

# Occlusion-Aware Crowd Navigation Using People as Sensors

Ye-Ji Mun<sup>1</sup>, Masha Itkina<sup>2</sup>, Shuijing Liu<sup>1</sup>, and Katherine Driggs-Campbell<sup>1</sup>

**Abstract**—Autonomous navigation in crowded spaces poses a challenge for mobile robots due to the highly dynamic, partially observable environment. Occlusions are highly prevalent in such settings due to a limited sensor field of view and obstructing human agents. Previous work has shown that observed interactive behaviors of human agents can be used to estimate potential obstacles despite occlusions. We propose integrating such social inference techniques into the planning pipeline. We use a variational autoencoder with a specially designed loss function to learn representations that are meaningful for occlusion inference. This work adopts a deep reinforcement learning approach to incorporate the learned representation into occlusion-aware planning. In simulation, our occlusion-aware policy achieves comparable collision avoidance performance to fully observable navigation by estimating agents in occluded spaces. We demonstrate successful policy transfer from simulation to the real-world Turtlebot 2i. To the best of our knowledge, this work is the first to use social occlusion inference for crowd navigation. Our implementation is available at [https://github.com/yejimun/PaS\\_CrowdNav](https://github.com/yejimun/PaS_CrowdNav).

## I. INTRODUCTION

Navigating in a pedestrian-rich environment is an important yet challenging problem for a mobile robot due to deficiencies in perception. In cluttered settings, spatial occlusions are inevitable due to obstructing human agents and a limited sensor field of view (FOV). Existing crowd navigation methods often neglect occlusions and assume complete knowledge of the environment is provided [1], [2]. When deployed in the real-world, these algorithms only consider the detected or observed human agents for collision avoidance. As a result, collisions may occur when occluded human agents suddenly emerge on the robot’s path. However, under similar limitations, humans can safely navigate as they instinctively reason about potential risks. Humans are able to complement their limited sensing capabilities using insights from their past experiences as well as their understanding of social norms (e.g. keeping an appropriate distance from others) [3]. Similar to humans, planning policies should be able to intelligently make inferences in occluded regions to safely navigate partially observable environments.

Previous literature in autonomous driving has proposed successful occlusion-aware planning algorithms [4], [5], but the setting considered is an inherently structured environment such as an intersection. In crowd navigation, the mobility of human agents is unrestricted resulting in highly diverse

This project was supported in part by the Ford-Stanford Alliance and a gift from Mercedes-Benz Research & Development North America, and in part by the National Science Foundation under Grant No. 2143435.

<sup>1</sup>Ye-Ji Mun, Shuijing Liu, and Katherine Driggs-Campbell are with the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign, USA. Email: {yejimun2, sliu105, krdc}@illinois.edu.

<sup>2</sup>Masha Itkina is with the Aeronautics and Astronautics Department, Stanford University, USA. Email: {mitkina}@stanford.edu.

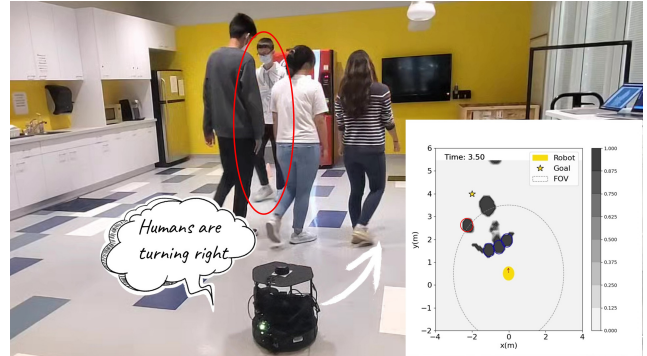


Fig. 1: Turtlebot2i reasoning about a potential occluded human (red circle) based on interactive behaviors of the observed humans. Our PaS inferred OGM in occluded regions is shown on the right.

behaviors [6], and, thus, making occlusion reasoning more challenging. Prior works [7]–[10] demonstrate that missing environmental information can be inferred by observing other people’s interactive behaviors. For example, a human slowing down or stopping abruptly may imply the presence of an obstacle in its path as humans tend to follow the principle of least action and keep their speed constant [3].

Our work is inspired by a growing body of literature on social inference. Afolabi *et al.* [9] first coin the term ‘People as Sensors’ (PaS) and demonstrate how occluded pedestrians can be inferred by observing human drivers’ reactions. This work employs occupancy grid maps (OGMs) [11] for representing agents and the environment as they do not require prior environment knowledge and can handle an arbitrary number of agents in the scene [12]. Itkina *et al.* [8] scale the PaS framework to multi-agent social inference in driving scenes by posing the occlusion inference task as a sensor fusion problem. In this work, we explore the use of PaS in unstructured crowd navigation settings to estimate the location of occluded, freely traversing human agents. We also go beyond inference by integrating the social inference features into planning and analyzing how our enhanced perception pipeline can improve collision avoidance strategies.

