CrowdNav-HERO: Pedestrian Trajectory Prediction Based Crowded Navigation with Human-Environment-Robot Ternary Fusion

Siyi Lu[1], Bolei Chen[1], Ping Zhong[1(✉)], Yu Sheng[1], Yongzheng Cui[1], and Run Liu[2]

[1] School of Computer Science and Engineering, Central South University, Changsha 410083, China
{siyilu,boleichen,ping.zhong,shengyu,214712191}@csu.edu.cn
[2] Research Center of Ubiquitous Sensor Networks, University of Chinese Academy of Sciences, Beijing 100049, China
liurun22@mails.ucas.ac.cn

Abstract. Navigating safely and efficiently in complex and crowded scenarios is a challenging problem of practical significance. A realistic and cluttered environmental layout usually significantly impacts crowd distribution and robotic motion decision-making during crowded navigation. However, previous methods almost either learn and evaluate navigation strategies in unrealistic barrier-free settings or assume that expensive features like pedestrian speed are available. Although accurately measuring pedestrian speed in large-scale scenarios is itself a difficult problem. To fully investigate the impact of static environment layouts on crowded navigation and alleviate the reliance of robots on costly features, we propose a novel crowded navigation framework with **H**uman-**E**nvironment-**Ro**bot (HERO) ternary fusion named CrowdNav-HERO. Specifically, **(i)** a simulator that integrates an agent, a variable number of pedestrians, and a series of realistic environments is customized to train and evaluate crowded navigation strategies. **(ii)** Then, a pedestrian trajectory prediction module is introduced to eliminate the dependence of navigation strategies on pedestrian speed features. **(iii)** Finally, a novel crowded navigation strategy is designed by combining the pedestrian trajectory predictor and a layout feature extractor. Convincing comparative analysis and sufficient benchmark tests demonstrate the superiority of our approach in terms of success rate, collision rate, and cumulative rewards. The code is published at https://github.com/SiyiLoo/CrowdNav-HERO.

Keywords: Crowded Navigation · Human-Environment-Robot Ternary Fusion · Pedestrian Trajectory Prediction · Deep Reinforcement Learning

S. Lu and B. Chen—Contribute equally.

B. Luo et al. (Eds.): ICONIP 2023, LNCS 14450, pp. 43–55, 2024.
https://doi.org/10.1007/978-981-99-8070-3_4

Fig. 1. Illustrates the process of our crowded navigation. The red curve indicates the motion trajectory of the agent. The black circle and star indicate the starting position and the target, respectively. The numbers indicate the pedestrian ID. (Color figure online)

1 Introduction

Intelligent agents navigating safely and efficiently in complex pedestrian environments is an important part of the socialization of embodied artificial intelligence. Despite the increasing attention in recent years, significant challenges remain. We consider a more realistic pedestrian navigation environment that is different from previous works, which either only contain fully static environments with static obstacles or purely dynamic environments with pedestrians. Our environment contains both static and dynamic obstruction. In previous work, navigating in crowd and static environments are always treated as two separate problems. Therefore, fixed obstacles are not sufficiently considered in the crowded navigation task. However, the importance of scene layout and semantic information in static environments cannot be ignored for pedestrian navigation. For example, pedestrian motion is constrained by environmental layout, and customers tend to linger and gather under shelves. Although it is possible to treat pedestrians as dynamic obstacles from a robotic perspective, if only obstacle avoidance strategies are used to avoid collisions with pedestrians without considering human-robot interaction, the pedestrian experience will be greatly compromised. With the recent rapid development of interactive learning, reinforcement learning has shown outstanding potential in solving social interaction problems. However, much of the existing work [2,4,5,24] on socially aware crowd navigation is based on stringent assumptions, where it is unrealistic to use pedestrian velocity as the basic feature input for navigation strategies since crowd velocity detection is inherently a thorny problem.

