CONFIDENTIAL. Limited circulation. For review only.

Towards Safe Robot Crowd Navigation by Simultaneously Considering Occlusion and Interaction

***Abstract*— Robot navigation in dense crowds represents a critical challenge in robotics, requiring autonomous robots to safely traverse populated environments without collisions. However, most existing approaches assume that the robot could directly access the dynamics of surrounding humans. This assumption is unrealistic in practical scenarios, where occlusions frequently prevent the robot from observing all nearby individuals. Furthermore, effectively understanding the subtle interactions between the robot and humans is essential for improving collision avoidance performance. To address these limitations, we propose a novel navigation framework that simultaneously considers both occlusion and varied interactions. First, we design an occlusion inference module that utilizes a variational autoencoder (VAE) to implicitly infer the states of occluded regions. Following this, an interaction learning module is constructed, wherein attention mechanism is employed to capture the complex interactions within crowd environments. Finally, we leverage an on-policy reinforcement learning algorithm with fused occlusion and interaction representations to train a safe and efficient crowd navigation policy. For comprehensive evaluation, we benchmark our framework against representative approaches in challenging crowd environments. The results demonstrate that our method achieves superior performance over existing methods.**

I. Introduction

With the rapid advancement of artificial intelligence, robots are increasingly integrated into human daily routines. Robotic applications are pervasive across diverse domains, such as autonomous vehicles on urban road [1], service robots in domestic environment [2][3], and delivery robots in commercial settings like restaurant [4]. When deploying robots in realistic environments, it has become very important to enable them to navigate in crowded spaces both safely and smoothly.

Extensive research has been conducted on crowd navigation. Originally, reaction-based methods define one-step interaction rules according to the current geometric configuration. For instance, the Optimal Reciprocal Collision Avoidance (ORCA) [5] modifies each agent's velocity vector to guarantee collision-free navigation. Furthermore, trajectory-based methods predict the motions of other agents and explicitly model the evolution of joint future states between an agent and its neighbors. However, these approaches may suffer from the "freezing robot problem": in densely populated environments, planners often classify all paths as unsafe, causing the robot to halt. This outcome is suboptimal, as feasible paths typically remain available. With the rapid advancement of machine learning, several deep reinforcement learning (DRL) based approaches [6][7][8] have been proposed. Nevertheless, these methods are generally trained in simulator under the unrealistic assumption of privileged access to complete environmental knowledge, such as the exact positions and velocities of all humans. This assumption overlooks the occlusions that frequently occur in practical settings, where humans hidden in occluded regions remain unobservable, as illustrated in Fig. 1. As a result, a navigation policy trained under such assumptions exposes robots to the risk of colliding with occluded humans once deployed in real-world environments. Inspired by the "People as Sensors" paradigm in autonomous driving, Pas-VAE [9] leverages occupancy grid map (OGM) to estimate the states of occluded areas while employing a variational autoencoder (VAE) to encode interactions among human agents. However, as the number of humans grows and the environment becomes more complex, Pas-VAE struggles to capture highly complex interactions between robot and crowds, ultimately leading to performance degradation.

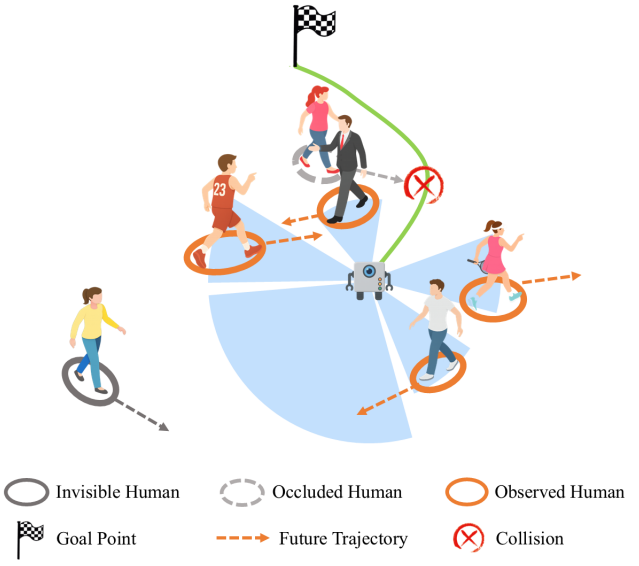


Fig. 1. Illustration of a crowd navigation scenario. Since the robot considers only currently observed humans during navigation, its planned path (green) may conflict with occluded humans, potentially leading to collisions.

Motivated by these challenges, we propose an end-to-end navigation framework that simultaneously incorporates occlusion reasoning and interaction representation for robot decision making in crowds. To better approximate practical conditions during training, we modify the simulation environment to explicitly model occlusion phenomena. We first develop an occlusion inference module that utilizes a variational autoencoder (VAE) leveraging historical partially observed occupancy grid maps to implicitly perform occlu-

Manuscript 2076 submitted to 2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 2, 2026.

sion inference. Within the constructed interaction learning module, two dedicated attention networks are employed to separately extract human–human interactions (HHI) and human–robot interactions (HRI). Subsequently, interaction features and occlusion representations are fused and fed into an on-policy reinforcement learning (RL) network to enable more informed decision-making.

The main contributions of this paper are summarized as follows:

- We propose a crowd navigation framework that simultaneously accounts for partial occlusion and complex interactions.
- We develop an occlusion inference method that directly utilizes historical partially observed occupancy grid maps as input to the VAE network, enabling implicit occlusion inference.
- Compared with prior approaches, our method demonstrates superior performance in challenging navigation scenarios.