We propose incorporating this social inference mechanism into a deep reinforcement learning (RL) algorithm for robust navigation in a partially observable, crowded environment. We train our policy network end-to-end with an occlusion inference module to augment the incomplete perception. For occlusion inference, we employ a variational autoencoder (VAE) [13] architecture to encode interactions between human agents into a low-dimensional latent space using specialized loss terms. The RL policy network takes the latent representation as input, which enables the robot to proactively avoid occluded agents. Simulation results show significant improvement in partially observable navigation

with our occlusion inference technique. We demonstrate successful policy transfer to the real-world Turtlebot2i.

**Contributions:** (1) We propose a deep RL framework for map-based crowd navigation that can make occlusion-aware action plans for a partially observable, cluttered environment. (2) We integrate a VAE into the deep RL algorithm that is trained using specialized loss terms to extract features for occlusion inference. (3) We demonstrate that the joint learning of the occlusion inference and path planning modules results in targeted map estimation that can handle temporary and long-term occlusions enabling proactive collision avoidance.

## II. RELATED WORKS

**Occlusion Inference:** Occlusion inference strategies must be adapted to the occlusion type (i.e. partial vs. full and temporary vs. persistent) and the nature of the environment. Several studies use semantic segmentation to inpaint the unobserved portions of partially occluded objects [14], [15]. During temporary occlusions, previously observed objects can be hallucinated from memory using recurrent neural networks (RNNs) and skip-connections [16], [17]. Wang *et al.* [18] hallucinate static objects using a long short-term memory (LSTM) [19] network and an auxiliary matching loss. Inspired by this approach, we also incorporate a matching loss, but our algorithm performs high-level reasoning for dynamic humans in the presence of long-term occlusions. A new line of work proposed reasoning about persistently fully occluded dynamic agents using the reactive behaviors of observed human agents [7]–[9], [20]. Amirian *et al.* [20] extract statistical patterns from past observations to estimate the probability of human occupancy in occluded regions of crowded scenes. Afolabi *et al.* [9] infer the presence of an occluded pedestrian in a crosswalk from the reactive behaviors of an observed driver. Itkina *et al.* [8] generalize this idea to multiple drivers as ‘sensors’ by employing sensor fusion techniques. We also use the social behaviors of human agents to inform occlusion inference of temporarily and persistently fully occluded agents. We incorporate the interactive features in an RL framework to improve navigation.

**Planning Under Occlusions:** A partially observable Markov decision process (POMDP) [21] is often used to explicitly consider hidden states when planning under occlusions [4], [22]. However, these approaches require the number of occluded agents to be pre-specified, and are intractable with a large number of agents. Deep RL methods have the capacity to capture complex features without requiring prior knowledge of the environment. Liang *et al.* [23] demonstrate sim-to-real steering in densely crowded scenes using deep RL. To handle occlusions, the robot learns to make sharp turns to avoid suddenly emerging pedestrians from occluded regions. We present a means to anticipate such occluded agents using observed social behaviors in crowds, resulting in smoother robot trajectories. Wang *et al.* [24] construct a deep RL algorithm to achieve 3D map-based robot navigation in static, occluded environments. Following this line of work, we propose a map-based deep RL approach that handles occlusions, while navigating highly dynamic environments.

**Crowd Navigation:** Classical crowd navigation techniques like social force models [3] and velocity-based methods [1], [25], [26] follow predefined reaction rules to avoid collisions (e.g. taking the right side of the path to avoid other agents). However, these reaction-based approaches can be short sighted and over-simplify pedestrian strategies for collision avoidance [27], [28]. Other works perform long horizon obstacle avoidance by first predicting human agent trajectories and then finding a feasible path that safely avoids the human agents [29]–[31]. These trajectory-based methods are known to suffer from the robot freezing problem in dense crowds where a feasible path may not be found. Learning-based approaches have been shown to more closely imitate human-like behaviors by learning implicit features that encode social behaviors [27]. Pair-wise interactions between agents are often learned to reason about a dynamic environment and perform collision avoidance [2], [32]. In such methods, the complexity grows with the number of agents in the scene. Additionally, only visible, fully detected agents are typically considered. In our algorithm, we employ OGMs to compactly represent an arbitrary number of agents and learn the mutual influence between agents simultaneously.

## III. PROBLEM STATEMENT

We consider a crowd navigation task where a mobile robot encounters occlusions caused by some agents obstructing other agents from view or by a limited FOV. The robot’s goal is to safely avoid all nearby human agents despite limited visibility and efficiently navigate to its target location.

We formulate the partially observable interactions between agents as a model-free RL problem with continuous state and action spaces,  $S$  and  $A$ . At each time  $t$ , the robot in state  $s_t \in S$  takes an action  $a_t \in A$  given an observation  $o_t \in \mathcal{O}$ . The policy  $\pi : \mathcal{O} \rightarrow A$  directly maps the observed state  $o_t$  to an action  $a_t$  that maximizes the future discounted return:

$$V^\pi(s_t) = \sum_{k=t}^{\infty} \gamma^k R(s_k, a_k, s'_k), \quad (1)$$

where  $R(s, a, s')$  is the reward function and  $\gamma$  is the discount factor. We assume that the human agents’ movements are not influenced by the robot. This assumption is common for crowd navigation as it prevents the robot from achieving collision avoidance effortlessly (i.e. the human agents circumvent the robot while the robot marches straight toward its goal) [32]. Since our aim in this work is to investigate if the robot can employ occlusion inference to prevent collisions in occluded settings, this assumption encourages the robot to actively reason about the presence of occluded agents.