In order to solve the two problems mentioned above, we propose a novel crowded navigation framework, CrowdNav-HERO. At first, a more realistic simulator, HERO-Sim, was built using MP3D datasets and Gibson environment to provide the basis for crowded navigation research in complex and realistic environments. Then, we build a pedestrian state prediction module based on Spatio-Temporal Graph Convolutional Network (STGCN) [15] which uses the set of simple multi-step human historical features instead of expensive single-step human velocity features. Finally, considering Human-Environment-Robot (HERO) ternary fusion in crowded navigation, the pedestrian trajectory predictor and complex obstacle map modeling are integrated into a dual-channel value

estimation network to learn crowded navigation strategies. The superiority and generalization of our method are demonstrated through extensive benchmarks and comparisons with state-of-the-art work in point navigation tasks.

2 Related Work

2.1 Socially Aware Crowded Navigation

In crowded navigation, agents need to interact with humans safely and politely while humans' intentions are unknowable. It is important to learn the relationships among humans because humans usually move in groups [14]. SARL [5] proposes to represent the human state by utilizing a local map, which can't efficiently extract the relationship features among humans. RGL [4] models the agent and the crowd as a relational graph so as to know how much each agent pays attention to the other. SG-DQN [24] introduces a social attention mechanism to reduce the tremendous computational burden of complex interaction representations. Further, Diego et. al [2] design two intrinsic rewards based on SG-DQN that promote strategy learning and accelerate convergence. Recently, method [13] provides a representation of the personal zones of pedestrians based on the prediction of future humans' states. Although the trajectory prediction-based methods are prone to freezing phenomena [22], trajectory prediction is still a promising module in crowd navigation.

2.2 Simulator for Crowded Navigation

Simulation stands out as one of the foremost methodologies employed for assessing social navigation strategies. The linchpin of these simulations resides in the collision avoidance algorithm, which exerts precise control over pedestrian movement. Pioneering the field, the Social Force Model (SFM) [8] was introduced as a means to address the intricate dynamics inherent to crowded scenarios. Within SFM, the behaviors of pedestrians are intricately woven into the fabric of forces that emanate from their interactions. Complementing this approach are velocity-centric methodologies, exemplified by the Velocity Obstacle (VO) paradigm [7]. A more contemporary advancement, the Optimal Reciprocal Collision Avoidance (ORCA) algorithm [19], seeks to empower pedestrians in avoiding collisions within the intricate tapestry of multi-agent environments. These methodological pursuits are underpinned by the presupposition of discernible pedestrian intentions and the inherent homogeneity in pedestrian conduct. Consequently, crowd simulation software fashioned upon these algorithms emerges as a pivotal research platform for the study of navigation in densely populated settings.

Notable examples include CrowdSim[1], an ORCA-centric crowd simulation software, and pedsim_ros[2], a popular crowd simulation system rooted in the SFM framework. Within our research endeavor, we embark upon the creation

[1] https://github.com/vita-epfl/CrowdNav.
[2] https://github.com/srl-freiburg/pedsim_ros.

of a simulator founded on the CrowdNav paradigm, distinguished by intricate spatial layouts, and characterized by the incorporation of HRO ternary fusion. Concurrently, we introduce a non-homogeneous obstacle avoidance algorithm based on ORCA.

3 Problem Formulation

In a crowded domestic scenario with N pedestrians, crowded navigation is modeled as a Partially Observed Markov Decision Process (POMDP) using a tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, P, R, \gamma, L)$. Here, $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space of the agent, $\mathcal{O}$ denotes the observation space, $P : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the transition function. $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ denotes the reward function, γ denotes the discount factor, and L denotes the maximum episode length. At time step t, the agent's state is denoted as $w^t = \{p_x, p_y, v_r, g_x, g_y, \rho, v_p, \theta\}$ and the i-th human's state is denoted as $u_i^t = \{\mathbf{p}^{t+1}, \mathbf{p}^{t+2}..., \mathbf{p}^{t+q}, \rho\}$. The agent state consists of position $\mathbf{p} = (p_x, p_y)$, linear velocity v_r, target position $\mathbf{g} = (g_x, g_y)$, radius ρ, maximum speed v_p, and heading angle θ. The i-th human's state no longer contains the velocity but consists of the predicted future states $\{\mathbf{p}^{t+1}, \mathbf{p}^{t+2}..., \mathbf{p}^{t+q}\} = STGCN(\mathbf{p}^{t-\lambda}, ..., \mathbf{p}^{t-1}, \mathbf{p}^t)$ and radius ρ.

