Artificial Intelligence Review (2024) 57:108
https://doi.org/10.1007/s10462-024-10739-w

Multi-robot social-aware cooperative planning in pedestrian environments using attention-based actor-critic

Lu Dong[1,5] · Zichen He[2,6] · Chunwei Song[3] · Xin Yuan[4] · Haichao Zhang[2]

Accepted: 24 February 2024 / Published online: 2 April 2024

Abstract
Safe and efficient cooperative planning of multiple robots in pedestrian participation environments is promising for applications. In this paper, a novel multi-robot social-aware efficient cooperative planner on the basis of off-policy multi-agent reinforcement learning (MARL) under partial dimension-varying observation and imperfect perception conditions is proposed. We adopt a temporal-spatial graph (TSG)-based social encoder to better extract the importance of social relations between each robot and the pedestrians in its field of view (FOV). Also, we introduce a K-step lookahead reward setting in the multi-robot RL framework to avoid aggressive, intrusive, short-sighted, and unnatural motion decisions generated by robots. Moreover, we improve the traditional centralized critic network with a multi-head global attention module to better aggregate local observation information among different robots to guide the process of the individual policy update. Finally, multi-group experimental results verify the effectiveness of the proposed cooperative motion planner.

Keywords Multi-agent reinforcement learning · Cooperative navigation · Social aware · Comfort aware · Multi-robot systems

1 Introduction

With the development of robotics and artificial intelligence, autonomous mobile robots are gradually deployed in our daily lives. For example, mobile service robots cannot avoid interacting with multiple pedestrians in scenarios such as airports, campuses, unmanned supermarkets, and intelligent warehouses (Dong et al. 2023). Therefore, it is a significant research hotspot to teach robots social safety awareness. Moreover, in the application scenarios described above, the single robot often faces problems such as limited sensing range, low planning efficiency, and weak stability during the operation. In contrast, multi-robot cooperation can better share the local observations of the environment from multiple perspectives between individuals. This mechanism facilitates extending the perception ability of each robot and ultimately improves planning efficiency.

Extended author information available on the last page of the article

Unlike pure robot co-planning scenarios in He et al. (2022b) and Song et al. (2022), communication between humans and robots is impossible in social interaction. Thus, it is challenging for each robot to perform autonomous collaborative navigation in pedestrian-rich environments. Early RL-based works (Phillips and Likhachev 2011; Wang et al. 2020b; He et al. 2022a) treat pedestrians as dynamic obstacles with simple state-update kinematics. This setting is convenient for robots to handle, but the behavior of pedestrians in reality has specific social properties and uncertainties. For example, pedestrians may suddenly change their movement speed or change their target points. Robots unable to generate proper policies to cope with such situations might cause unsafe issues. In Everett et al. (2018), Everett et al. (2021), Semnani et al. (2020) and Matsuzaki and Hasegawa (2022), researchers design a specialized one-step comfort zone intrusion penalty function to improve the social safety of human–robot interaction. Although this setting is more practical, they choose to deal with the relative social relationship between robots and humans on the basis of a simple proximity function. In real-world scenarios, some pedestrians near the robot may be moving in the same direction or away from it. The importance of these pedestrians may not be as critical as others further away but moving towards this robot since these humans are more likely to collide with the current robot. Meanwhile, one-step reward consideration only would make the robot short-sighted (Chen et al. 2019). In Zhou et al. (2021) and Nishimura and Yonetani (2020), researchers assume that each robot is fully perceptive of environments. This setting is suitable for handling small scene tasks. In Liu et al. (2021), researchers propose an effective social-aware planning method for the single robot. However, in real life, there are usually multiple robots serving simultaneously in public places, such as airports and hospitals. The field of view (FOV) of each robot is limited by the sensing range of the dominant sensor. Robot lacks a way to utilize multi-view global information from others and has to tackle the time-varying issue of the observation dimensions of the human flow.

At the level of multi-robot cooperative operations, centralized works like (Tang et al. 2018; Wang et al. 2021; Yu et al. 2021) suffer from scalability and the lack of effective global information aggregation patterns. Decentralized approaches like (Desaraju and How 2011; Sartoretti et al. 2019; Liu et al. 2022) are efficient but prone to local optimality and self-interested issues. Our previous work (Song et al. 2022) is based on the centralized training and decentralized execution (CTDE) MARL paradigm and has been demonstrated to achieve state-of-the-art results in dense and pure-robot co-planning tasks. We hope to extend this architecture to further address pedestrian participation multi-robot co-planning tasks.

To address the problems mentioned above, we propose a novel social-aware multi-robot co-planning method called Multi-agent Social-Aware Attention-based Actor-Critic (MSA3C) for the task scenario described in Fig. 1. In this task scenario, multiple robots with the limited FOV need an efficient approach to navigate collaboratively in a common environment full of uncertain pedestrians. First, we introduce the attention-based centralized critic to better aggregate multiple observations from different perspectives of multiple robots, and we utilize its output value to provide better guidance to each robot for local policy updates. Also, we design a rollout data processing pipeline to adapt to the off-policy MARL setting and handle the variable dimension issue of human flow observations within the FOV of each robot. This trick is much more practical and can effectively improve the scalability of our method. Then, we design a temporal-spatial graph (TSG)-based social encoder to extract the relative importance of the surrounding pedestrians to assist each robot in making better social decisions. In addition, to enhance the social comfort awareness of each robot and avoid unnatural, aggressive, and short-sighted decisions,

we introduce the K-step lookahead reward function during the training phase to better evaluate the impact of current action commands of robots on future human–robot interaction. To sum up, our main contributions are as follows:

- We propose a CTDE-based MSA3C framework for handling multi-robot cooperative planning tasks with social safety and comfort awareness under the limited FOV condition. Our method achieves great performance in multiple experiments compared to various baselines.
- We design a multi-agent rollout replaybuffer to align the time-varying dimension of historical transitions and introduce a parameter-sharing social encoder for each robot based on TSG network to help robots better understand the relative social relationship of surrounding pedestrians.
- We incorporate a predictive K-step lookahead reward function into the MARL paradigm during the training phase to enhance the social comfort awareness of each robot and prevent the adoption of unnatural and shortsighted policies.

The rest of this paper is organized as follows. The related work is introduced in Sect. 2. A detailed discussion of our method is presented in Sect. 3. The experiment procedure and results of our method are shown in Sect. 4. Finally, we conclude this paper in Sect. 5.