II. Related Work

A. Classical Model-based Method

The problem of robot crowd navigation has been investigated over the past decades. Some reaction-based methods [5][10][11] rely on immediate observations of surrounding humans to generate reactive collision-avoidance behaviors. For example, the Optimal Reciprocal Collision Avoidance (ORCA) [5][12] models other agents as velocity obstacles and computes optimal velocities under the reciprocal assumption. Social Force [10] treats humans and robots as particles subject to attractive and repulsive forces, generating safe trajectories. However, reaction-based methods do not account for the future evolution of neighboring agent, and they are inherently short-sighted.

Another prominent research direction is trajectory-based methods, which predict the future motions of nearby agents to enable foresighted navigation. RR-GP [13] integrates Gaussian process obstacle prediction with RRT-Reach reachability analysis, and employs a chance-constrained RRT [14] for planning. Besides, Dynamic Channel [15] proposes a Delaunay triangulation–based framework that evaluates channel feasibility from humans' positions and velocities, and employs delayed A* search [16] to generate optimal trajectory. Despite their foresight, these methods often produce overly conservative plans that restrict large portions of the space, and may even lead to deadlock in crowds.

B. Reinforcement Learning for Crowd Navigation

In recent years, reinforcement learning (RL) has been widely explored for crowd navigation, with many studies focusing on modeling agent interactions and cooperation. Initially, CADRL [6] introduced a decentralized navigation framework where Deep V-Learning captures pairwise interactions between agents through a value network, allowing them to select collision free actions. But it needs to be pretrained with supervised learning. In LM-SARL [17], an attention mechanism is employed to extract pairwise interactions between the robot and every surrounding human. What's more, [18] and [8] adopt graph convolutional networks to encode higher-order interactions into each agent's state representation. Due to using predefined values to hardcode HH attention scores, they cannot effectively distinguish the difference between HR and HH interactions. DS-RNN [7] utilizes a spatio-temporal graph [19] to reason about spatial and temporal relationships, thereby supporting robot decision-making in crowd navigation. Nevertheless, it neglects to model the interactions among humans. To solve this, [20] proposed a recurrent graph neural network with attention modules to model heterogeneous spatio-temporal interactions among agents. However, these methods assume full observability, which seldom holds in realistic environments. There are often occlusions that prevent the robot from detecting all nearby humans, thereby exposing a substantial gap between simulation studies and realistic deployment.

C. Occlusion Inference

In the field of autonomous driving, researchers have first studied the problem of occlusion reasoning, namely predicting the probabilistic occupancy of regions that are hidden from direct observation. Afolabi et al. [21] pioneer the "People as Sensor" paradigm, leveraging human behaviors as observations to infer occluded environment regions for mapping. Building on this idea, [22] models multiple drivers as distinct sensor sources and integrates their behavior-based inferences using sensor-fusion techniques to improve occlusion-aware mapping. But in practical settings, directly accessing the states of other vehicles remains challenging. Besides, there are several studies that employ variational autoencoders [23] to explicitly predict future environmental states for robot navigation [24][25]. Moreover, Pas-VAE [9] employs a variational autoencoder to leverage the social behaviors of human agents for occlusion inference. However, as the crowd scale becomes larger, this method struggles to effectively capture interactions, leading to degraded obstacle avoidance performance in crowd navigation.

III. Methodology

A. Preliminaries

In this work, we first formulate the robot crowd navigation task as a partially observable Markov decision process (POMDP) represented by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{R}, \gamma, \mathcal{S}_0 \rangle$. At timestep t, let $\mathbf{s}_t \in \mathcal{S}$ represent the state of the environment, which totally includes the robot state and the states of all humans in the environment. However, due to limited sensing and occlusions, the robot cannot directly observe $\mathbf{s}_t$. Instead, it receives a partial observation $o_t \in \mathcal{O}$, consisting of the robot proprioceptive state and the set of detected humans at time t, where $\mathbf{o}_t = [\mathbf{w}^t, \mathbf{h}_1^t, \mathbf{h}_2^t, \mathbf{h}_3^t, \ldots, \mathbf{h}_n^t]$ if a total number of n humans are observed by the robot. Notably, the value of n may vary across different timesteps within a certain range. As the robot's state, $\mathbf{w}^t = [p_x, p_y, v_x, v_y, g_x, g_y, v_{max}, \theta, radius]$ represents position, velocity, goal position, maximum speed, heading angle, and radius of the robot base, respectively.