In our settings, a static environment is represented by a global grid map $\mathcal{M}_t$ of size $C \times H \times W$, where each element corresponds to a physical world cell of size $25cm^2$. The map is labeled with a one-hot vector of C channels indicating the semantic category for each pixel. The agent's observation is restricted to a local map m_t of size $C \times M \times M$, and its steering δ is divided into 25 uniformly spaced orientations. The linear velocity v_r is randomly sampled from $[0, v_p]$. The optimal navigation policy, $\pi^* : S^t \mapsto a^t$, maps the current observation state $S^t = \{w^t, u_i^t, m_t\}(i = 1, 2, ..., N)$ to action $a^t = \{v_r, \delta\}$ at time step t, maximizing the expected return, which defined as:

$$\begin{aligned} \pi^*\left(S^t\right) =& \gamma \int_{S^{t+\Delta t}} P\left(S^t, a^t, S^{t+\Delta t}\right) V^*\left(S^{t+\Delta t}\right) dS^{t+\Delta t} \\ &+ \arg\max_{a^t} R\left(S^t, a^t\right) \\ V^*\left(S^t\right) =& \sum_{k=t}^{T} \gamma^k R^k\left(S^k, \pi^*\left(S^k\right)\right). \end{aligned} \tag{1}$$

where $R(S^t, a^t)$ is the reward received at time step t, $\gamma \in (0, 1)$ is the discount factor, V^* is the optimal value function, $P(S^t, a^t, S^{t+\Delta t})$ is the state transition from time t to time $t+\Delta t$. Similar to the formulation in [2], the following reward function is considered to encourage the RL of crowded navigation policy:

$$R = \begin{cases} -0.005\mathcal{T}, \ if\ the\ agent\ is\ alive \\ -0.25, \ if\ collide\ with\ an\ obstacle \\ -0.5, \ if\ collide\ with\ a\ human \\ \alpha(d_{min} - d_{dis})\Delta t, \ if\ too\ close\ to\ humans \\ +5, \ if\ reach\ target. \end{cases} \tag{2}$$

Equation 2 has multiple terms, including penalties and rewards. $\mathcal{T}$ represents cumulative time steps, and α denotes the weight of sub-reward. d_{min} and d_{dis} are the closest human distance and comfortable distance, respectively. Δt is the duration of discomfort distance, set to a single time step. The first term encourages fewer navigation steps and punishes excessive exploration. The second and third terms penalize collisions with obstacles and pedestrians, respectively. The fourth term penalizes impolite navigation that violates human comfort space. The last term rewards successful navigation substantially.

4 HRO Ternary Fusion Simulator

Most existing simulators [2,4,24] ignore the effect of the environment on humans and agents [2,4,24] or assume that the obstacles in the workspace are naive circles or lines [23]. We construct an HRO ternary fusion simulator, HERO-Sim, to deal with it.

4.1 Simulator Setting

HERO-Sim consists of a crowd, an agent, and maps of the domestic environments. In specific navigation tasks, the robot is invisible to humans, thus humans can only interact with humans and obstacles in the simulator. In other words, humans will never notice and avoid the robot, and the robot has to take full responsibility for collision avoidance. In addition, to fully evaluate the impact of pedestrian density on the performance of the crowded navigation strategies, the number of pedestrians is set to be adjustable. Because pedestrians are widely distributed in all corners of the scene, we only consider pedestrians close to the agent while the agent making a decision. Therefore, if not specifically stated, only pedestrians located within the observation field of view of the agent m_t will be considered.

4.2 Static Environment Construction and Collision Avoidance

To make the simulation environments complex and realistic enough, sufficient semantic maps are extracted from realistic domestic scenarios [3,20]. Specifically, we detect the contours of object instances in the map graphically and represent each object as a series of line segments that are connected at the beginning and end. It is worth noting that each object instance is a closed polygon, ensuring that agents do not enter or pass through them. These line segments are considered as constraints for ORCA to solve the linear programming for collision-free motion.