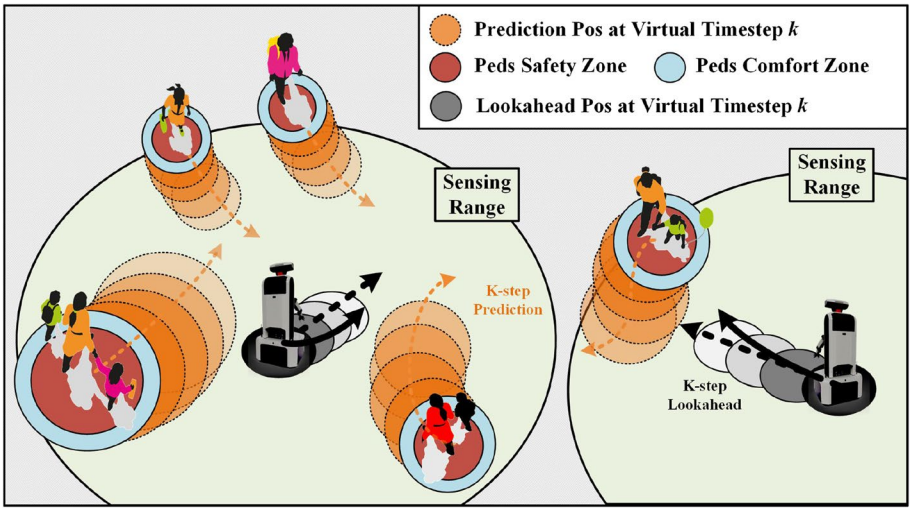

Fig. 1 Multiple robots with limited sensing range perform decentralized cooperative inference planning in the pedestrian participation environment. Social-aware and CTDE-based multi-agent reinforcement learning architecture helps robots interact better with pedestrians while planning efficiently and collaboratively. K-step lookahead interaction reward item motivates robots to generate stronger awareness of crowd comfort and safety

2 Related work

2.1 Multi-robot cooperative planning

Classical multi-robot co-planning methods can be divided into centralized methods and decentralized methods. Among centralized methods, (Mellinger et al. 2012) is a typical optimization-based collaborative trajectory generation method. Yu and LaValle (2016) is a typical co-planning work on the basis of the heuristic search. Centralized methods have completeness or probability completeness. However, these methods rely on accurate global information acquisition or aggregation approaches. Also, they suffer from several issues, such as low scalability and discrete dimensional explosion. As for the decentralized pattern, Desaraju and How (2011) present a decentralized multi-agent rapidly exploring random tree and can sample multiple feasible paths for multiple robots simultaneously. The application of this approach requires the construction of explicit communication channels between robots. Reaction-based velocity obstacle (VO) methods (Douthwaite et al. 2018) have been widely researched for their high efficiency and real-time properties. Later, Reciprocal VO (RVO) (Snape et al. 2011) improves the oscillation issue of VO. In Berg et al. (2011), optimal reciprocal collision avoidance (ORCA) and its variants are presented to improve the optimality and efficiency of RVO further. In Wang et al. (2018); Huang et al. (2019b), researchers extend the application range of ORCA to handle heterogeneous and non-holonomic constraints existing in co-planning tasks. VO-based approaches rely on perfect perception conditions and suffer from freezing robot issues. Moreover, such methods cannot be generalized well across different scenarios without the cumbersome handcrafted process.

2.2 Learning-based social aware robot motion planning

Currently, mobile robots have been widely deployed in real-life scenarios where humans are involved (Dong et al. 2023). Therefore, relevant studies related to teaching robots to learn to interact reasonably with pedestrians and decrease the intrusion into their motion comfort zones have attracted the attention of many researchers (Chen et al. 2019). Meanwhile, the development of deep learning (DL) and RL has provided novel avenues for data-driven social aware planning technologies. Michael Everett et al. have been working on promoting the research on decentralized multi-robot motion planning deployed in pedestrian environments (Everett et al. 2018, 2021; Semnani et al. 2020). Their research works have also inspired later researchers (Fan et al. 2020; Chen et al. 2019; Liu et al. 2021; Matsuzaki and Hasegawa 2022). However, these works only integrate one-step human–robot interaction reward and handle social relations on the basis of proximity function. The robot cannot fully understand the importance of each pedestrian in its FOV and is prone to be short-sighted. Some researchers select raw sensor data (e.g., 2D Lidar (Qiu et al. 2022)) as input to design end-to-end multi-agent RL patterns to handle multi-robot co-planning in real scenes. The sensor-level approaches overcome the problem of variable observation dimension but introduce large-size input and interpretability issues. Moreover, the MARL framework used by these methods adopts a simple concatenation trick (Yu et al. 2021) to aggregate joint observations from different robots. This unfocused way brings unnecessary redundant information to individuals. In Qureshi et al. (2021) and Rivière et al. (2020), scholars introduce supervised learning to teach the robot planning policy. The actual performance of these methods relies on the quality of labeled data or demonstrations.

3 Methodology

3.1 Dec-POMDP configuration

First, we model the problem of multi-robot cooperative planning in human crows as a decentralized partially observable Markov decision problem (Dec-POMDP) consisting of the tuple $G = (\mathcal{N}, S, O, A, P, r, \Omega)$. Here, $\mathcal{N} = \{1, \ldots, n\}$ is a finite set describing the number of robots in the environment. $s \in S$ represents the full state space of all agents (including robots and pedestrians). $o_i \in \Omega \sim O(s, i)$ denotes the partial observation of each robot i for the external world at each timestep. O is the observation kernel. $P\left(s' \mid s, a\right)$ is the state transition function of the environment. A represents the joint action space of all robots. r is the team reward for all robots.

3.1.1 Observation

In this paper, the observation setting of the robot i at timestep t is as follows:

$$\mathbf{o}_t = [\mathbf{s}_{\text{ego}}, \mathbf{o}_{\text{other}}] \tag{1}$$

where $\mathbf{s}_{\text{ego}} = [p_x, p_y, r, g_x, g_y, v_{\text{pref}}, v_x, v_y, \theta]$ is the entire state vector of the ego-robot, including the position, the radius of the safety domain, the relative position of the target point, the preferred velocity, and the orientation. $\mathbf{o}_{\text{other}}$ is an imperfect perception vector with variable dimensionality. It only integrates the relative position of all agents (including pedestrians and other robots) in the FOV of ego-robot at timestep t.