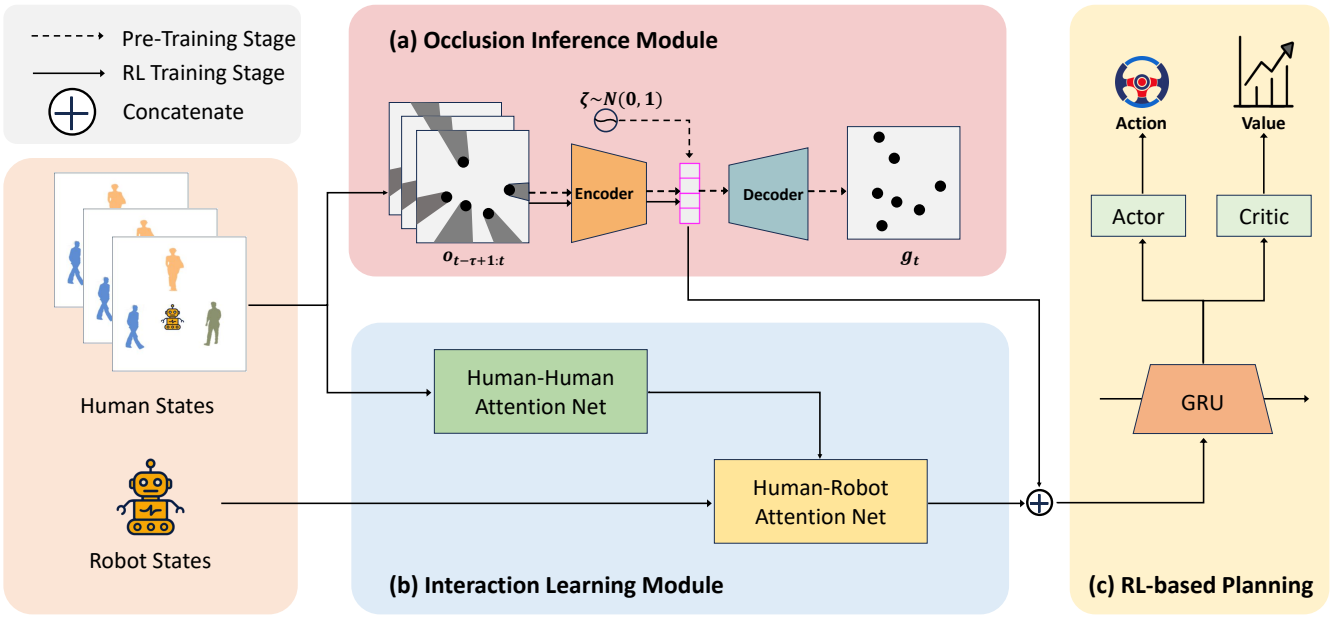


Fig. 2. Framework overview of the proposed method. The framework is composed of three modules: (a) Occlusion Inference Module, (b) Interaction Learning Module, and (c) RL-based Planning Module.

Then, let $x_i^t \in \mathbb{R}^2$ denote the 2D spatial coordinates of the i-th detected human at time step t. Given the K previous trajectories of the i-th detected human $x_i^{t-K:t-1}$, a pre-trained trajectory predictor model [26] is utilized to obtain the M future trajectories $\hat{x}_i^{t+1:t+M}$. We emphasize that the $\hat{x}_i^{t+1:t+M}$ is computed from historical observation and is used only as an auxiliary input, which is not part of the environment state. So, the observable representation of the i-th detected human at time t could be expressed as $\mathbf{h}_i^t = [x_i^{t-K:t-1}, x_i^t, \hat{x}_i^{t+1:t+M}]$.

At the beginning of each episode, the robot starts from an initial state $s_0 \in \mathcal{S}_0$. Guided by its policy $\pi(a_t|o_t)$, the robot takes an action $a_t \in \mathcal{A}$ at each timestep t. This action yields a reward r_t and leads to a transition into the next state s_{t+1} under an unknown state transition distribution $\mathcal{P}(\cdot \mid s_t, a_t)$. Meanwhile, all other humans also take actions according to their own behavior policy model and move to the next states with uncertain state transition probabilities. The process continues until the robot reaches its goal, t exceeds the maximum episode length T, or the robot collides with any humans.

The discount factor γ is in the range $[0, 1]$. $\mathcal{R}$ is the reward function. The objective is to maximize the cumulative discounted reward to determine an optimal policy. Accordingly, the value function $\mathcal{V}$ under policy π is expressed as

$$\mathcal{V}^{\pi}(s_t) = \mathbb{E}_{\pi}\Big[\sum_{k=t}^{\infty} \gamma^{k-t} R_k\Big]. \tag{1}$$

B. Method Overview

The overall pipeline of our navigation framework is illustrated in Fig. 2. The occlusion inference module adopts a VAE structure to reconstruct ground truth OGM from partially observed inputs. In this module, the dashed arrows denote the pre-training stage of the VAE, while the solid arrows represent the reinforcement learning stage, where the inferred occlusion information is integrated into the navigation policy network. Then, we develop an interaction learning module that uses two different attention networks to model human-human interactions (HHI) and human-robot interactions (HRI) separately, enabling the policy to focus on the relevant spatio-temporal interaction representation. Finally, the RL-based planning module takes the concatenated feature representation as input and outputs both the navigation action and the value estimate.

C. Occlusion Inference Module

In many crowd scenarios, there often occurs occlusion phenomenon, where some humans are obstructed by others and thus become undetectable to the robot. The lack of occlusion information hampers the robot's ability to accurately reason dynamic changes in its surrounding environment, which affects its decision-making and planning capabilities. Effectively modeling the uncertainty of occluded regions is crucial for enhancing the robot's obstacle avoidance performance.

Following the prior work [9], we represent the surrounding environment using a 2D occupancy grid map (OGM) in a top-down view. At each timestep t, two types of local maps are generated in the simulation environment, including partially observed OGM $\mathbf{o}_t$ and ground truth OGM $\mathbf{g}_t$. Let H and W denote the height and width of the OGM, respectively. As illustrated in Fig.3 (b), the partially observed local map is designed to emulate the occlusion conditions commonly encountered in practical scenarios, with each grid cell assigned a value of 0 (free), 0.5 (occluded), or

1 (occupied). Meanwhile, the ground-truth map encodes the occupancy information of all humans within the local perception range, including both observed and occluded individuals, as depicted in Fig.3 (c). Each of its grid cells is either occupied or free.

Inspired by SCOPE [25], we employ a variational autoencoder (VAE) to model the stochastic distribution of occupancy grid map (OGM). In our setting, a sequence of τ partially observed OGMs $\{\mathbf{o}_{t-\tau+1}, \ldots, \mathbf{o}_t\}$ is taken as the encoder input, while the ground-truth OGM $\mathbf{g}_t$ serves as the decoder supervision target. This supervised pre-training allows the VAE model to learn the meaningful latent representations $\mathbf{z}$ of occluded region states conditioned on past partial observations, as illustrated in Fig. 2 (a).