We employ OGMs to represent the environment map surrounding the robot from a bird’s-eye view as shown in Fig. 2. As collisions are unlikely to occur with distant agents, we consider a local OGM around the robot for policy learning. We generate two local OGMs centered around the robot at time  $t$ : a ground-truth OGM  $G_t \in \{0, 1\}^{H \times W}$  and an observation OGM  $O_t \in \{0, 0.5, 1\}^{H \times W}$ , where  $H$  and  $W$  are the OGM height and width, respectively. The ground-truth OGM  $G_t$  captures the true occupancy information for all visible and occluded obstacles, as indicated with free (0)

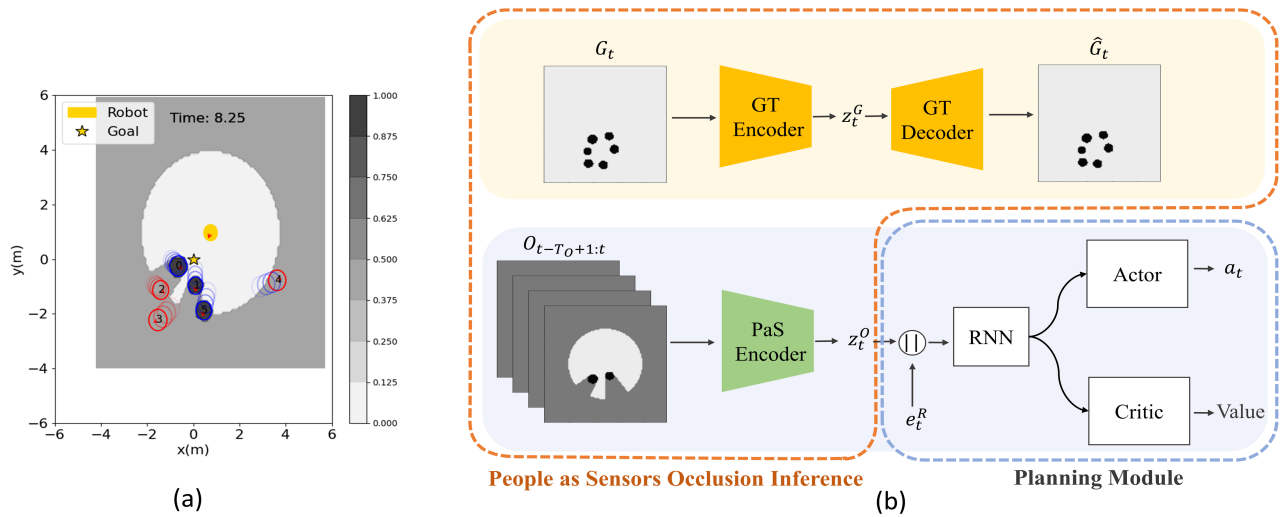


Fig. 2: (a) Illustration of our crowd navigation environment with three observed humans (blue circles) and three occluded humans (red circles). The robot is shown as a yellow circle with its goal as a yellow star. The robot’s observation is represented as the OGM. (b) Our occlusion-aware crowd navigation algorithm consists of a PaS occlusion inference module and a path planning module. The GT-VAE (top) is first pre-trained with ground-truth OGM data. Under the supervision of the GT-VAE, the PaS encoder (bottom) and the actor-critic network in the planning module are jointly optimized for occlusion inference and proactive collision avoidance.

or occupied (1) values in each cell. The observation OGM  $O_t$  includes the region of uncertainty (0.5) referred to as the occluded space, in which some human agents may be hidden.

Within the observation map  $O_t$ , we constrain the robot’s FOV to a smaller circular region around the robot, as shown in Fig. 2(a). This FOV simulates the limited range of a ray-based sensor (e.g. LiDAR). Outside of this circular area is unknown space. We assume that the whole observation map  $O_t$  is the critical area for collision avoidance. Thus, the objective of our algorithm is to infer the unknown occupancy in  $O_t$  caused by dynamic obstructions and a limited FOV. We build the observation map using ray tracing to find detected obstacles. Then, the cells behind the detected obstacles from the robot’s perspective are treated as occluded. We focus on estimating fully occluded obstacles, and, as such, assume partially occluded agents can be detected in the OGM.

#### IV. METHODOLOGY

Our approach comprises an occlusion inference module for map estimation and a path planning module for navigation as shown in Fig. 2 (b). We train the two components in an end-to-end manner so that our perception system and decision-making protocol jointly inform and improve each other.

##### A. Occlusion Inference Module

Our occlusion inference module handles both short-term and long-term occlusions. We define short-term occlusion as the case when a human agent who is not observable at the current time step has been seen previously within a certain observation time window  $T_O$ . An agent who has not been seen throughout this period constitutes a long-term occlusion. We aim for the robot to learn to infer these occluded agents as it learns its navigation policy. The joint optimization between the map estimation and planning modules facilitates a perception branch tuned for navigation and a planner optimized for environmental uncertainties.

