better with less number of danger cases generated. It is worth noting that the best performance of this metric achieved by our method demonstrates the importance of considering the asymmetric influence between agents.

D. Attention Visualization

In order to analyze how robot attend to human changes, we show the attention values learned by the model using Eq.(5), and this process is illustrated in Fig.5. The gating value η_{ij}^{l} plays a role of a soft attention mechanism [47], manifesting different magnitude of attention values to the human/goal objects. Elevated attention values indicate a more pronounced influence of the human/goal object on the robot. For better visualization, we only display the attention between human-robot edges because we are using the fixed trajectories from datasets and thus the attention between humans is unnecessary. Note that the attention between human-human is used for the robot to learn the influence function during training, we only make it invisible in visualization. As shown in the figures of Fig.5, the influence exerted by neighboring pedestrians on the robot is illustrated by the black arrows. The numerical values indicate the extent to which each pedestrian affects the robot, signifying the robot's attention towards the respective pedestrians/goal. Meanwhile, short red arrows depict the direction in which the robot is headed.

From the illustration of figures, the robot's priority of attention weights changes over the state. The robot initially concentrates on the left top approaching pedestrians since they are with the same moving trend and are possible to collide, so the highest priority attention weight is distributed in Fig.5 (a). Then after several steps in Fig.5 (b), the front pedestrian passed and a group approached humans from the right side. The attention on the right side increases due to the potential collision risk. The robot then turns a left angle to avoid colliding right with the humans. After that, with the pedestrians approaching closer, the attention weights of the right-side group of humans become higher accordingly in Figs.5 (c-d). Then the crowd on the left and right sides converged in Figs.5 (e-f) over time, severely impeding the robot's forward path. The robot precisely concentrates on the most critical pedestrian with correct attention priority and takes optimal action to avoid collisions. After moderately navigating out of the crowd, the robot's attention shifted to the goal, the goal's attention priority increased, and then the robot proceeded directly there in Figs.5 (g-h).



(a) STUDENTS003 (b) ZARA02 (c) HOTEL (d) ETH

Fig. 6. Visualization of robot trajectories during the evaluation phase. The sizes of the circles represent the timestamps.

It is worth noting that in Figs.5 (c-d), the converging crowd that tends to block the forward way of the robot may lead to a "freezing" problem. However, our method finer distinguishes the attention and the robot always focuses on the most critical human in the crowd. So the robot still takes the left action even though there will be another potential collision risk with the left human. After that, the attention weight on this human increase as the left human moves closer. The robot then changes the collision avoidance priority and prioritizes the calculation of the collision-free action for this human. So our method is more adaptive in extremely crowded environments than previous works and prevents the robot trapped in "freezing" danger.

As previously demonstrated, the attention mechanism can assist the robot in identifying targets that require higher attention during navigation. Especially in crowded situations, the robot with assistance from the attention mechanism in alleviating computational load and avoids potential risks.

E. Trajectories Visualization

Fig.6 demonstrates the trajectories that our method generated in the evaluation datasets. The circles of different sizes and colors correspond to the various timestamps. The robot's path is denoted by a red line, while the start point and destination are represented by a blue outer circle and star, respectively. In sparse environments like ETH, HOTEL, and ZARA2 datasets, our method successfully avoids approaching pedestrians and reaches the goal while generating a smooth and straight path for the robot. In the dense environment STUDENTS003 dataset, while successfully performing the navigation tasks, the path taken by our method indicates that even in the most congested area, where the timestamp is between 6 and 10 seconds, the robot continues to move smoothly and does not pause in place for additional hesitation. More simulation cases can be found in the supplementary video materials (https://youtu.be/CsaVqC4YS-E).

VI. Discussion

Similar to other deep learning approaches, the current model has some limitations, such as the inability to decelerate or accelerate abruptly to avoid colliding with pedestrians suddenly arising from the corner or suddenly deviating from their normal path. In addition, in the simulation environment, the robot's motion seems to perfectly synchronize with the human movement, whereas in the real datasets, we have observed that its motion is quite slow compared with pedestrians, which may result in the robot being kicked off by the pedestrians. Besides, we infer the pedestrians' intention using only the past observations of pedestrian position and velocity. This may not be enough for an accurate prediction of future intention. Other factors like the hidden intentions of humans and sudden changes of decisions are not captured in observation history. Thus, creating robot behavior to adapt to pedestrian preferences and behaviors is still an open challenge that is worthwhile to address in future works.

VII. Conclusion

In this work, we propose a crowd navigation policy with relation graph learning and investigate the importance of asymmetric influence between dynamic humans and robot. In the state processing stage, the human/robot states are represented by a bidirectional graph, with their spatial-temporal interactions stored as the node and edge attributes. Then a MP-GatedGCN model learns the asymmetric influence of the robot and pedestrians by using an edge-wise gating mechanism between the nodes of the graph to distribute proper attention to the adjacent pedestrians. This attention is finally utilized for the DRL network to optimize the robot's action to derive

safe and efficient navigation. We show that our model outperforms the baselines and illustrates its performance on the capability to handle the challenging crowd navigation scenes on real-world pedestrian trajectory datasets with varying crowd densities.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

REFERENCES

[1] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1726–1743, Dec. 2013.