Formally, we assume that $\hat{\mathbf{g}}_t$ is generated from a latent variable z sampled from a Gaussian prior $p_\theta(z) = \mathcal{N}(0, I)$. The predictive distribution is:

$$p_\theta(\hat{\mathbf{g}}_t \mid \mathbf{o}_{t-\tau+1:t}) = \int p_\theta(\mathbf{z})\, p_\theta(\hat{\mathbf{g}}_t \mid \mathbf{z}, \mathbf{o}_{t-\tau+1:t})\, dz, \quad (2)$$

where θ are the parameters of the generative model (decoder). Since direct optimization is intractable, we introduce an inference network (encoder) $q_\phi(\mathbf{z} \mid \mathbf{o}_{t-\tau+1:t})$ to approximate the posterior distribution. The training objective is maximize the evidence lower bound (ELBO), where we replace the standard log-likelihood with a mean squared error (MSE) reconstruction loss:

$$\begin{aligned}\mathcal{L}(\theta, \phi; \mathbf{g}_t) = \mathbb{E}_{q_\phi(z|\mathbf{o}_{t-\tau+1:t})}\Big[- \|\mathbf{g}_t - \hat{\mathbf{g}}_t\|^2 \Big] \\ - \mathrm{KL}\big(q_\phi(\mathbf{z} \mid \mathbf{o}_{t-\tau+1:t}) \,\|\, p_\theta(\mathbf{z})\big),\end{aligned} \quad (3)$$

where $\hat{\mathbf{g}}_t$ is the reconstructed ground truth OGM produced by the decoder. The first term corresponds to the MSE reconstruction loss, enforcing accurate prediction of the ground truth OGM, while the second term is the Kullback–Leibler divergence regularizing the approximate posterior towards the prior. In practice, we implement this by minimizing the negative ELBO, $-\mathcal{L}(\theta, \phi; \mathbf{g}_t)$.

Ultimately, we obtain a pre-trained VAE model capable of predicting the occupancy probabilities of occluded regions. However, during the reinforcement learning process of the navigation policy network, we do not directly utilize the reconstructed outputs of this model. Instead, the latent feature distribution $\mathbf{z}$ is employed to implicitly represent occlusion information. This design avoids over-reliance on reconstruction results while effectively incorporating the uncertainty of occluded regions into policy learning. By integrating this implicit representation into the decision-making process, the navigation policy can better adapt to dynamic crowd environments and achieve improved performance in complex scenarios.

D. Attention based Interaction Module

To understand crowd behavior comprehensively, we construct a spatial–temporal graph to model the crowd navigation scenario. Formally, the graph is defined as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ denotes the set of nodes representing agents, including both observed humans and the robot. The edges $\mathcal{E}$

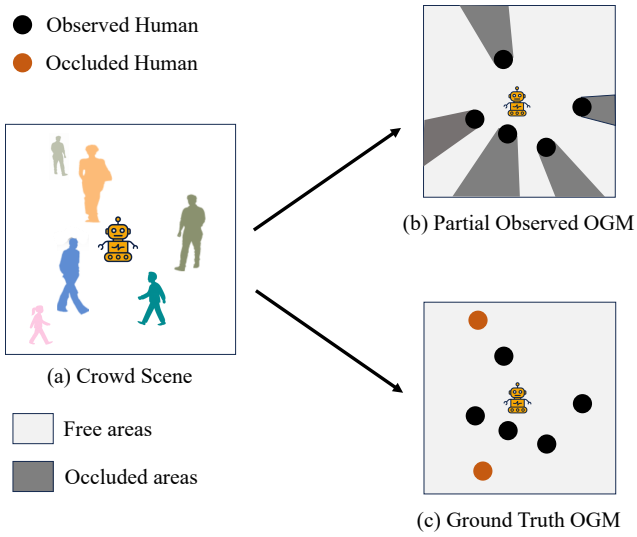


Fig. 3. Illustration of OGM conversion. (a) shows a crowd scene. (b) depicts how this scene is converted into a partially observed OGM, where only detected humans are visible. (c) presents the corresponding ground truth OGM that includes both observed and occluded humans.

express their interactions, capturing both spatial and temporal relationships. To further enhance representation, an attention mechanism is used to evaluate the relative importance of each connection. The influence of each node is determined by the strength of its connections with other nodes where stronger connections lead to higher attention weights. This allows the neural network model to allocate more focus on critical interactions and thereby improving its capacity to model complex crowd dynamics.

Specifically, the spatial graph captures the spatial interactions among agents at moment t. Since human–robot interactions (HRI) differ from human–human interactions (HHI), we employ two distinct attention mechanisms to model them separately. To begin with, the self-attention network is leveraged to capture the interactions among human agents. Each human node aggregates spatial interaction information from its neighboring humans through spatial edges, thereby encoding how the behavior of surrounding individuals influences its behavior. By incorporating human's future trajectories, the robot can exhibit proactive and foresighted behaviors, allowing it to anticipate potential collisions and navigate more effectively. Therefore, the current position of each human and their predicted future trajectory are concatenated and then encoded by a multilayer perceptron (MLP) to obtain the node features $\mathbf{e}^t \in \mathbb{R}^{n \times d_e}$. Subsequently, the current node's feature is transformed using a linear neural network to obtain the embedding

$$\mathbf{Q}_{hh}^t = \Phi_{Q_{hh}}(\mathbf{e}^t), \quad (4)$$

$$\mathbf{K}_{hh}^t = \Phi_{K_{hh}}(\mathbf{e}^t), \quad (5)$$

$$\mathbf{V}_{hh}^t = \Phi_{V_{hh}}(\mathbf{e}^t). \quad (6)$$

where $\Phi_{Q_{hh}}$, $\Phi_{K_{hh}}$ and $\Phi_{V_{hh}}$ represent linear neural network. These embeddings are then used to calculate attention weights α_{hh}^t based on the softmax function over the similarity between the query and key vectors, which could be

formulated as

$$\alpha_{hh}^t = \text{Softmax}\left(\frac{\mathbf{Q}_{hh}^t(\mathbf{K}_{hh}^t)^\top}{\sqrt{d}}\right). \tag{7}$$