Short-term occlusion is handled by using a sequence of past observation OGMs  $O_{t-T_0+1:t}$  as input to our network. Based on these observations, an agent’s trajectory pattern can be extrapolated to its current location in the grid. Estimating the presence of long-term occlusions requires higher-level reasoning. For such unseen agents, we use observed agents’ interactive behaviors to infer the presence of occluded agents, following our past work [8]. We extract these interactive features from  $O_{t-T_0+1:t}$  using two VAEs: a ground-truth VAE (GT-VAE) and a People as Sensors VAE (PaS-VAE). The GT-VAE is pre-trained and provides supervision for the PaS-VAE that is trained with the policy network.

**Ground-Truth VAE:** The GT-VAE acts as a supervisor for the occlusion inference task in our method. It encodes the most recent omniscient OGM  $G_t$  into a compact 1D latent vector  $z_t^G$ . We train GT-VAE using the ELBO loss [13], consisting of an  $\ell_2$ -norm reconstruction loss and a Kullback–Leibler divergence (KL loss). As the omniscient OGM  $G_t$  is not available during test time, the GT-VAE is only used during training. To accelerate policy training, the GT-VAE is pre-trained independently from the RL policy and its parameters are frozen during policy learning.

**People as Sensors VAE:** The PaS-VAE takes in a sequence of observations ( $O_{t-T_0+1:t}$ ) that only has partial information of the environment map as shown in Fig. 2 (b). The sequence of observations allows us to learn from observed social behaviors to inform occlusion inference. For example, a human agent slowing down or taking a detour to avoid another nearby agent can be gleaned from the sequence. To enable the PaS-VAE to learn the missing information from these spatiotemporal interactions, we take advantage of the pre-trained GT-VAE, which contains complete spatial map data. Assuming it is sufficiently well-trained, the GT-VAE can serve as a supervisor model and provide full map information through its latent encoding  $z_t^G$ . Thus, we perform occlusion inference by matching the encodings between the

GT-VAE and the PaS-VAE using our PaS matching loss:

$$L_{\text{PaS}} = \|z_t^G - z_t^O\|_2^2, \quad (2)$$

where  $z_t^O$  in the PaS-VAE encodes the estimated map, including occluded agent information, given the OGM inputs  $O_{t-T_O+1:t}$ . Intuitively, the PaS-VAE can augment its environmental knowledge by associating the social behaviors in its limited observation with the current, full spatial features encoded in the supervisor encoder output  $z_t^G$ . Unlike GT-VAE, the PaS-VAE directly influences policy learning as its latent vector  $z_t^O$  is passed to the policy network, and the two networks are jointly optimized.

We also consider standard VAE loss terms such as the KL loss to regularize the PaS-VAE and an adapted reconstruction loss:  $L_{\text{est}} = \|G_t - \hat{O}_t\|_2^2$  to minimize the  $\ell_2$ -distance between the ground-truth OGM  $G_t$  and our estimated OGM  $\hat{O}_t$ . Instead of optimizing a separate PaS decoder, we use the frozen GT-VAE decoder to output the estimated OGM  $\hat{O}_t$  for the estimation loss  $L_{\text{est}}$  as the encodings between GT-VAE and PaS-VAE should be similar by the PaS loss. However, we found the estimation loss  $L_{\text{est}}$  degrades the navigation performance in favor of more accurate map reconstruction, thus we did not use it for the results in Section VI.

### B. Policy Learning

To enable occlusion-aware path planning, the planning module takes as input the latent encoding  $z_t^O$  provided by the PaS-VAE. Along with the encoded map, the network takes the robot’s state vector as input to help localize itself during navigation:  $s^R = (x_R, y_R, v_x, v_y)$  where  $(x_R, y_R)$  is the robot’s relative position to the goal and  $(v_x, v_y)$  is its velocity. The latent encoding  $z_t^O$  and the robot’s state embedding  $e_t^R$  are concatenated and fed into the planner as shown in Fig. 2 (b).

For the planning module, we adopt the PPO algorithm [33]. The module comprises an RNN and two multi-layer perceptrons (MLPs), corresponding to the actor and critic networks. The critic network approximates the value function, while the actor network updates the policy distributions suggested by the critic. Our PPO implementation is adapted from [32], [34]. However, the planning module can be replaced with any model-free RL algorithm.

## V. EXPERIMENTS

### A. Simulation Setup

Fig. 2 (a) shows our simulation environment. The scene includes six human agents, each modeled as a circle with a radius in  $[0.3, 0.4]$  m. Each agent’s initial position is set along a circle with a 4 m radius and offset with random noise sampled from a uniform distribution over  $[-1, 1]^2$ . The goal positions are symmetric about the origin. All human agents are controlled by Optimal Reciprocal Collision Avoidance (ORCA) [1] with a maximum speed in  $[0.5, 1.5]$  m/s. The human agents reciprocally influence each other’s trajectories but do not react to the robot as discussed in Section III.

The robot is modeled as a circle with a 0.3 m radius. We set the robot’s goal at the origin, where the interactions between agents are the richest and occlusions are likely.

The robot’s starting position is randomly sampled from a  $10\text{ m} \times 10\text{ m}$  square centered at the origin, with a minimum of 6 m from the goal. The robot is assumed to be a holonomic system with a maximum speed and acceleration of 2 m/s and 1 m/s<sup>2</sup>, respectively. Centered at the robot, the observation OGM  $O_t$  is a  $10\text{ m} \times 10\text{ m}$  square, and the robot’s FOV is a circle with a 3 m radius. The OGM resolution is 0.1 m.