3.1.2 Action space

All robots in this paper are holonomic. This setting improves the training efficiency. We can utilize the hybrid planning framework in our previous work (He et al. 2022b) to handle extra optimization objectives and constraints, and improve the quality of the final motion trajectory at the back-end. In this paper, each robot have continuous action space $a_t = [v_x, v_y]$ in the range of $[v_{\min}, v_{\max}]$. These two parameters depend on the performance of motors. In fact, we scale the speed range to $[-1, 1]$ during the training phase and rescale it during the environment update phase.

3.1.3 Reward setting

In this paper, we split the reward setting of the robot i into a basic configuration and a K-step lookahead reward setting. Note that we assume that each robot has integrated object detection module and can accurately identify whether the interaction object is a human or another robot.

The total reward configuration is shown as follows:

$$r_{R_i} = \begin{cases} -1, & \text{if any}(\mathbf{d}_{\text{RRs}}) < 0 \text{ or any}(\mathbf{d}_{\text{RPs}}) < 0. \\ -F_{\text{scale}} \mid d_g^t \mid + r_{\text{time}} + r_{\text{comfort}} + r_{K_{\text{step}}}, & \text{otherwise.} \end{cases} \tag{2}$$

where $\mathbf{d}_{\mathrm{RRs}} = [\tilde{d}_{R_iR_1}, \ldots, \tilde{d}_{R_iR_j}]$ represents the relative distance vector between the robot i and all other robots $\{1, \ldots, j\}$ in its FOV. $\tilde{d}_{R_iR_j} = d_{R_iR_j} - r_{R_i} - r_{R_j}$ where r_{R_i} and r_{R_j} represent the radius of the safety zone of the robot i and the robot j. Similarly, $\mathbf{d}_{\mathrm{RPs}} = [\tilde{d}_{R_iP_1}, \ldots, \tilde{d}_{R_iP_m}]$ is the relative social distance vector between the robot i and other pedestrians $\{1, .., m\}$ in its FOV. $d_g^t = ||p_g^t - p_r^t||_2$ is the relative L2 distance between the robot i and the target point at timestep t. $F_{\mathrm{scale}} = \frac{1}{R_{\mathrm{env}}^2}$ is the scaling constants. R_{env} is the radius of the current scenario. $r_{\mathrm{time}} = -0.001$ is the time penalty. This item promotes the robot to reach the goal as quickly as possible. Also, $r_{\mathrm{comfort}} = -\mathbb{I} \cdot 0.5$. $\mathbb{I}$ indicates whether the robot i invades the social comfort zone of pedestrians at the current timestep.

Inspired by Liu et al. (2022), we also design a prediction-based reward function called K-step lookahead reward $r_{K_{\mathrm{step}}}$ to induce multiple robots to learn more socially reasonable collaborative motion strategy. As represented in Fig. 2, the K-step lookahead reward configuration between the robot i and the human j is as follows:

$$r_{K_{\mathrm{step}}}^{R_iP_j} = \begin{cases} \min_{tp=\{\mathrm{t}+1,\ldots,\mathrm{t}+\mathrm{K}\}} -e^{-k}, & \text{if } \min \mathbf{d}_{R_i'P_i'}^{tp} < d_{\mathrm{comfort}}. \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where t is the current timestep. $d_{\mathrm{comfort}} = 0.25$ is the comfort threshold of social distance. We consider that during the human–robot interaction, the intrusion action of robots can cause "discomfort" and deliberate avoidance behavior of pedestrians. $\mathbf{d}_{R_i'P_j'}^{tp} = [\tilde{d}_{R_i'P_j'}^{t+1}, \ldots, \tilde{d}_{R_i'P_j'}^{t+K}]$ represents the lookahead social distance vector between the robot i and pedestrian j in its FOV. The final K-step lookahead reward of the robot i is the minimum of all $r_{K_{\mathrm{step}}}^{R_iP_j}$ where $j = \{1, 2, \ldots, m\}$ represents the index of the pedestrian that appears in FOV of the robot i.

$$r_{K_{\mathrm{step}}} = \min_{j=\{1,2,\ldots,m\}} r_{K_{\mathrm{step}}}^{R_iP_j} \tag{4}$$

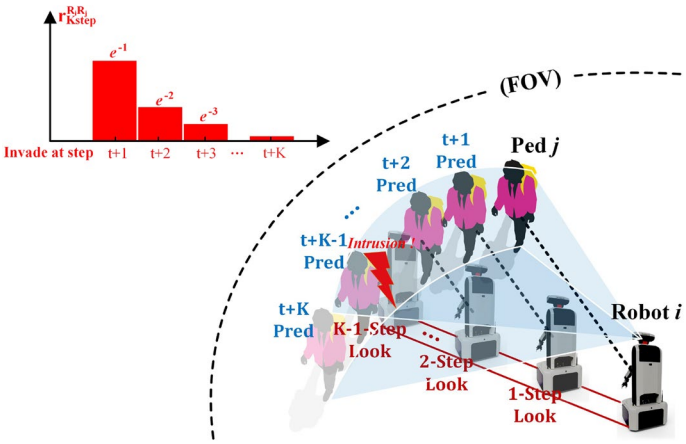

Fig. 2 At timestep t, robot i makes a K-step trajectory prediction for pedestrian j in its FOV, and preforms K-step uniform trajectory evolution on the basis of the current velocity. At each look-ahead step, it robot i virtually invades the comfort zone of pedestrian j or collides with it, a specific penalty signal is produced. The magnitude of this penalty decays exponentially with the increasing of lookahead step K

In addition, it is noteworthy that implementing the K-step lookahead reward setting relies on two key points. First, we need a pre-trained trajectory prediction module (e.g., Social-GAN (Gupta et al. 2018), MID (Gu et al. 2022), etc.) to predict the K-timestep trajectory evolution of pedestrians in FOV at each timestep in the lookahead virtual time domain. Second, for the motion evolution of the robot i in the lookahead virtual time domain, we choose to perform a uniform robot state update according to the current evaluated real action pair. This trick accelerates the training process and allows the algorithm to focus on quantifying the social reasonableness of the policy at the actual present timestep.

To sum up, our final joint reward setting of multi-robot social-aware cooperative planning task is as follows:

$$r_{\boldsymbol{R}} = \sum_{i=1}^{N} r_{R_i} \tag{5}$$

where N is the number of robots in the environment. This mode of joint reward decomposition specifies the contribution of each agent to the team and effectively attenuates the credit assignment issue in the MARL setting.