These attention weights are then utilized to compute the human embeddings v_{hh}^t through weighted aggregation of the values. This process ensures that the model captures the relevant interactions between each human and its neighbors effectively.

$$v_{hh}^t = \alpha_{hh}^t \mathbf{V}_{hh}^t \tag{8}$$

Following this, we adopt a cross-attention network to model human-robot interactions (HRI), quantifying the significance of each human's impact on the robot. The cross-attention is computed by mapping the robot's state $\mathbf{w}^t$ into the query space and the human embeddings v_{hh}^t into the key and value spaces, as shown in follow:

$$\mathbf{Q}_{hr}^t = \Phi_{Q_{hr}}(\mathbf{w}^t), \tag{9}$$

$$\mathbf{K}_{hr}^t = \Phi_{K_{hr}}(v_{hh}^t), \tag{10}$$

$$\mathbf{V}_{hr}^t = \Phi_{V_{hr}}(v_{hh}^t). \tag{11}$$

The attention weights α_{hr}^t for HRI are calculated in the same manner as for HHI using the scaled dot-product attention mechanism. We then obtain the weighted HRI feature $v_{hr}^t \in \mathbb{R}^{1\times d_{hr}}$ by applying the attention weights α_{hr}^t to the human embeddings $\mathbf{V}_{hr}^t$, effectively capturing the significance of human-robot interactions. This weighted feature encapsulates the critical information from human agents, allowing the robot to better respond to the humans in crowd environment.

The temporal edges connect the graphs $\mathcal{G}$ at adjacent timesteps. With the objective of deriving the final interaction feature representation $\mathbf{f}_{inter}$, we employ a gated recurrent unit (GRU) as the temporal function to process these connections, enabling the robot to integrate temporal dependencies and make proactive decisions.

E. RL based Planning Module

To generate collision-free and foresighted navigation actions, we employ a reinforcement learning (RL) network to train the planning module. At each timestep t, the final state representation h_{final}^t fed to the RL network is represented by

$$h_{final}^t = GRU(h_{final}^{t-1}, [\mathbf{w}^t, \mathbf{f}_{inter}, \mathbf{z}]), \tag{12}$$

where $\mathbf{w}^t$ represents the robot's state, $\mathbf{f}_{inter}$ denotes the interaction features, and $\mathbf{z}$ corresponds to the implicit occlusion features only derived from the VAE's encoder. Besides, h_{final}^{t-1} is the final hidden state from the previous timestep, which captures the accumulated context and memory of past interactions and environmental dynamics, This formulation enables the RL network to effectively capture both the robot's internal state and the surrounding dynamic environment, including occlusion uncertainties and complex interactions, when making navigation decisions. For training the policy network, we use the proximal policy optimization (PPO) algorithm [27], which optimizes both the policy and the value function in tandem. PPO is chosen due to its stable performance in continuous action spaces and its ability to handle large state-action spaces typically encountered in crowd navigation tasks.

The reward function is formulated to guide the reinforcement learning process by encouraging behaviors that bring the robot closer to the goal while avoiding collisions. This reward function incorporates multiple conditions to address different stages of the robot's movement:

$$R(s_t, a_t) = \begin{cases} 10 & \text{if } d_t^{goal} < r_{\text{robot}}, \\ -20 & \text{if } d_t^{\min} < 0, \\ 2.5\left(d_t^{\min} - 0.25\right) & \text{if } 0 \le d_t^{\min} < 0.25, \\ 2\left(-d_t^{goal} + d_{t-1}^{goal}\right) & \text{otherwise.} \end{cases} \tag{13}$$