### B. Implementation Details

**Reward Function:** Similar to [32], we reward the robot for reaching its goal and penalize (near) collisions with other agents. To encourage the robot to move toward the goal, it receives a positive (negative) reward for getting closer to (further away from) its goal. The reward function is:

$$R(s_t, a_t) = \begin{cases} 10 & \text{if } d_t^g < r_{\text{robot}} \\ -5 & \text{if } d_t^{\text{min}} < 0 \\ 2.5(d_t^{\text{min}} - 0.25) & \text{if } 0 \leq d_t^{\text{min}} < 0.25 \\ 2(-d_t^g + d_{t-1}^g) & \text{otherwise} \end{cases} \quad (3)$$

where  $s_t = (O_t, s_t^R)$  is the observed state,  $d_t^g$  is the robot’s distance from its goal,  $r_{\text{robot}}$  is the robot’s radius, and  $d_t^{\text{min}}$  is the  $\ell_2$  distance of the robot to its closest agent at time  $t$ .

**Network Structure:** The GT-VAE and PaS-VAE consist of 5 convolutional blocks, containing  $4 \times 4$  convolutional layers, Batch Normalization [35], and a rectified linear unit (ReLU), as well as two linear layers to obtain the mean and variance for the latent variable. While the GT-VAE takes in a single OGM  $G_t$  at the current time step, the PaS-VAE takes a sequence of  $T_O = 4$  observation OGMs  $O_{t-T_O+1:t}$  as input, containing 1 s of observation history. Our policy network consists of a single layer for the robot state embedding, an RNN with a hidden state of size 128, and two subsequent 4-layered MLPs for the actor and critic networks.

**Baselines:** We compare the navigation performance of our method to baseline models trained with different perception levels: omniscient ground-truth (GT) and limited sensor observation (OBS) without occlusion estimation. We compare our model-free approach with ORCA [1], a state-of-the-art (SOTA) model-based method. We adapted ORCA to limited FOV for fair comparison. To ablate our results, we evaluate model-free baselines that have a similar network structure as our PaS-VAE in Fig. 2 but exclude some training components. The GT/OBS-FE baselines take an OGM ( $G_t$  or  $O_t$ ) at the current time step as input and are trained without the PaS matching and KL losses. For these baselines, our PaS encoder performs feature extraction (FE) for policy learning without map estimation. The Seq-GT/OBS-FE baselines take sequential OGMs ( $O_{t-T_O+1:t}$  or  $G_{t-T_O+1:t}$ ) as input to explore the effect of incorporating observed behaviors into path planning. The Seq-GT-VAE serves as the oracle model. It is the same as our PaS-VAE but instead takes a sequence of ground-truth OGMs  $G_{t-T_O+1:t}$  as input.

**Training:** We trained all models for  $15 \times 10^6$  time steps. The training details are adapted from [32]–[34]. Each policy gradient update used 180 frames from six episodes. We collected 4 frames per second. The networks are optimized using Adam [36] with a  $10^{-5}$  learning rate. The gradient