4.3 Crowd Interaction Optimization

Traditional simulators use ORCA [19] to guide pedestrian movement, assuming that pedestrians equally avoid collisions. However, researchers [18] show that human interaction-aware decision-making can be mathematically formulated as

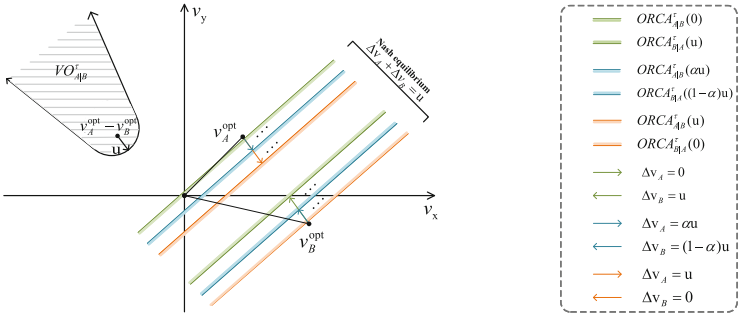

Fig. 2. The set $ORCA^{\tau}_{A|B}(\alpha u)$ of permitted velocity for human A for optimal reciprocal collision avoidance with human B is a half-plane delimited by the line perpendicular to u through the point $v_A^{opt} + \Delta v_A$, $(\Delta v_A = \alpha u)$. Different α can produce different velocity half-plane, Nash equilibrium can be formed as long as $\Delta v_A + \Delta v_B = u$.

searching for Nash equilibrium [9]. So we assume that the human in HERO-Sim acts rationally and tries not to change the original speed while avoiding collisions. As shown in Fig. 2, u is the smallest change required to the relative velocity of human A and human B to avert collision within τ time as defined in ORCA. We give the cost function of humans' decisions:

$$
\begin{aligned}
J_A\left(\Delta v_A, \Delta v_B\right) &= \begin{cases} \Delta v_A & \Delta v_A + \Delta v_B \geqslant u, \\ \infty & \text{else.} \end{cases} \\
J_B\left(\Delta v_B, \Delta v_A\right) &= \begin{cases} \Delta v_B & \Delta v_A + \Delta v_B \geqslant u, \\ \infty & \text{else.} \end{cases}
\end{aligned}
\tag{3}
$$

$J_A\left(\Delta v_A, \Delta v_B\right)$ is the cost function of human A, and Δv_A[3] is the velocity change of human A. In order to minimize the cost function of human A(i.e., solving problem $\min_{\Delta v_A} \ J_A\left(\Delta v_A, \Delta v_B\right)$). Since the cost function only has finite values in a specific interval, this problem can be transformed into a Linear Programming (LP) problem presented as Eq. 4. Clearly, its closed-form solution is given by $\Delta v_A = u - \Delta v_B$, i.e., $\Delta v_A + \Delta v_B = u$.

$$
\min_{\Delta v_A} \Delta v_A \ \left(\text{s.t. } \Delta v_A + \Delta v_B \geqslant u \right) \tag{4}
$$

In order to avoid collisions, velocity pairs $(v_A^{opt} + \Delta v_A, v_B^{opt} - \Delta v_B)$ are randomly selected from the set of Nash equilibrium to give human A and human B. Different Nash equilibria reflect different pedestrian obstacle avoidance strategy preferences. Random sampling pedestrians' obstacle avoidance strategy from the set of Nash equilibrium improves the diversity of pedestrians. Also consider each pedestrian's own personality, which makes them behave consistently among the episodes.

[3] It is worth noting that we don't consider the direction of velocity.