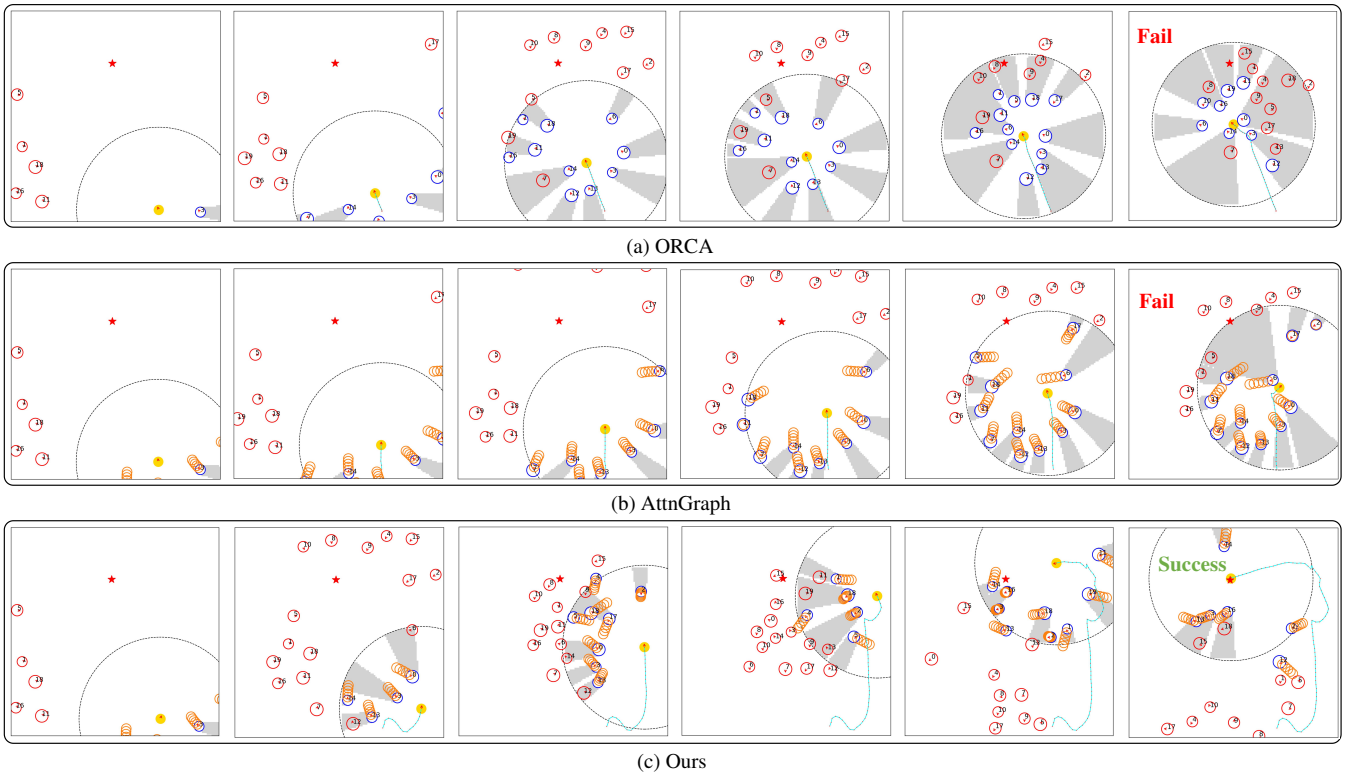
Here, d_t^{goal} denotes the l_2 distance between the robot and the goal at time t, and $d_t^{\min}$ represents the minimum distance between the robot and human agent in the environment at time t, computed from the simulator's full state including both observed and occluded humans. r_{robot} is the robot's radius. When the robot is within a predefined proximity of the goal, it receives a high reward to encourage it to approach the goal position. Conversely, if a collision occurs when $d_t^{\min} < 0$, a severe penalty -20 is imposed to discourage unsafe behaviors. And the term $2\left(-d_t^{goal} + d_{t-1}^{goal}\right)$ is potential-based reward that guides the robot to approach the goal. Notably, it is a standard distance-difference shaping term that measures progress toward the goal across consecutive timesteps. The d_{t-1}^{goal} is stored from the previous step and is used solely to compute this progress signal.

IV. Experimental Verification and Results

A. Experimental Setup

We simulate a 2D environment in a $12m \times 12m$ area, as shown in Fig. 4. The simulated environment adopted for policy network training is more realistic than those in some previous works. First, our robot's sensor has a limited detection range of 5 meters, represented by a dashed circle, whereas some works unrealistically assume the robot has an infinite detection range. Additionally, we also simulate occlusion conditions occurring in the actual environments, where information about humans blocked by others is unavailable to the robot. These occluded areas within the detection range are visually represented in gray.

Furthermore, the simulation environment accommodates up to 20 humans, creating a denser and more interactive crowd. In each round, the robot's starting and target positions, as well as the positions of the humans, are randomly selected. To simulate continuous pedestrian flow, individuals immediately move to a new random target once they reach their current destination. In order to model the diversity of human traits, pedestrians are not uniform but instead parameterized individually. Each human is given a maximum walking speed sampled from $[0.5, 1.5]m/s$ and a body radius sampled from $[0.3, 0.5]m$, which introduces natural



(a) ORCA

(b) AttnGraph

(c) Ours

Fig. 4. Comparison of different methods on the same test episode. The yellow disk represents the robot, the red arrow indicates the direction of each agent, and the red star marks the robot's goal. The robot's sensor range is outlined with a dashed boundary, and occluded regions are shown in gray. Humans are represented as hollow circles. The blue indicates pedestrians within the sensing range and observed by the robot, while the red denotes pedestrians either outside the sensing range or within it but occluded. The predicted future trajectories of pedestrians are illustrated in orange. The green is the historical trajectory of robot.

variability into the simulated crowd. With the purpose of randomizing human behaviors, humans intermittently alter their goals to random positions within an episode. All humans are controlled by ORCA and react only to other humans but not to robot, following a commonly used crowd navigation evaluation protocol. This setup prevents the model from learning an overly aggressive strategy in which the robot forces people to clear the way, thus maximizing rewards.

For motion dynamics, we adopt holonomic kinematics for both the robot and humans. At each time step t, the action $a_t = [v_x, v_y]$ specifies the desired velocity along the x and y axes. The action space is continuous with a maximum speed of $1m/s$. We assume that agents can immediately reach their desired velocities and sustain them for a time interval Δt. Accordingly, the positions are updated as $p_x[t+1] = p_x[t] + v_x[t]\Delta t$ and $p_y[t+1] = p_y[t] + v_y[t]\Delta t$.

To effectively compare our method with others, we conduct experiments on 500 randomly selected and unseen test cases, ensuring that the assessment reflects the model's ability. The performance is measured using a set of navigation-oriented metrics that provide a comprehensive evaluation of navigation quality. Specifically, we report the success rate, defined as the percentage of episodes in which the robot reaches its goal, along with the collision rate and the timeout rate. In successful episodes, the evaluation further includes the average navigation time measured in seconds and the path length measured in meters. These metrics jointly capture navigation safety and efficiency in crowd environments.

Following is some training details. We train the policy using reinforcement learning for 2×10^7 timesteps. We optimize the policy with PPO using Adam optimizer and learning rate 4×10^{-5}. At each policy update, we collect 480 transitions from 16 parallel environments with 30 steps per environment, and perform 5 PPO optimization epochs with 2 mini-batches.