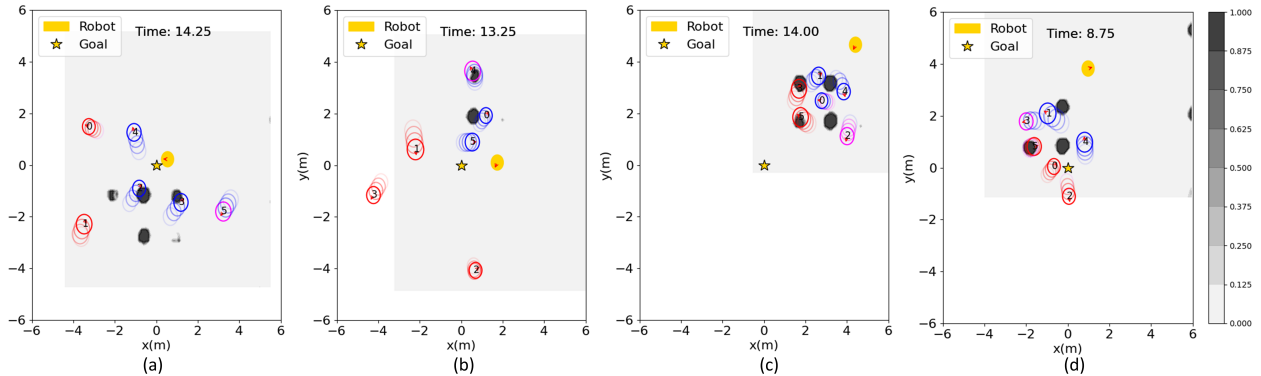


Fig. 3: Illustration of our occlusion inference performance. The figure shows human agent trajectories in our sequential observation input for 1 s and the reconstructed OGMs  $\hat{O}_t$  from our PaS encoding. The observed and occluded humans are denoted as blue and red circles, respectively. If an agent is temporarily occluded but has been seen in the past 1 s, it is denoted in magenta. In the OGM  $\hat{O}_t$ , higher occupied probability cells are darker in shade. The estimated OGMs from our approach show properties that promote better navigation. In Fig. 3(a), the PaS encoding favors estimation of occluded agents that may pose a potential danger to the robot such as an approaching agent (humans 2 and 3) rather than human 4 who is moving away at a distance from the robot. In Fig. 3(b), despite fewer observed interactions at the boundary of the robot’s FOV, temporarily occluded human 4 is successfully inferred by extrapolating from previous observations. In Fig. 3(c), the slow speed of observed human 0 can be used to infer occluded agents ahead like humans 3 and 5. In Fig. 3(d), human 3 making an avoidance maneuver (i.e. a turn) provides insight that occluded obstacles like human 5 may be to its left. Our algorithm successfully provides insight for better crowd navigation in the presence of occlusion based on observed social behaviors.

norm is clipped at 0.5. The KL coefficient was 0.00025. We used 100 episodes for validation and 500 episodes for testing.

## VI. RESULTS

In this section, we evaluate our occlusion inference module and compare our end-to-end occlusion-aware planning policy to the baselines. We demonstrate the benefits of reasoning about occlusions in crowds in terms of safety and efficiency.

**PaS Occlusion Inference:** To evaluate the ability of our algorithm to infer occupancy in occluded regions based on observed agent behaviors, we visualize the PaS latent encodings as estimated OGMs using the GT decoder in Fig. 3. We observe that the estimated obstacles do not perfectly align with the agents due to end-to-end training with the deep RL planner. Instead, the estimated OGM captures important features for navigation, rather than for perfect reconstruction.

Fig. 3(a) shows that the estimated occupancy from the PaS encoding emphasizes inference of occluded agents who may interfere with the robot’s path (humans 2 and 3) rather than agents who are moving away at a distance from the robot (human 4). Since map estimation is trained end-to-end with the policy network, the PaS-VAE learns to draw attention to the risks that directly affect the robot’s policy objectives.

Our method successfully estimates temporarily occluded pedestrians, such as human 4 at the edge of the robot’s FOV in Fig. 3(b). Often agents that are farther away interact less with observable agents, limiting the social cues. However, the sequence of observation inputs to the PaS-VAE and the RNN in the policy module provide sufficient temporal information for the perception pipeline to reasonably estimate human 4.

Lastly, our algorithm infers plausible locations for fully occluded agents who have not previously been detected or seen for a long time. Fig. 3(c) and (d) show examples of observed interactive behaviors, such as slowing down or performing an evasive maneuver, that enable PaS-VAE to

infer the presence of occluded agents in the vicinity. Thus, jointly optimizing the map estimation and planning modules results in a perception branch that is tuned for navigation and able to handle temporary and long-term occlusions.

To quantitatively evaluate map estimation, we use image similarity (IS) [37]. IS assesses the structural similarity of OGMs using Manhattan distance (lower is better). IS has been shown to be more representative of OGM quality than a cell-wise metric (e.g., mean squared error) [38], [39]. We evaluate IS {occupied, free, occluded, total (std. error)} on the estimated OGM  $\hat{O}_t$  relative to the ground-truth OGM  $G_t$ . We obtained IS values of {43.5, 0.0440, 398, 441(2.81)} for our PaS estimated map and of {27.7, 14.8, 400, 443(6.78)} for the observation OGM. While our PaS map can be imprecise in occupancy estimation according to the occupied IS score, the effective free space estimation (low free IS score) facilitates our method’s efficient path planning. Rather than mapping all observable agents precisely, our task-aware approach prioritizes the observable and inferred occluded agents that directly affect the robot’s path to the goal.

**Occlusion-Aware Collision Avoidance:** Table I shows our quantitative navigation results. The SOTA model-based baseline, ORCA, degrades in performance significantly when the robot’s FOV is limited. This performance drop is inevitable for model-based approaches when the model assumptions break (i.e. reciprocal collision avoidance in ORCA assumes full visibility). We employ our model-free, learning-based approach to intelligently reason about a partially observable environment based on observed social context.

We ablate the performance contribution of each component in our algorithm. Our PaS-VAE success rate is comparable to the oracle baseline (Seq-GT-VAE) and performs much better than the limited sensor view baselines (OBS-FE and Seq-OBS-FE) by a maximum of a 12% margin. Comparing the sequential (Seq-GT/OBS-FE) and single (GT/OBS-FE)

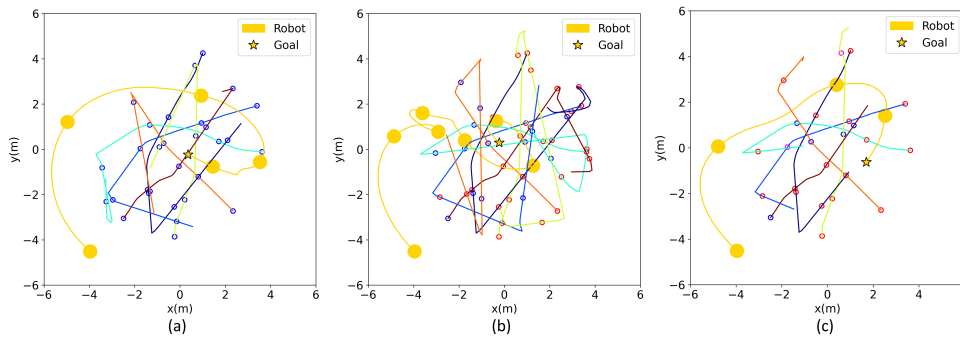


Fig. 4: Trajectories of the robot (yellow) and human agents for (a) Seq-GT-VAE, (b) OBS-FE, and (c) our PaS-VAE. Our approach takes a comparable route to the oracle Seq-GT-VAE. While OBS-FE is highly reactive to unexpected agents resulting in sharp turns, our algorithm reaches the goal using a more efficient and smooth trajectory.