[2] P. Du et al., "Online monitoring for safe pedestrian-vehicle interactions," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–8.

[3] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[4] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1928–1935.

[5] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6015–6022.

[6] Y. Chen, C. Liu, B. E. Shi, and M. Liu, "Robot navigation in crowds by graph convolutional networks with attention learned from human gaze," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2754–2761, Apr. 2020.

[7] J. V. D. Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*. Berlin, Germany: Springer, 2011 pp. 3–19.

[8] J. Snape, J. V. D. Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 696–706, Aug. 2011.

[9] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Auto. Robots*, vol. 35, no. 1, pp. 51–76, Jul. 2013.

[10] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1289–1307, Sep. 2016.

[11] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: The case for cooperation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 2153–2160.

[12] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation," *Robot., Sci. Syst.*, vol. 8, pp. 193–200, Jul. 2012.

[13] V. V. Unhelkar, C. Pérez-D'Arpino, L. Stirling, and J. A. Shah, "Human–robot co-navigation using anticipatory indicators of human walking motion," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 6183–6190.

[14] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 797–803.

[15] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 285–292.

[16] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3052–3059.

[17] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, "Relational graph learning for crowd navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10007–10013.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[19] N. Bhujel, Y. W. Yun, H. Wang, and V. P. Dwivedi, "Self-critical learning of influencing factors for trajectory prediction using gated graph convolutional network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 7904–7910.

[20] X. Bresson and T. Laurent, "Residual gated graph ConvNets," 2017, *arXiv:1711.07553*.

[21] C. Cao, P. Trautman, and S. Iba, "Dynamic channel: A planning framework for crowd navigation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5551–5557.

[22] K. Driggs-Campbell, V. Govindarajan, and R. Bajcsy, "Integrating intuitive driver models in autonomous planning for interactive maneuvers," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3461–3472, Dec. 2017.

[23] K. Driggs-Campbell, R. Dong, and R. Bajcsy, "Robust, informative human-in-the-loop predictions via empirical reachable sets," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 3, pp. 300–309, Sep. 2018.

[24] V. Mnih et al., "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[25] H. Jiang, H. Wang, W.-Y. Yau, and K.-W. Wan, "A brief survey: Deep reinforcement learning in mobile robot navigation," in *Proc. 15th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Nov. 2020, pp. 592–597.

[26] K. Wu, H. Wang, M. Abolfazli Esfahani, and S. Yuan, "BND-DDQN: Learn to steer autonomously through deep reinforcement learning," *IEEE Trans. Cognit. Develop. Syst.*, vol. 13, no. 2, pp. 249–261, Jun. 2021.

[27] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1343–1350.

[28] H. Jiang et al., "ITD3-CLN: Learn to navigate in dynamic scene through deep reinforcement learning," *Neurocomputing*, vol. 503, pp. 118–128, Sep. 2022.

[29] H. Zhang, J. Cheng, L. Zhang, Y. Li, and W. Zhang, "H2GNN: Hierarchical-hops graph neural networks for multi-robot exploration in unknown environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3435–3442, Apr. 2022.

[30] Z. Rao et al., "Visual navigation with multiple goals based on deep reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5445–5455, Dec. 2021.

[31] P. Long, W. Liu, and J. Pan, "Deep-learned collision avoidance policy for distributed multiagent navigation," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 656–663, Apr. 2017.

[32] J. Jin, N. M. Nguyen, N. Sakib, D. Graves, H. Yao, and M. Jagersand, "Mapless navigation among dynamics with Social-safety-awareness: A reinforcement learning approach from 2D laser scans," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 6979–6985.

[33] H. Shi, L. Shi, M. Xu, and K.-S. Hwang, "End-to-end navigation strategy with deep reinforcement learning for mobile robots," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2393–2402, Apr. 2020.

[34] K. Li, Y. Xu, J. Wang, and M. Q.-H. Meng, "SARL: Deep reinforcement learning based human-aware navigation for mobile robot in indoor environments," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 688–694.

[35] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*.

[36] W. Zhang, H. Liu, Y. Liu, J. Zhou, and H. Xiong, "Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 1186–1193.

[37] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," 2019, *arXiv:1902.06673*.

[38] V. Bapst et al., "Unveiling the predictive power of static structure in glassy systems," *Nature Phys.*, vol. 16, no. 4, pp. 448–454, Apr. 2020.

[39] N. Xu, P. Wang, L. Chen, J. Tao, and J. Zhao, "MR-GNN: Multi-resolution and dual graph neural network for predicting structured entity interactions," 2019, *arXiv:1905.09558*.