3.2 Algorithm description of MSA3C

The overall MSA3C algorithm architecture of our multi-robot social-aware cooperative planning method is shown in Fig. 3. MSA3C belongs to the CTDE paradigm. The CTDE paradigm-based co-planning inherits the advantages of centralized and decentralized methods, respectively (He et al. 2022b). We design the local TSG-based social encoder module to perform the social-interaction hidden state extraction and handle the problem of time-varying input dimension caused by the dynamic change of pedestrians in the limited FOV of each robot. By combining it with the parameter-sharing mechanism, the scalability of the planner is effectively improved. In addition, we utilize a multi-head global observation attention module as an alternative to the traditional concatenation centralized critic network (He et al. 2022b; Liang et al. 2021; Wang et al. 2020a). This approach provides a more focused and oriented evaluating mode of the decision-making importance of different social features, weakens the weight of irrelevant information in the global information, allows better aggregation of sensing information shared between robots, which can be leveraged to guide decentralized policy network updates better and more directional.

3.2.1 Rollout replay buffer

In the training phase, we need to push the joint interaction experience generated by all robots to the replaybuffer. Different from the previous approach of pushing transitions (He et al. 2022b), these data need to be partitioned and stored in a fixed timestep sequence owing to the utilization of GRU-based local TSG in the later module. As shown in Fig. 4, we design the rollout replaybuffer to deal with the rollout data and record padding positions used to align the timestep length in order to mask those pseudo-data in the subsequent loss back-propagation process.

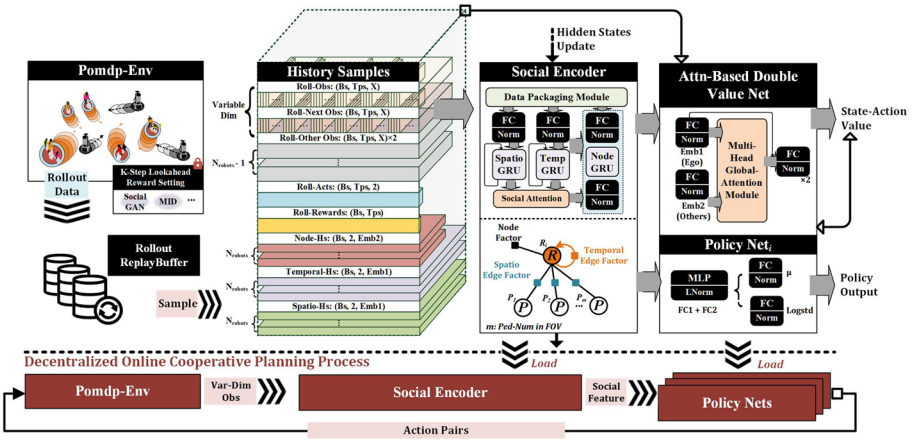

Fig. 3 The overall architecture of the Multi-robot Social-Aware Attention-based Actor-Critic (MSA3C) algorithm

3.2.2 Social encoder

As shown in Fig. 4, although we align the timestep length of each batch, the observation dimension is inconsistent due to the dynamic changing of the number of pedestrians per timestep in the limited FOV of each robot. So, the first step is to align the maximum observation dimension in each batch via the data packaging module in Fig. 3. Then, the encapsulated data are fed into the local TSG as the input of spatial-edge RNN, temporal-edge RNN, and node RNN, respectively.

The TSG is widely applied in the field of pedestrian trajectory prediction (Vemula et al. 2018; Chen et al. 2019; Huang et al. 2019a). In this paper, we utilize part of this graph to extract the human–robot interaction feature, and take this social feature as input to the MARL module. The social feature contains historical information and can reflect the potential importance of each human in FOV for the robot to make the next decision. The spatial-edge RNN of the TSG is responsible for capturing the spatial information, such as the relative orientation and distance of each robot with respect to other agents (e.g., humans, other robots, etc.) in its limited FOV. The specific procedure is as follows:

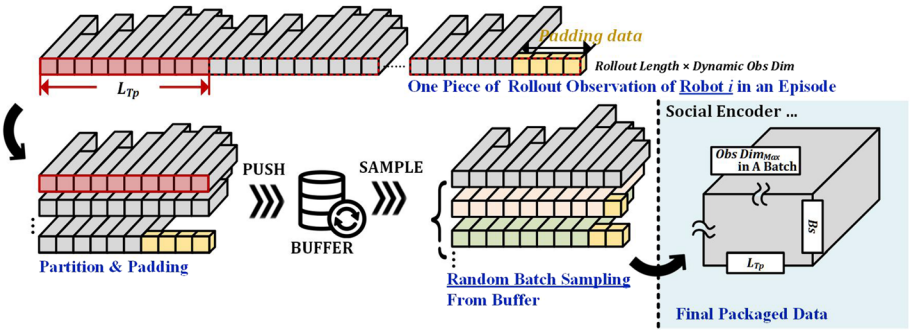

Fig. 4 Rollout data processing pipeline: from rollout replaybuffer to the social encoder

$$h^t_{r_i\mathbf{A}} = GRU(h^{t-1}_{r_i\mathbf{A}}, \phi(\mathbf{x}^t_{r_i\mathbf{A}}; W^{\text{emb}}_{\text{spatial}}); W^{\text{h}}_{\text{spatial}}) \tag{6}$$

where $\mathbf{x}^t_{r_i\mathbf{A}}$ represents the relative position vector between robot i and all other agents in its FOV at timestep t. ϕ is the embedding layer, including a fully connected (FC) layer and a layer normalization (LN) layer. $W^{\text{emb}}_{\text{spatial}}$ are the embedding weights. $h^t_{r_i\mathbf{A}}$ is the hidden state of the GRU at timestep t. $W^{\text{h}}_{\text{spatial}}$ are the weights of spatial-edge RNN.

Temporal-edge RNN is responsible for capturing the position evolution representation of the current robot in adjacent frames. The specific process is similar to (6):

$$h^t_{r_ir_i} = GRU(h^{t-1}_{r_ir_i}, \phi(\mathbf{x}^t_{r_ir_i}; W^{\text{emb}}_{\text{temporal}}); W^{\text{h}}_{\text{temporal}}) \tag{7}$$

where $\mathbf{x}^t_{r_ir_i}$ is the position changing vector of the robot i at adjacent timesteps. Then, we put $h^t_{r_ir_i}$ and $h_{r_i\mathbf{A}}$ over a dot product multi-head social attention module to obtain the attention scores of different human–robot interactions. The specific process is shown as follows:

$$\mathbf{x}^t_{\text{attn}_{r_i}} = \text{softmax}(F_{\text{scale}} W_q h^t_{r_i\mathbf{A}} h^{tT}_{r_ir_i} W^T_k) \cdot (W_v h_{r_i\mathbf{A}}) \tag{8}$$

where F_{scale} is the scale constant. W_q are the Query weights, W_k are the Key weights, and W_v are the Value weights. Finally, we concatenate $\mathbf{x}_{\text{attn}_{r_i}}$ with the ego state of robot i and send them to the node RNN after embedding operation. The embedding feature is passed through the final feature extraction layer to generate the fixed-length social feature:

$$h^t_{r_i} = GRU(h^{t-1}_{r_i}, \phi(\text{cat}[\mathbf{x}^t_{\text{attn}_{r_i}}, \mathbf{x}^t_{r_i}]; W^{\text{emb}}_{\text{node}}); W^{\text{h}}_{\text{node}}) \tag{9}$$

$$\mathbf{x}^t_{\text{social}_i} = \phi(h^t_{r_i}; W^{\text{emb}}_{\text{social}}) \tag{10}$$

$\mathbf{x}^t_{\text{social}_i}$ would be fed to the MARL part in the later stage.