TABLE I: Navigation results in terms of success rate, collision rate, discomfort rate, navigation time, and path length. The success rate is the collision-free goal reaching rate. Timeouts, when neither success nor collision occur, are not included. The best (second best) performance, excluding the oracle Seq-GT-VAE, is denoted in bold (underlined). Our PaS algorithm outperforms the limited view OBS baselines and performs comparably to the omniscient GT baselines.

Method	Success rate	Collision rate	Discomf. rate	Nav. time (s)	Path length (m)
GT-ORCA	0.7	0.3	<u>0.02</u>	17.54	8.06
OBS-ORCA	0.44	0.56	0.04	<b>13.47</b>	<b>6.59</b>
OBS-FE	0.79	0.21	<u>0.02</u>	16.15	13.8
GT-FE	0.83	0.17	<b>0.01</b>	17.31	13.33
Seq-OBS-FE	0.82	0.18	0.02	16.39	13.68
Seq-GT-FE	0.87	0.13	<b>0.01</b>	17.56	16.53
PaS-VAE (Ours)	<b>0.91</b>	<b>0.09</b>	<u>0.02</u>	<u>13.91</u>	12.26
Seq-GT-VAE	0.97	0.03	0.00	14.85	13.39

OGM input baselines, we see that past observations help improve planning. Despite the RNN in all baseline policy networks, the sequential observation input is still helpful to learn the human agent dynamics and improve collision avoidance. Aside from better behavior modeling, PaS occlusion inference components in our approach (i.e. the PaS matching and KL losses) allow the joint system to use past observations coupled with intelligent inference to improve navigation in partially observable settings. Interestingly, despite the Seq-GT-FE and the oracle baseline taking the same omniscient sequential input, the oracle surpasses the other by a 10% success rate with our PaS occlusion inference components.

We observe that excessive map information induces slower navigation times and more conservative robot behavior. The omniscient view baselines (GT-ORCA, GT-FE, and Seq-GT-FE) took longer to reach the goal than the limited view baselines (OBS-ORCA, OBS-FE, and Seq-OBS-FE). Our PaS-VAE has the fastest navigation time and traverses the shortest path to the goal among the learning-based approaches. We attribute the improved efficiency to our map estimation module being trained end-to-end to achieve good performance in crowd navigation. We note that the PaS policy can be more aggressive than the omniscient baselines as indicated by the discomfort rate. In a dense crowd, keeping a large comfort distance with other agents is not always desirable as it often results in the robot freezing problem [40]. In the real world, human agents are generally

cooperative and willing to allow passage to others [40]. Thus, our PaS policy results in more human-like crowd navigation, while remaining highly safe in terms of collisions.

Fig. 4 shows qualitative navigation results. For both Seq-GT-VAE and our method, the robot approaches the goal in a smooth trajectory. In contrast, due to the limited FOV, OBS-FE is more reactive, sharply changing direction to avoid collision with unforeseen agents, and results in a longer navigation path and time. By estimating agents in occluded regions, our method anticipates potential upcoming pedestrians and plans accordingly for smoother and more efficient navigation that is akin to the oracle policy.

## VII. REAL-WORLD EXPERIMENT

We implemented our occlusion-aware policy on a Turtlebot 2i as shown in Fig. 1. We used a 2D RPLIDAR A3 for localization and person detection. We used the Gmapping [41] ROS package for the environment map and the adaptive Monte Carlo localization (AMCL) [42] package to get the robot’s state. To detect pedestrians, we adopted the DR-SPAAM algorithm [43]. We trained our PaS model as described in Section V but with four simulated humans and using unicycle dynamics considering the Turtlebot 2i’s speed limits. In real-world crowd navigation, when the robot observed three humans suddenly turn right, it successfully estimated an occluded human agent on the left and safely avoided collision. Our video demonstration is available here.

## VIII. CONCLUSION

We proposed an occlusion-aware deep RL planner that proactively avoids collision with agents in crowded, partially observable settings. We use social behaviors of human agents as the key insight to make occlusion inferences. We extract these interactive features from a sequence of observations using a VAE and our specialized matching loss. This augmented perception information is trained in tandem with policy learning in an end-to-end manner. Our policy achieves performance similar to a policy with ground-truth map information and significantly outperforms baselines with no occlusion inference. We also demonstrate successful policy transfer to a Turtlebot2i platform in a real-world setting. As future work, we aim to extend our occlusion inference module to occlusion-aware map prediction and test it on a real pedestrian data set with a larger number of humans.