[40] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.

[41] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[42] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *Stat*, vol. 1050, p. 20, Oct. 2017.

[43] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2004, pp. 2149–2154.
[44] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 655–664, 2007.
[45] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 261–268.
[46] R. Zhang et al., "AGIL: Learning attention from human for visuomotor tasks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 663–679.
[47] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

Haoge Jiang received the B.S. degree in information and telecommunications engineering from Ming Chuan University, Taiwan, in 2019. He is currently pursuing the Ph.D. degree with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research focuses on the intersection of deep learning, reinforcement learning, autonomous navigation, and robotics.

Niraj Bhujel received the B.E. degree in electronics and communication engineering from Pokhara University, Nepal, and the Ph.D. degree in electrical and electronics engineering from Nanyang Technological University, Singapore, in July 2022. He is currently pursuing the Master of Science degree in communication networks with the Asian Institute of Technology, Thailand. He is also a Research Scientist with the Institute for Infocomm Research (I^2R), A*STAR. He has published his research works in multiple human tracking and path predictions in respected journals and conferences, such as RAL and IROS. His research focuses lie in the intersection of deep learning, robotics, and computer vision. He was a recipient of the A*STAR Singapore International Graduate Award. He was a recipient of the Asian Development Bank Scholarships (ADB-JSP).

Zhuoyi Lin received the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2022. Currently, he is a Scientist with the Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore. His research has been published in leading conferences and journals in related domains (e.g., IJCAI, ICDM, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS). His research interests include machine learning and data mining and their applications.

Kong-Wah Wan received the B.Sc. and M.Sc. degrees from the National University of Singapore and the Ph.D. degree from Nanyang Technological University, Singapore. A Principal Scientist with the Institute for Infocomm Research (I^2R), he is currently heading the Perception Laboratory, Robotics Department. He has authored numerous patents and technical publications in these fields, including best paper awards in international scientific conferences. His research interests include video/image analysis, machine learning, and robotics navigation. He is also actively collaborating with the local industry and government agencies to deploy technologies, such as robust ID recognition, robotic conveying systems, and visual navigation for UAVs. His accolade include the PST Distinguished ExCEL Innovation Project in 2018 for Project "Automated Passenger In-Car Clearance System," the GovInsider "Best Drone and Robotics Project" Award in 2018 for Project "Outdoor Robotic Surveillance Systems," the MHA Firefly Silver Award in 2021 for Project MATAR, and the Dunman Partnership Award (Silver) in 2023.

Jun Li received the B.S. and M.Eng. degrees from the University of Science and Technology of China (USTC) in 1997 and 2002, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2007. He is currently a Senior Scientist with the Institute for Infocomm Research (I^2R), Singapore. His research interests include biometrics/soft biometrics, AI-enabled robotic perception, and cognitive navigation without precise maps. He has authored a number of high quality papers in these areas. One of these papers was featured on the cover of *Nature Machine Intelligence* in June 2022, which he has coauthored for the contribution of perception modules to the detection and analysis of unseen tools. He has also translated his research into commercial solutions and granted several software licenses and patents.

Senthilnath Jayavelu (Senior Member, IEEE) received the Ph.D. degree in aerospace engineering from the Indian Institute of Science (IISc), India. He is currently a Senior Scientist with the Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore. He has published over 100 high-quality papers and won five best paper awards. His current research interests include artificial intelligence, multi-agent systems, generative models, online learning, reinforcement learning, and optimization. He is a member of Artificial Intelligence, Analytics and Informatics (AI3), A*STAR. He has been serving as a Guest Editor/organizing chair/co-chair/PC member in leading AI and data analytics journals and conferences.

Xudong Jiang (Fellow, IEEE) received the B.E. and M.E. degrees from the University of Electronic Science and Technology of China (UESTC) and the Ph.D. degree from Helmut Schmidt University, Hamburg, Germany. From 1986 to 1993, he was a Lecturer with UESTC. From 1998 to 2004, he was with the Institute for Infocomm Research, A*STAR, Singapore, as a Lead Scientist and the Head of the Biometrics Laboratory. He joined Nanyang Technological University (NTU), Singapore, as a Faculty Member in 2004, where he was the Director of the Centre for Information Security from 2005 to 2011. He is currently a Professor with the School of EEE, NTU, and also the Director of the Centre for Information Sciences and Systems. He has authored over 200 articles with over 50 articles in IEEE journals, including nine articles in IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and 18 articles in IEEE TRANSACTIONS ON IMAGE PROCESSING. His current research interests include pattern recognition, computer vision, machine learning, image processing, and biometrics. He served as an IFS TC Member for IEEE Signal Processing Society and an Associate Editor for IEEE SIGNAL PROCESSING LETTERS and IEEE TRANSACTIONS ON IMAGE PROCESSING. He serves as a Senior Area Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING and the Editor-in-Chief for *IET Biometrics*.