3.2.3 Multi-robot global attention-based actor-critic

Our MSA3C is proposed on the basis of our previous multi-agent local-and-global attention actor-critic (MLGA2C) in Song et al. (2022). Here, we replace the local attention module in MLGA2C with a social encoder to better help the robot understand human–robot interaction in FOV to improve the safety of planning in human crowds. The final training paradigm of is shown in Fig. 5, where

The final part of MSA3C integrates a global-attention-based critic network and a actor network. Different homogeneous robots are parameter-sharing to improve training efficiency. The attention-based double value network receives concatenation tensors $e^i_{\text{social}} = [\mathbf{x}_{\text{social}_i}, \boldsymbol{a}_i](i = 1, \ldots, N)$ from different robots. Next, the processes of social feature embedding for the robot i and other robots $-i$ are performed via (11), (12).

$$e^i_{\text{self}} = \phi_{\text{self}}\left(\text{cat}\left[\mathbf{x}^i_{\text{social}}, \boldsymbol{a}^i\right]; W^{\text{emb}}_{\text{self}}\right) \tag{11}$$

$$e^{-i}_{\text{other}} = \phi_{\text{other}}\left(\text{cat}\left[\mathbf{x}^i_{\text{social}}, \boldsymbol{a}^i\right]; W^{\text{emb}}_{\text{other}}\right) \tag{12}$$

The purpose of global attention computation between robots in a centralized value network is to capture the most critical shared inter-robot feature for the current robot. This

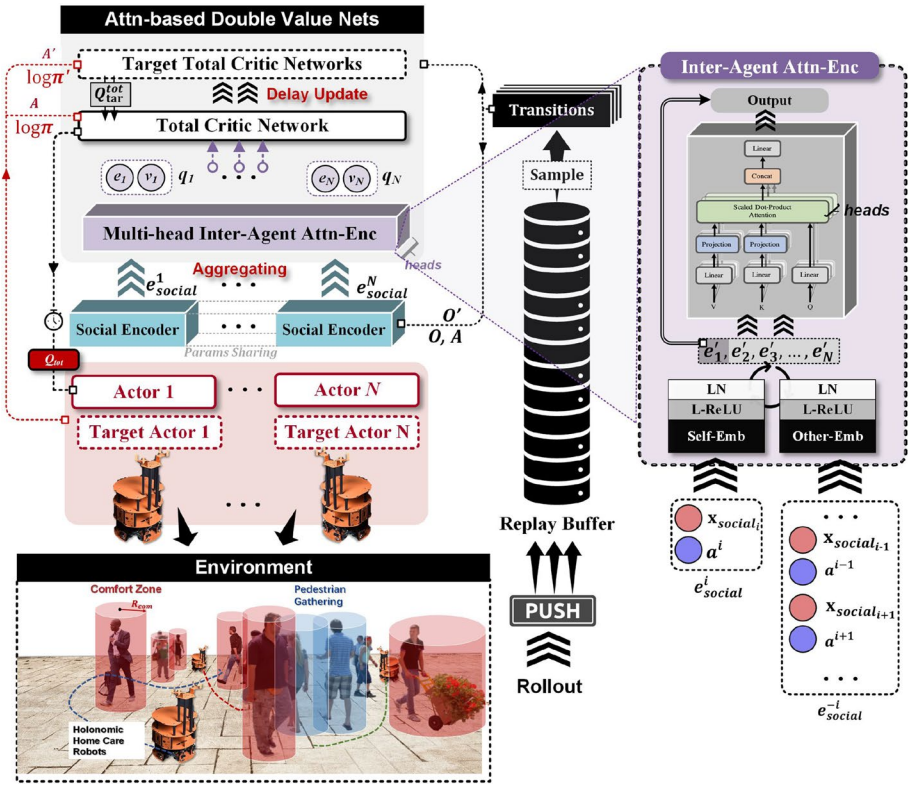

Fig. 5 The training details of the global-attention-based actor-critic

information aggregation pattern facilitates more accurate decision-making by the current robot.

Specifically, for the robot i, the joint action state evaluation Q can be described as follows:

$$Q_{r_i} = \phi_{\text{emb}}\left(\left[\text{softmax}(W_q^{\text{RL}} e^i_{\text{self}} (W_k^{\text{RL}} e^{-i}_{\text{other}})^T)) \cdot (W_v^{\text{RL}} e^i_{\text{self}}), e^i_{\text{self}}\right]\right) \tag{13}$$

where $W_q^{\text{RL}}, W_k^{\text{RL}}, W_v^{\text{RL}}$ represent the weights of the dot product attention module between robots. The output value of the final layer is the joint state-action Q value of the robot i. The loss function of our centralized Q is defined as follows:

$$L_Q(\psi) = \sum_{i=1}^{N} \mathbb{E}_{(\boldsymbol{o},\boldsymbol{a},\boldsymbol{r},\boldsymbol{o}',\boldsymbol{h}_{\text{temp}},\boldsymbol{h}_{\text{spa}},\boldsymbol{h}_{\text{node}})\sim D}\left[\sum_{j=\{1,2\}}\left(Q^{\psi_j}_{r_i} - y_{r_i}\right)^2\right] \tag{14}$$

where

$$y_{r_i} = \mathbb{E}_{\boldsymbol{a}'\sim\pi_\theta\left(\mathbf{x}'_{\text{social}_i}\right)}\left[\boldsymbol{r} + \gamma\left(-\alpha \log\left(\pi_\theta\left(\boldsymbol{a}'_i \mid \mathbf{x}'_{\text{social}_i}\right)\right) + \min\left(Q^{\bar{\psi}_1}_{r_i}, Q^{\bar{\psi}_2}_{r_i}\right)\right)\right] \tag{15}$$

where D represents our rollout replaybuffer. To avoid the overestimation problem of classical deep Q learning, we introduce two separated Q heads $Q^{\psi_1}_{r_i}, Q^{\psi_2}_{r_i}$ and corresponding target

Q heads $Q_{r_i}^{\bar{\psi}_1}, Q_{r_i}^{\bar{\psi}_2}$. π_θ is the policy network with shared parameters. y_{r_i} is the soft TD target value. We have the same policy entropy setting about y_{r_i} as in the previous paper (He et al. 2022b). γ is the discount factor. α is a trainable parameter that can be used to auto-adjust the exploration degree of the robot during the training phase.