## REFERENCES

- [1] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*, Springer, 2011, pp. 3–19.
- [2] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 285–292.
- [3] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, p. 4282, 1995.
- [4] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Safe reinforcement learning with scene decomposition for navigating complex urban environments," in *Intelligent Vehicles Symposium (IV)*, IEEE, 2019, pp. 1469–1476.
- [5] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 2034–2039.
- [6] E. Rehder, F. Wirth, M. Lauer, and C. Stiller, "Pedestrian prediction by planning using deep neural networks," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 5903–5908.
- [7] K. Hara, H. Kataoka, M. Inaba, K. Narioka, R. Hotta, and Y. Satoh, "Predicting vehicles appearing from blind spots based on pedestrian behaviors," in *International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2020, pp. 1–8.
- [8] M. Itkina, Y.-J. Mun, K. Driggs-Campbell, and M. J. Kochenderfer, "Multi-agent variational occlusion inference using people as sensors," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2022.
- [9] O. Afolabi, K. Driggs-Campbell, R. Dong, M. J. Kochenderfer, and S. S. Sastry, "People as sensors: Imputing maps from human actions," in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 2342–2348.
- [10] L. Sun, W. Zhan, C.-Y. Chan, and M. Tomizuka, "Behavior planning of autonomous cars with social perception," in *Intelligent Vehicles Symposium (IV)*, IEEE, 2019, pp. 207–213.
- [11] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.
- [12] M. Itkina, K. Driggs-Campbell, and M. J. Kochenderfer, "Dynamic environment prediction in urban scenes using recurrent representation learning," in *International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2019, pp. 2052–2059.
- [13] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *International Conference on Learning Representations (ICLR)*, 2014.
- [14] C. Lu and G. Dubbelman, "Semantic foreground inpainting from weak supervision," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1334–1341, 2020.
- [15] P. Purkait, C. Zach, and I. Reid, "Seeing behind things: Extending semantic segmentation to occluded regions," in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 1998–2005.
- [16] F. Ebert, C. Finn, A. X. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections," in *Conference on Robot Learning (CoRL)*, 2017, pp. 344–356.
- [17] J. Dequaire, P. Ondruška, D. Rao, D. Wang, and I. Posner, "Deep tracking in the wild: End-to-end tracking using recurrent neural networks," *The International Journal of Robotics Research*, vol. 37, no. 4–5, pp. 492–512, 2018.
- [18] Y. Wang, G. N. Narasimhan, X. Lin, B. Okorn, and D. Held, "ROLL: Visual self-supervised reinforcement learning with object reasoning," in *Conference on Robot Learning (CoRL)*, 2020.
- [19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [20] J. Amirian, J.-B. Hayet, and J. Pettre, "What we see and what we don't see: Imputing occluded crowd structures from robot sensing," *arXiv*, 2021.
- [21] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *Machine Learning Proceedings*, Elsevier, 1995, pp. 362–370.
- [22] M. Bouton, A. Cosgun, and M. J. Kochenderfer, "Belief state planning for autonomously navigating urban intersections," in *Intelligent Vehicles Symposium (IV)*, IEEE, 2017, pp. 825–830.
- [23] J. Liang, U. Patel, A. J. Sathyamoorthy, and D. Manocha, "Crowdsteer: Realtime smooth and collision-free robot navigation in densely crowded scenarios trained using high-fidelity simulation," in *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2021, pp. 4221–4228.
- [24] L. Wang, H. Ye, Q. Wang, Y. Gao, C. Xu, and F. Gao, "Learning-based 3D occupancy prediction for autonomous navigation in occluded environments," in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 4509–4516.
- [25] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2008, pp. 1928–1935.
- [26] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, 2011.
- [27] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 1343–1350.
- [28] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, "Relational graph learning for crowd navigation," in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 10007–10013.
- [29] 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.
- [30] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 2016.
- [31] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: The case for cooperation," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2013, pp. 2153–2160.
- [32] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, "Decentralized structural-RNN for robot crowd navigation with deep reinforcement learning," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2021.
- [33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv*, 2017.
- [34] I. Kostrikov, *PyTorch implementations of reinforcement learning algorithms*, 2018.
- [35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, PMLR, 2015, pp. 448–456.
- [36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.
- [37] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1384–1397, 2006.
- [38] M. Toyungyernsub, M. Itkina, R. Senanayake, and M. J. Kochenderfer, "Double-prong convlstm for spatiotemporal occupancy prediction in dynamic environments," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 13931–13937.
- [39] B. Lange, M. Itkina, and M. J. Kochenderfer, "Attention augmented convlstm for environment prediction," in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 1346–1353.
- [40] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.
- [41] B. Balasuriya, B. Chathuranga, B. Jayasundara, et al., "Outdoor robot navigation using Gmapping based Slam algorithm," in *Moratuwa Engineering Research Conference (MERCon)*, 2016, pp. 403–408.
- [42] P. Pfaff, W. Burgard, and D. Fox, "Robust Monte-Carlo localization using adaptive likelihood models," in *European Robotics Symposium*, Springer, 2006, pp. 181–194.
- [43] D. Jia, A. Hermans, and B. Leibe, "Domain and modality gaps for lidar-based person detection on mobile robots," *arXiv*, 2021.