B. Quantitative Evaluation

At the outset, we conduct a comparative experiment to verify the importance of considering occlusion information in crowd navigation. To ensure fairness and objectivity, we evaluate both optimization-based and learning-based approaches under two different settings: with occlusion and without occlusion. The only distinction between these two configurations lies in whether the robot has access to information about humans that are occluded by others, while all remaining experimental conditions are remained the same. This design allows us to isolate the effect of occlusion and directly measure its impact on navigation performance. As shown in Fig. 5, the results consistently demonstrate a decline in success rates when occlusions are present. For example, the success rate of ORCA [5] decreases from 0.69

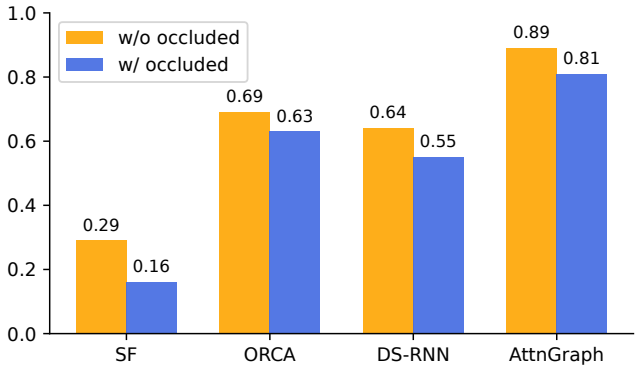


Fig. 5. Comparison of navigation success rates across different methods under two conditions: without occlusion (yellow) and with occlusion (blue). The horizontal axis denotes the evaluated methods, while the vertical axis represents the success rate.

to 0.63, DS-RNN [7] drops from 0.64 to 0.55, and even AttnGraph [20], which achieves the highest overall performance, falls from 0.89 to 0.81. This downward trend across all evaluated methods underscores the significant influence of occlusion on navigation effectiveness. These findings highlight that ignoring occlusion severely undermines the performance of navigation policies, particularly in complex and dynamic crowd environments. Consequently, explicitly modeling occlusions is essential for safe and reliable crowd navigation in real robotic systems.

Table I presents a comprehensive comparison of different navigation methods under occlusion settings. For clarity, the best result in each column is highlighted in bold, while the second-best result is underlined. Traditional optimization-based baselines such as ORCA and SF exhibit limited performance, achieving success rates of 0.63 and 0.16, respectively, while also suffering from high collision or timeout rates. Among the learning-based methods, PaS-VAE and DS-RNN achieve moderate success rate. However, they still show considerable collision rates of 0.29 and 0.41, indicating insufficient performance in dense crowd scenarios. AttnGraph and i-AW-CL achieve stronger overall performance, which demonstrates the benefits of attention-based interaction modeling. Due to neglecting occlusions, they exhibit a non-negligible collision rate, which can degrade obstacle-avoidance performance.

By contrast, our proposed method significantly outperforms all other previous approaches across success rate and collision rate. It achieves the highest success rate of 0.95 while maintaining the lowest collision rate of 0.05. Although our method yields a slightly longer average navigation time than AttnGraph, the safety gains outweigh this marginal cost, and it maintains a moderate path length that balances efficiency and safety. Overall, these results demonstrate that our method not only ensures reliable goal-reaching but also enhances safety by effectively reducing collision risks. This highlights the necessity of particularly modeling occlusion and interactions to achieve safe and efficient navigation performence in crowd scenes.

TABLE I

THE NAVIGATION RESULTS IN TERMS OF SUCCESS RATE, COLLISION RATE, TIMEOUT RATE, NAVIGATION TIME, AND PATH LENGTH.

Method	Success Rate	Collision Rate	Timeout Rate	Nav Time	Path Length
ORCA [5]	0.63	0.34	<u>0.03</u>	22.47	<u>19.74</u>
SF [10]	0.16	0.20	0.64	30.16	34.79
PaS-VAE [9]	0.71	0.29	**0.00**	18.99	**16.92**
DS-RNN [7]	0.55	0.41	0.04	21.43	31.43
AttnGraph [20]	0.81	0.19	**0.00**	**14.32**	19.93
i-AW-CL [28]	<u>0.92</u>	<u>0.08</u>	**0.00**	16.25	21.60
Ours	**0.95**	**0.05**	**0.00**	<u>15.80</u>	22.05

C. Ablation Study

We study the importance of the interaction learning module M_{inter} and the occlusion inference module M_{occ}. As shown in Table II, enabling either module alone leads to limited or unstable improvements, indicating that interaction modeling without occlusion awareness remains constrained under partial observability, while occlusion inference without structured interaction reasoning is insufficient for safe decision-making. In contrast, jointly enabling both modules yields a substantial performance gain, demonstrating a clear complementary effect.

To assess the effectiveness of using the VAE latent representation z for decision-making under occlusion, we conduct an ablation experiment in which the policy is conditioned on features extracted from the VAE reconstruction instead of the latent z. Specifically, we decode z with the VAE decoder to obtain a fully reconstructed image, and then extract features using a convolutional neural network. All other settings are kept same to ensure fair comparison. Under this variant, it attains a success rate of 0.75, a collision rate of 0.09, and a timeout rate of 0.16, suggesting that planning based on reconstructed observations is insufficient to match the performance achieved when directly leveraging the latent representation z. This motivates conditioning the policy on the latent feature z because reconstructing a full observation under severe occlusion is unreliable and introduce artifacts that corrupt the policy input. In contrast, latent variable z provides a compact representation that better reflects occlusion-induced uncertainty, reducing sensitivity to reconstruction noise and improving performance under partial observability.