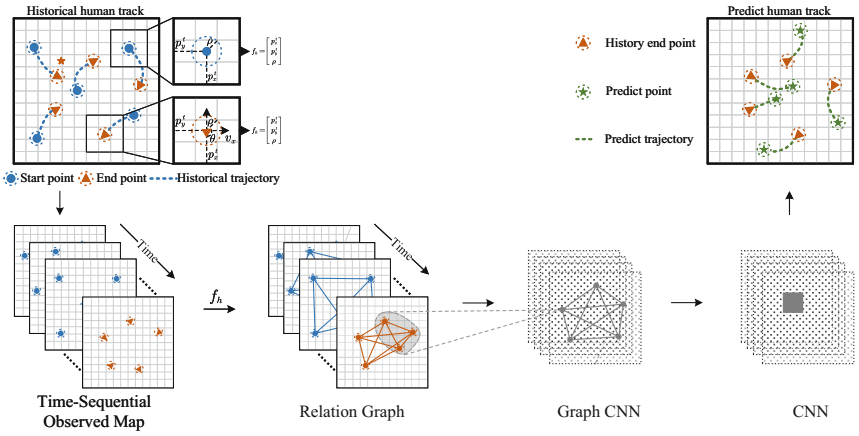

Fig. 3. Spatial-Temporal Pedestrian Trajectory Prediction

5 A Crowded Navigation Framework with HERO Ternary Feature Fusion

In this section, an STGCN-based [15] pedestrian trajectory predictor is first introduced. Then, a two-channel value estimation network is built in order to account for the coupled effects of pedestrians and the environment.

5.1 Spatial-Temporal Pedestrian Trajectory Prediction

Previous work [2,4,5,24] often use the current pedestrian state to predict the next pedestrian state before the agent's decision-making. But many features of the pedestrian state are difficult to obtain precise, such as velocity. So we replaced the use of expensive pedestrian features using simple pedestrian features from multiple moments in history. As shown in Fig. 3, at each time step, the features and relationships of pedestrians are denoted as nodes and edges respectively. We employ a two-layer GCN [11] with residual connections to construct the spatial feature encoder. The l-th layer node features are denoted as $G^{(l)} = [g_1^{(l)}, ..., g_N^{(l)}]^T$, and the node aggregation at layer l is then calculated as:

$$G^{(l+1)} = ReLU(D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}}G^{(l)}W_S^{(l)}). \tag{5}$$

where $\hat{A} = A + I$, I is the unit matrix, A is the adjacency matrix, D is the degree matrix, $W_S^{(l)}$ is the learnable weight matrix of layer l. Then, a multi-layer Temporal Convolutional Network (TCN) [1] is utilized to comprehensively manage time-series features. The input of the TCN is the GCN-encoded past λ-step features of N humans $T_N^\lambda = \{\tau_1^\lambda, ..., \tau_N^\lambda\}$, where $\tau_i^\lambda = \{\tilde{\mathbf{p}}_i^{t-\lambda+1}, ..., \tilde{\mathbf{p}}_i^t\}$, $i = 1, 2, ..., N$. We assume that pedestrian location (p_x, p_y) follows bi-variate Gaussian distribution [15] such that $\mathbf{p}_i^t \sim \mathcal{N}(\mu_i^t, \sigma_i^t, \rho_i^t)$. Denote $\hat{\mathbf{p}}_i^q$ as the predicted future distribution

of i-th human in q steps, $\hat{\zeta}_i^q = \{\{\mu_i^t, \sigma_i^t, \rho_i^t\}, ..., \{\mu_i^{t+q-1}, \sigma_i^{t+q-1}, \rho_i^{t+q-1}\}\}$. Let $\hat{\Lambda}_N^q = \{\hat{\zeta}_1^q, ..., \hat{\zeta}_N^q\}$, the update of $\hat{\mathbf{\Lambda}}_N^q$ at l-th layer in TCN is represented as:

$$\begin{aligned} \hat{\mathbf{\Lambda}}_N^q &= T_N^\lambda && (l=0) \\ \hat{\mathbf{\Lambda}}_N^{q(l+1)} &= \mathcal{F}_{d_l}(\hat{\mathbf{\Lambda}}_N^{(l)})|_{\hat{\mathbf{\Lambda}}_N^{q(l)}} = \sum_{k=1}^{K} \eta_k \hat{\mathbf{\Lambda}}_N^{(q-(K-k)d_l)(l)} && (l>0) \end{aligned} \tag{6}$$

where K is the one-dimensional convolution kernel size, d_l is the expansion factor controlling the jump connection of layer l, $\eta_1 \sim \eta_K$ are the learnable parameters of the convolution kernel. The pedestrian's future q-step trajectory T_N^q can be obtained by sampling from bi-variate Gaussian distributions with parameter $\hat{\mathbf{\Lambda}}_N^q$.