The update of the policy network is consistent with our previous work (Song et al. 2022):

$$\begin{aligned}\nabla_\theta J(\pi_\theta) = \sum_{i=1}^{N} \mathbb{E}_{(\boldsymbol{o},\boldsymbol{a},\boldsymbol{h}_{\text{temp}},\boldsymbol{h}_{\text{spa}},\boldsymbol{h}_{\text{node}})\sim D,\boldsymbol{a}_i\sim\pi_\theta}\Big[\nabla_\theta \log \pi_\theta \\ \times \left(-\alpha \log \pi_\theta + \min\left(Q_{r_i}^{\psi_1}, Q_{r_i}^{\psi_2}\right)\right)\Big]\end{aligned} \tag{16}$$

Also, the reparameterization trick (Duan et al. 2021) are deployed here to ensure the continuity of the gradient back-propagation process::

$$\boldsymbol{a}_i \sim \pi_\theta = \tanh\left(\mu_\theta + \sigma_\theta \odot \varepsilon\right), \varepsilon \sim \mathcal{N}(0, I) \tag{17}$$

where μ_θ and σ_θ are the outputs of the policy network. Thus the expectation for action commands is rewritten as an expectation for the standard Gaussian noise ε, which ensures the derivable property of the policy network.

3.2.4 Decentralized cooperative planning

The decentralized execution of our MSA3C-based cooperative planning methods is much easier. As shown in Fig. 3, each robot utilizes its local observation to make decisions, and this process can be online and in real time. In the realistic deployment stage, this cooperative planner can be combined with the model predictive controller or others to achieve the autonomous and collaborative operation of multiple robots.

4 Experiments and discussion

4.1 Experimental configuration

4.1.1 Basic setting

To verify the performance of MSA3C, we have developed a Python-based gym simulation platform for multi-robot cooperative planning in pedestrians based on Everett et al. (2018) and Chen et al. (2019). As shown in Fig. 1, we utilize the circle to represent the safety zone and the social comfort zone of pedestrians. At the beginning of each episode, we randomly assign the safety radius, the social comfort radius, and the preferred velocity to simulate different numbers, body shapes, and motion states of each pedestrian unit. We assign each pedestrian a random goal change probability parameter with the value range [0.2, 0.3] to simulate the motion uncertainty of humans during the training phase. Also, we add timestep-varying random noise to the safety radius of each pedestrian unit to simulate the sensing error of the robot. Unlike previous work (Chen et al. 2019), we utilize social-force (SF) model to simulate the movement and interaction of pedestrians. SF can describe the self-organization of several group effects of the observed pedestrian behavior more

realistically (Mehran et al. 2009). In addition, we adopt a non-collaborative human–robot interaction mode. Pedestrians are defaulted to be the influencer, and robots are the reactor. This setting prevents robots from learning aggressive cooperative planning policies to obtain high returns (Liu et al. 2022). It is worth mentioning that we adopt a multi-stage training scheme. In stage I, our goal is to encourage multiple robots to learn a high-efficiency cooperative pattern. In stage II, we introduce the K-step prediction reward setting to develop the comfort social awareness of each robot while employing a larger collision penalty (-5) to ensure the planning safety.

All experiments are performed on a server with an Intel(R) Xeon(R) Silver 4214CPU and a GeForce RTX 3090 GPU. We have summarized the basic parameter setting in Table 1.

4.1.2 Network setting of MSA3C

MSA3C consists of the social encoder, the attention-based double value net, and the policy net. The social encoder including the temporal-edge RNN with the size of [2,64,256], the spatial-edge RNN with the unit dimension of [2,64,256], the multi-head social attention block with three embedding layers (W_q, W_k, W_v) of dimension [256,128], and the node RNN with the unit dimension [7,64,128,256]. Each RNN module is equipped with the LayerNorm layer and the LeakyReLU activation function. The global attention-based double value net contains two critic heads. Each critic head is composed of the ego-embedding layer with the size of [258, 128], the others-embedding layer with the size of [258, 128], the multi-head global attention block with three embedding layers ($W_q^{\mathrm{RL}}, W_k^{\mathrm{RL}}, W_v^{\mathrm{RL}}$) of dimension [128, 128], and the Q value output block with the unit dimension [256, 128, 128, 1]. Similarly, each sub-block of the global critic head is equipped with the Layernorm layer and the LeakyReLU activation function. The decentralized policy network contains three hidden FC layers with the dimension of [256,128,128,128,2].

4.2 Metrics

To quantify the performance of different algorithms, we set the following evaluation metrics for the pedestrian participation co-planning task inspired by Chen et al. (2019), Zhou et al. (2021) and Liu et al. (2022):

Table 1 The basic parameter setting of the environment and MSA3C

Env setting	Value	RL setting	Value
r^P_{Safety}	0.5–1.3 m	$K_{\mathrm{lookahead}}$	5
r^R_{Safety}	0.6 m	r_{time}	− 1e−3
v^P_{pref}	0.5–1.5 m/s	lr	5e−4
v^R_{pref}	1 m/s	τ	0.01
d_{comfort}	0.25 m	Policy delay	2
R_{scenario}	[6 m, 8 m, 10 m]	Batch size	256
N_{peds}	5–20	α_{init}	0.02
N_{robots}	3	Buffer size	2e5
FOV	2π, [5 m, 10 m]	episode	5e4
Timestep	0.25 s	$l_{\mathrm{rollout\text{-}tps}}$	10

1. "CSR": Success rate of the co-planning process. We specify that all robots reach their target points within the limited timesteps to be considered as a "success".
2. "CR": Collision rate. CR describes the probability that the robot collides with other agents.
3. "APL": Average co-planning path length of all robots. This metric describes the cooperative capability of different algorithms.
4. "NTC": N-robot co-planning timestep consumption. This metric describes the cooperative efficiency of different algorithms.
5. "CIR": Comfort zone intrusion rate. This metric reflects the social awareness of robots during the co-planning process.