D. Real-World Experiment

To validate the effectiveness of our method in real-world scenarios, we deploy the model trained in the simulator onto a four-wheel differential drive robot for field testing. The robot is equipped with a ZED depth camera, an laptop equipped with a NVIDIA RTX 2060 GPU, and a Livox 360 LiDAR. We utilize the LV-DOT [29] for pedestrian detection and the FAST-LIO algorithm [30] to estimate the robot's pose. We measure an average runtime of approximately 25.3 Hz during execution on the real robot. For more details, please refer to the supplementary video.

TABLE II
ABLATION RESULTS FOR EACH MODULE.

M_{inter}	M_{occ}	Success Rate	Collision Rate	Timeout Rate
✓	×	0.87	0.12	0.01
×	✓	0.72	0.27	0.01
✓	✓	0.95	0.05	0.00

V. CONCLUSIONS

In this paper, we propose a novel robot crowd navigation framework that simultaneously takes into account both occlusion and interaction effects. Briefly, we leverage the VAE to implicitly infer the states of occluded regions and utilize two different attention network to exploit both HHI and HRI. The occlusion and interaction features are then combined and used as input to a reinforcement learning network to train an excellent navigation policy. Extensive experimental results demonstrated that our approach outperforms previous methods in a variety of scenarios, highlighting its necessity in incorporating occlusion inference and heterogeneous interactions. The limitation of our current approach lies in the precision of the occlusion inference algorithm. Future work will explore more suitable neural networks, such as diffusion model, to better infer the occupancy probabilities of occluded regions, with the purpose of further enhancing navigation performance.

REFERENCES

[1] P. Cai and D. Hsu, "Closing the planning–learning loop with application to autonomous driving," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 998–1011, 2023.

[2] W. Yuan, Z. Li, and C.-Y. Su, "Multisensor-based navigation and control of a mobile service robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 4, pp. 2624–2634, 2021.

[3] W. Dong, D. Huang, J. Liu, C. Tang, and H. Zhang, "Rtagrasp: Learning task-oriented grasping from human videos via retrieval, transfer, and alignment," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 1–7.

[4] A. Santamaria-Navarro, S. Hernández, F. Herrero, A. López, I. d. Pino, N. Rodríguez-Linares, C. Fernández, A. Baldó, C. Lemardelé, A. Garrell, J. Vallvé, H. Taher, A. M. Puig-Pey, L. Pagès, and A. Sanfeliu, "Toward the deployment of an autonomous last-mile delivery robot in urban areas: The ona prototype platform," *IEEE Robotics Automation Magazine*, vol. 32, no. 3, pp. 26–39, 2025.

[5] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 3–19.

[6] 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)*, 2017, pp. 285–292.

[7] 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.

[8] 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 Robotics and Automation Letters*, vol. 5, no. 2, pp. 2754–2761, 2020.

[9] Y.-J. Mun, M. Itkina, S. Liu, and K. Driggs-Campbell, "Occlusion-aware crowd navigation using people as sensors," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 031–12 037.

[10] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, pp. 4282–4286, May 1995.

[11] T. Kretz, J. Lohmiller, and P. Sukennik, "Some indications on how to calibrate the social force model of pedestrian dynamics," *Transportation Research Record*, vol. 2672, no. 20, pp. 228–238, 2018.

[12] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE international conference on robotics and automation*. Ieee, 2008, pp. 1928–1935.

[13] 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," *Autonomous Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[14] B. Luders, M. Kothari, and J. How, "Chance constrained rrt for probabilistic robustness to environmental uncertainty," in *AIAA guidance, navigation, and control conference*, 2010, p. 8160.

[15] C. Cao, P. Trautman, and S. Iba, "Dynamic channel: A planning framework for crowd navigation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5551–5557.

[16] D. Foead, A. Ghifari, M. B. Kusuma, N. Hanafiah, and E. Gunawan, "A systematic literature review of a* pathfinding," *Procedia Computer Science*, vol. 179, pp. 507–514, 2021, 5th International Conference on Computer Science and Computational Intelligence 2020.

[17] 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)*, 2019, pp. 6015–6022.

[18] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, "Relational graph learning for crowd navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 007–10 013.

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

[20] S. Liu, P. Chang, Z. Huang, N. Chakraborty, K. Hong, W. Liang, D. L. McPherson, J. Geng, and K. Driggs-Campbell, "Intention aware robot crowd navigation with attention-based interaction graph," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 015–12 021.

[21] O. Afolabi, K. Driggs–Campbell, R. Dong, M. J. Kochenderfer, and S. S. Sastry, "People as sensors: Imputing maps from human actions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2342–2348.

[22] M. Itkina, Y.-J. Mun, K. Driggs-Campbell, and M. J. Kochenderfer, "Multi-agent variational occlusion inference using people as sensors," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4585–4591.

[23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, vol. abs/1312.6114, 2014.

[24] Z. Xie and P. a. Dames, "Stochastic occupancy grid map prediction in dynamic scenes," in *Conference on Robot Learning*. PMLR, 2023, pp. 1686–1705.

[25] Z. Xie and P. Dames, "Scope: Stochastic cartographic occupancy prediction engine for uncertainty-aware dynamic navigation," *IEEE Transactions on Robotics*, vol. 41, pp. 4139–4158, 2025.

[26] 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.

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

[28] Y.-L. Li and H.-P. Tsai, "Inverse attention-weighted model with heterogeneous spatio-temporal interaction graph for autonomous navigation systems," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 2711–2717.

[29] Z. Xu, H. Shen, X. Han, H. Jin, K. Ye, and K. Shimada, "Lv-dot: Lidar-visual dynamic obstacle detection and tracking for autonomous robot navigation," *arXiv preprint arXiv:2502.20607*, 2025.

[30] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

Manuscript 2076 submitted to 2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 2, 2026.