5.2 Dual-Channel Value Estimation Network

The inputs of our dual-channel value estimation network include the observation states of the agent and pedestrians, and an agent-centric local map, $S^t = \{w^t, u_i^t, m_t\}, i = 1, 2, ..., N$. In the socially aware channel, predicted pedestrian states $\hat{f}_h$ and agent's features f_r are input to the GCNs, followed by a ReLU activation σ. A two-layer GCN is employed for human-human and human-agent messaging to model the social awareness, followed by a Multi-Layer Perception (MLP) to summarize the crowded features:

$$F_{sa} = MLP(ReLU^{(l)}(GCN^{(l)}([f_r, \hat{f}_h]))), l = 1, 2 \tag{7}$$

In the map channel, a map predictor of UNet [17] architecture is employed to anticipate the local map m_{t+1} conditional on m_t and the current action a_t. The output of the map predictor is normalized and binarized to obtain the anticipated local map. This process is denoted as:

$$m_{t+1} \leftarrow Bi(Norm(UNet(Trans(m_t, a_t)))). \tag{8}$$

Another UNet network is utilized to extract spatial features in the local maps. The navigation target is embedded as a matrix of dimension $C \times M \times M$, which is concatenated with the local map matrix as the input. Previous network interpretability works [10] show that latent features from different layers of the UNet model contain different types of fine-grained details of objects. Therefore, features from different layers of the decoder are concatenated and fed into an MLP to summarize multi-granularity features:

$$F_{mg} = MLP(cat(UNet([m_t, \phi(O)]))). \tag{9}$$

Where $\phi(\cdot)$ denotes the target embedding. Finally, the outputs of the two channels are weighted by the learnable weights followed by a Gate Recurrent Unit (GRU) [6] to obtain the final value estimation:

$$V^* = GRU(ReLU(W([F_{sa}, F_{mg}]) + b)). \tag{10}$$

6 Experiments

6.1 Experimental Settings

HERO-Sim is used for training and testing, agent-centric local maps (-l) and historical cumulative global maps (-g) [21] are utilized in conjunction with navigation policies to implement multiple crowded Navigation policy variants. Navigating to the end without collision means success, any collision with pedestrians or obstacles and timeout greater than 50-time steps implies failure. The maximum velocity of both the humans and the agent is 1 m/s. The radius of the humans and the agent is set to 0.3 m. γ in formula (1) is set to 0.9. d_{dis} and α in formula (2) are set to 0.5 m and 1.0, respectively. The size of the agent-centric local map is set to 6.4 m $\times$ 6.4 m.

Classical method ORCA and three existing state-of-the-art methods, RGL [4], SG-D3QN [24] and IR-RE3 [2], are implemented as baseline methods to be compared with our method in terms of Success Rate (SR), Collision Rate with Obstacles (CO), Collision Rate with Pedestrians (CP), Average Navigation Step (AS), and Average Reward (AR). The start and end positions of the point navigation task are randomly sampled in the maps. By sampling evenly in the testing environments, 100 navigation episodes are employed for navigation evaluation. An example of point navigation is shown at the top of Fig. 1.

Table 1. Statistical experimental results of point navigation. * indicates the improved ORCA, which is able to avoid obstacles to a certain extent.

Method	**SR** (%) ↑	**CO** (%) ↓	**CP** (%) ↓	**AS** ↓	**AR** ↑
ORCA* [19]	0.18	0.77	0.04	9.16	0.083
RGL-l [4]	0.51	0.38	0.09	14.8	0.325
SG-D3QN-l [24]	0.58	0.34	0.06	**11.4**	**0.470**
IR-RE3-l [2]	0.58	0.37	0.05	11.8	0.426
Ours-l	**0.62**	**0.33**	**0.03**	12.1	0.467
RGL-g [4]	0.52	0.36	0.06	13.4	0.386
SG-D3QN-g [24]	0.61	0.34	0.05	12.1	0.477
IR-RE3-g [2]	0.59	0.38	0.03	**11.4**	0.476
Ours-g	**0.64**	**0.30**	**0.02**	12.8	**0.481**