4.3 Quantitative and qualitative analysis

Table 2 summarizes the performance of different algorithms. "NpMr" represents that there are "N" pedestrians and M robots in the current scenario. "FX" represents the current FOV of each robot is "X"m. For each "NpMr" scenario, we randomly run 500 times against different algorithms. Each algorithm is run with the same random seed to ensure the consistency of the pedestrian randomness and the same initial positions of robots. Most importantly, we specify 150 limited timesteps for each scenario to compare the efficiency of different algorithms.

Table 2 Summary table of quantitative experiment results in different pedestrian participation scenarios with limited timesteps (150)

Algorithms	Scene	Different Metrics				
		CSR%↑	CR%↓	APL↓	NTC↓	CIR%↓
MASAC-F10	5p3r	Nan	Nan	Nan	Nan	Nan
MLGA2C-F10	5p3r	98.0	44.2	26.8	45.7	10.2
MSA3C-F10	5p3r	92.6	0.9	27.7	51.4	3.0
	10p3r	80.2	2.2	31.8	71.6	4.1
	20p3r	61.4	5.8	36.9	81.4	6.7
MSA3CPred-F10	5p3r	98.8	2.4	28.7	51.7	1.2
	10p3r	99.8	2.5	33.1	65.2	1.6
	20p3r	92.6	8.4	41.2	81.0	2.9
MSA3CPred-F5	5p3r	99.2	11.6	34.2	65.8	2.3
	10p3r	97.2	14.9	40.2	79.3	2.7
	20p3r	89.5	16.6	43.6	84.3	3.8
SARL-F10*	5p3r	85.2	12.2	40.4	56.5	4.9
ORCA-PS-F10*	5p3r	98.4	2.8	30.2	66.4	1.4
	10p3r	92.4	3.5	38.2	89.7	1.6
	20p3r	60.4	7.8	39.5	102.9	2.5
SF-F10*	5p3r	98.0	35.5	33.1	80.8	7.3
	10p3r	68.2	77.9	37.1	89.4	9.6
	20p3r	71.2	135.1	42.9	99.0	12.5

"*" means that this algorithm belongs to the decentralized or distributed algorithm

Fig. 6 Collaborative trajectories of multiple robots generated by different co-planning algorithms in a pedestrian interaction scenario with fixed random seeds. The red curve represents the pedestrian motion trajectory, and the black curve represents the robot motion trajectory

4.3.1 Baselines

We set up multiple baseline algorithms for comparison. We first select off-policy MASAC under local observation conditions as the MARL architecture. The MASAC-based co-planning method has achieved good collaborative performance in pure robot environments with the naive concatenation global information aggregation pattern He et al. (2022a). MLGA2C is a recent work that achieving SOTA cooperative performance in dense pure robot scenarios Song et al. (2022). SARL in Zhou et al. (2021) is a representative decentralized social aware planning method with the single agent RL (SARL) setting. We utilize this method to verify the effect of offline multi-robot information aggregation on improving the overall co-planning performance. ORCA-PS-F10 is the typical optimal reactive-based multi-robot co-planning method under the local perfect sensing condition in Berg et al. (2011). SF stands for the classical social force approach Mehran et al. (2009).

4.3.2 Quantitative analysis

First, we would analyze the primary 5p3r group. Since MASAC cannot converge in this pedestrian participation environment setting, the detailed data on each metric cannot be output. A comparison on APL and NTC metrics shows that our previous global attention critic-based MLGA2C still obtains excellent cooperative performance He et al. (2022a). However, the simple proximity function cannot correctly inspire the robot to learn to handle social relationships with uncertain pedestrians in its FOV. Higher CR% and CIR% imply that the robot develops an aggressive policy with the primary target of goal-reaching. As for SARL-F10, we adopt a recent decentralized social aware planning method in Zhou et al. (2021). Although SSR% and CIR% performance of SARL-F10 is not bad, the lack of global information makes it cannot achieve good performance in cooperative metrics, like APL and NTC. Our MSA3Cs under different settings not only both have great CSR% and lower CR% compared to other baselines, but also produce great a better cooperative performance on the metrics of APL and NTC between multiple robots. Most importantly, with the help of the TSG-based social encoder and the K-step lookahead reward function, our MSA3Cs help robots generate more reasonable social awareness, which leads to lower CR% and CIR%.

In the more complex and dense 10p3r and 20p3r scenarios, basic MLGA2C fails to converge. With the introduction of our TSG-based social encoder, MSA3C-F10 can aid robots in making more socially safe decisions. After further coupling the K-step lookahead reward function, MSA3CPred-F10 performs well in different metrics, especially in CIR%. This means robots under this setting generate social comfort awareness. We also find that the K-step lookahead trick motivates robots to master the skill of avoiding dense social interaction regions. This skill avoids freezing robot issue and help the full version of MSA3CPred achieve higher CSR% in limited timesteps compared to other MARL-based and reactive-based methods.

Furthermore, we narrow the sensing range of each robot to 5m. The results of MSA3CPred-F5 show that the global attention module-based MARL architecture can still effectively aggregate the limited local observation of each robot and help multiple robots