6.2 Quantitative Evaluations for Crowded Navigation

To evaluate the performance of crowded navigation quantitatively, the various metrics of the 5 methods in the MP3D setting are presented in Table 1. As expected, the SR of ORCA* is low due to the violation of the reciprocal assumption. Since ORCA* has a short navigation distance, there are few opportunities

to interact with pedestrians, so the CP metric is relatively low.By combining RGL with agent-centric local maps, RGL-l achieves an absolute 33% improvement in terms of SR relative to ORCA*, which is attributed to the learning of interaction mechanisms and multi-granularity maps. Benefiting from the usage of pedestrian trajectory prediction to remedy the lack of pedestrian speed features, our approach achieves the lowest CP metric. Although a longer AS means more robust and longer-lasting navigation, the AS decreases somewhat with increasing collision rates (CO and CP). In the point navigation task, our approach maintains a moderate level of AS a competitive AR. When replacing local maps with cumulative global maps, there is a small performance improvement for each method. The reason may be that the navigation goals provided by the point navigation task already provide global guidance and the small-scale maps are sufficient to support the local motion decision-making of the agent.

6.3 Quantitative Evaluation of Impact of Environment on Navigation

We quantitatively evaluate the impact of the static environment and the crowd on navigation through experimental performance in different environments, including a crowd-only simulator(-crowd), crowd-only dataset(-dataset), and simulator with the crowd and static environment(-crowd+env). It's difficult to unify experimental parameters in a different environment, so we only select the success rate and collision rate as evaluation indicators. Experience results in Table 2 show that the success rate is the highest in the simulation environment with only pedestrians. When the pedestrian trajectory from the real trajectory or add static environment, the success rate decreases, and the latter decreases greatly. The experimental results indirectly show the influence of the pedestrian path generation model and static obstacles on navigation.

Table 2. Statistical experimental results of point navigation on simulator only include crowd, real crowd trajectory dataset, and HERO-Sim.

Method	**SR** (%) ↑	**CO** (%) ↓	**CP** (%) ↓
ORCA-crowd [19]	0.43	-	0.57
ORCA-crowd+env [19]	0.18	0.77	0.04
RGL-crowd [4]	0.96	-	0.02
RGL-dataset	0.82	-	0.16
RGL-crowd+env	0.52	0.36	0.06
Ours-crowd	0.98	-	0.02
Ours-dataset	0.98	-	0.00
Ours-crowd+env	0.64	0.30	0.02

6.4 Quantitative Evaluations on Real Pedestrian Dataset

A dataset containing real crowd trajectory can help us verify the performance of our approach in the real world. There are two datasets widely used in both social robot navigation and human motion prediction literature: ETH [16] and UCY [12]. Real crowd trajectory datasets do not include static environments. It is mainly used to test the navigation performance after replacing the velocity feature with the trajectory prediction module. In order for the pedestrian and the robot to fully interact, we randomly generate the starting point and target point of the robot according to the crowd movement range and mainly compared it with RGL. The experience result shows in Table 3, our method significantly outperforms RGL in terms of SR and CP But navigation time is slightly lower than RGL. It is proved that the trajectory prediction modules can assist navigation methods to improve navigation safety indication.

Table 3. Statistical experimental results of crowded navigation on the real pedestrian dataset.

Method	**SR** (%) ↑	**CP** (%) ↓	**AS** ↓	**AR** ↑
RGL [4]	0.82	0.16	12.75	0.6050
Ours	1.00	0.00	15.20	0.8236

7 Conclusion

This paper explores the interplay between pedestrians and static environments in crowded navigation and aim to simplify the process of obtaining environmental features for robots by eliminating the reliance on costly features. To this end, we introduce a new simulator, HERO-Sim, which utilizes a Human-Environment-Robot ternary fusion approach for training and evaluation. Our simulator incorporates realistic static environments constructed using real environment datasets. Additionally, we improve the group interaction model by employing Nash equilibrium, which enhances the group's obstacle avoidance preferences and reduces the homogenization of group behavior. To account for the combined effects of pedestrians and complex static environments, we propose a novel crowded navigation framework, CrowdNav-HERO. This framework incorporates a STGCN-based trajectory prediction module, which replaces the need for velocity with a set of historical trajectory.

In the current research on social navigation, pedestrian behavior is assumed to be homogeneous. In future work, we will study the navigation problem in more complex real pedestrian scenarios.

Acknowledgment. This work received partial support from the National Natural Science Foundation of China (62172443), the Natural Science Foundation of Hunan

Province (2022JJ30760), and the Natural Science Foundation of Changsha (kq2202107, kq2202108). We are grateful for resources from the High Performance Computing Center of Central South University.

References

1. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
2. Baselga, D.M., Riazuelo, L., Montano, L.: Improving robot navigation in crowded environments using intrinsic rewards. In: IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29–June 2, 2023. pp. 9428–9434. IEEE (2023). https://doi.org/10.1109/ICRA48891.2023.10160876
3. Chang, A.X., et al.: Matterport3D: learning from RGB-D data in indoor environments. In: 2017 International Conference on 3D Vision, 3DV 2017, Qingdao, China, October 10–12, 2017, pp. 667–676 (2017)
4. Chen, C., Hu, S., Nikdel, P., Mori, G., Savva, M.: Relational graph learning for crowd navigation. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 10007–10013 (2020)
5. Chen, C., Liu, Y., Kreiss, S., Alahi, A.: Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In: 2019 international conference on robotics and automation (ICRA), pp. 6015–6022 (2019)
6. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1724–1734. ACL (2014)
7. Fiorini, P., Shiller, Z.: Motion planning in dynamic environments using velocity obstacles. Int. J. Robot. Res. **17**(7), 760–772 (1998)
8. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Phys. Rev. E **51**(5), 4282 (1995)
9. Holt, C.A., Roth, A.E.: The nash equilibrium: a perspective. Proc. Natl. Acad. Sci. **101**(12), 3999–4002 (2004)
10. Islam, M.A., et al.: Shape or texture: Understanding discriminative features in CNNs. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021 (2021)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings. OpenReview.net (2017), https://openreview.net/forum?id=SJU4ayYgl
12. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: Computer Graphics Forum, vol. 26, pp. 655–664 (2007)
13. Liu, S., et al.: Socially aware robot crowd navigation with interaction graphs and human trajectory prediction. arXiv preprint arXiv:2203.01821 (2022)
14. McPhail, C., Wohlstein, R.T.: Using film to analyze pedestrian behavior. Sociol. Methods Res. **10**(3), 347–375 (1982)
15. Mohamed, A., Qian, K., Elhoseiny, M., Claudel, C.: Social-STGCNN: a social spatio-temporal graph convolutional neural network for human trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14424–14432 (2020)

16. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: modeling social behavior for multi-target tracking. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 261–268 (2009)
17. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18, pp. 234–241 (2015)
18. Turnwald, A., Olszowy, W., Wollherr, D., Buss, M.: Interactive navigation of humans from a game theoretic perspective. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 703–708 (2014)
19. Van Den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: Robotics Research: The 14th International Symposium ISRR, pp. 3–19 (2011)
20. Xia, F., Zamir, A.R., He, Z., Sax, A., Malik, J., Savarese, S.: Gibson Env: real-world perception for embodied agents. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9068–9079 (2018)
21. Zhai, A.J., Wang, S.: PEANUT: predicting and navigating to unseen targets. arXiv preprint arXiv:2212.02497 (2022)
22. Zhang, X., et al.: Relational navigation learning in continuous action space among crowds. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 3175–3181 (2021)
23. Zhou, Z., et al.: Navigating robots in dynamic environment with deep reinforcement learning. IEEE Trans. Intell. Transp. Syst. **23**(12), 25201–25211 (2022)
24. Zhou, Z., Zhu, P., Zeng, Z., Xiao, J., Lu, H., Zhou, Z.: Robot navigation in a crowd by integrating deep reinforcement learning and online planning. Appl. Intell. **52**(13), 15600–15616 (2022)