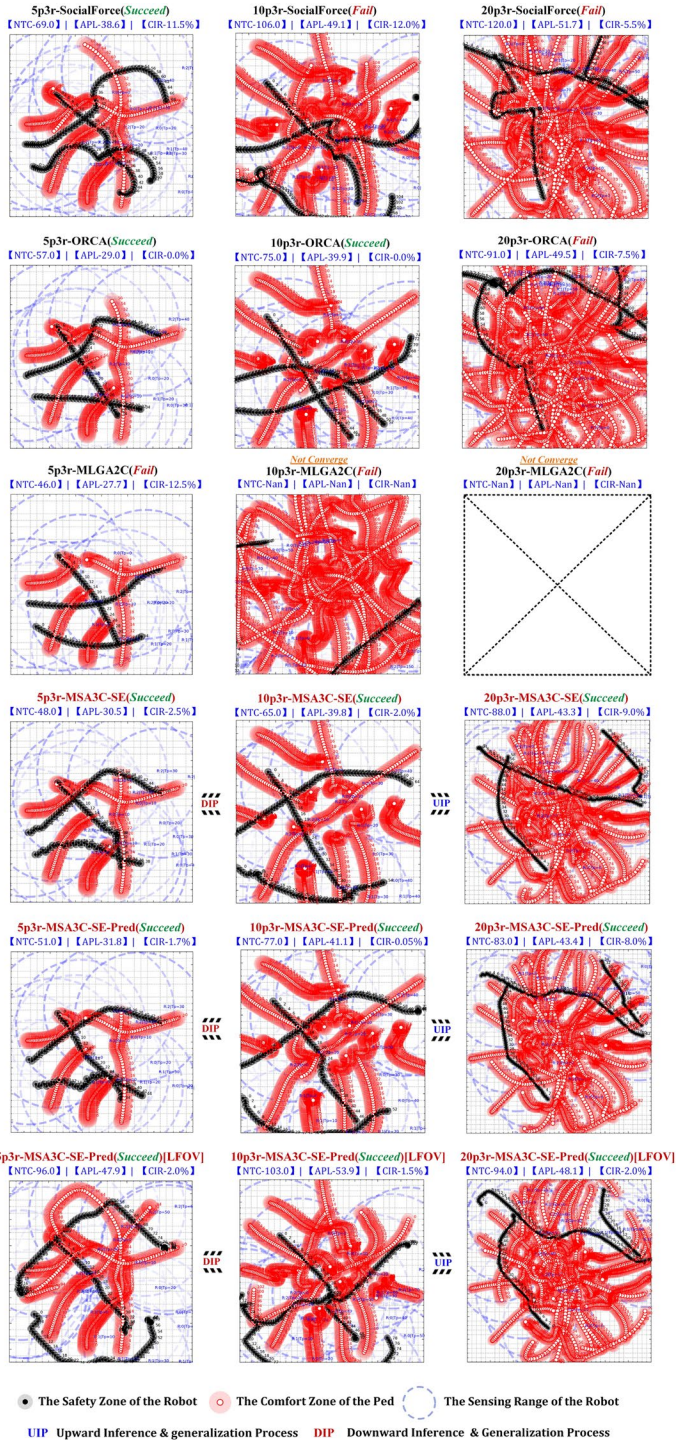
5p3r-SocialForce(Succeed)
【NTC-69.0】|【APL-38.6】|【CIR-11.5%】
10p3r-SocialForce(Fail)
【NTC-106.0】|【APL-49.1】|【CIR-12.0%】
20p3r-SocialForce(Fail)
【NTC-120.0】|【APL-51.7】|【CIR-5.5%】
5p3r-ORCA(Succeed)
【NTC-57.0】|【APL-29.0】|【CIR-0.0%】
10p3r-ORCA(Succeed)
【NTC-75.0】|【APL-39.9】|【CIR-0.0%】
20p3r-ORCA(Fail)
【NTC-91.0】|【APL-49.5】|【CIR-7.5%】
5p3r-MLGA2C(Fail)
【NTC-46.0】|【APL-27.7】|【CIR-12.5%】
Not Converge
10p3r-MLGA2C(Fail)
【NTC-Nan】|【APL-Nan】|【CIR-Nan】
Not Converge
20p3r-MLGA2C(Fail)
【NTC-Nan】|【APL-Nan】|【CIR-Nan】
5p3r-MSA3C-SE(Succeed)
【NTC-48.0】|【APL-30.5】|【CIR-2.5%】
10p3r-MSA3C-SE(Succeed)
【NTC-65.0】|【APL-39.8】|【CIR-2.0%】
20p3r-MSA3C-SE(Succeed)
【NTC-88.0】|【APL-43.3】|【CIR-9.0%】
DIP
UIP
5p3r-MSA3C-SE-Pred(Succeed)
【NTC-51.0】|【APL-31.8】|【CIR-1.7%】
10p3r-MSA3C-SE-Pred(Succeed)
【NTC-77.0】|【APL-41.1】|【CIR-0.05%】
20p3r-MSA3C-SE-Pred(Succeed)
【NTC-83.0】|【APL-43.4】|【CIR-8.0%】
DIP
UIP
5p3r-MSA3C-SE-Pred(Succeed)[LFOV]
【NTC-96.0】|【APL-47.9】|【CIR-2.0%】
10p3r-MSA3C-SE-Pred(Succeed)[LFOV]
【NTC-103.0】|【APL-53.9】|【CIR-1.5%】
20p3r-MSA3C-SE-Pred(Succeed)[LFOV]
【NTC-94.0】|【APL-48.1】|【CIR-2.0%】
DIP
UIP
The Safety Zone of the Robot
The Comfort Zone of the Ped
The Sensing Range of the Robot
UIP Upward Inference & generalization Process
DIP Downward Inference & Generalization Process

Table 3 Summary table of ablation experiment results in different pedestrian participation scenarios with limited timesteps (150)

Scene	Module setting			Different metrics				
	-Attn	-TSG	-Pred	CSR%↑	CR%↓	APL↓	NTC↓	CIR%↓
5p3r	✓	✗	✗	95.6	29.2	30.8	52.5	6.6
10p3r	✓	✗	✗	54.5	45.9	39.8	78.7	15.5
20p3r	✓	✗	✗	Nan	Nan	Nan	Nan	Nan
5p3r	✓	✓	✗	92.6	1.9	27.7	51.4	3.0
10p3r	✓	✓	✗	80.2	2.2	31.8	71.6	4.1
20p3r	✓	✓	✗	61.4	7.8	36.9	81.4	6.7
5p3r	✓	✓	✓	98.8	2.4	28.7	51.7	1.2
10p3r	✓	✓	✓	99.8	2.5	33.1	65.2	1.6
20p3r	✓	✓	✓	92.6	8.4	41.2	91.0	2.9

obtain decent collaborative performance on CSR%, APL, and NTC. The TSG-based social encoder and K-step lookahead trick help robots to maintain efficient co-planning ability while still achieving a good level of social safety and social comfort on CR% and CIR% metrics in the human–robot interacting process.

4.3.3 Ablation results

In this subsection, we conducted multiple groups of ablation experiments to further verify the effectiveness of each critical sub-module in MSA3C including attention-based double value net (-Attn), temporal spatial graph-based social encoder (-TSG), and K-step lookahead reward setting (-Pred). The results are summarized in Table 3. First, we can notice that compared to the MASAC baseline in Table 2, the introduction of the 'Attn' module can help the co-planner achieves better cooperating social navigation performance in the 5p3r task environment, which proves that the inter-agent attention encoder aggregator between different robots is much more effective than the naive concatenation form in MASAC. Furthermore, with the introduction of the TSG-based social encoder which can well handle the social relation between the robot and other agents, the co-planner achieve a higher CSR% and better overall social co-planning performance in 5p3r and 10p3r task environments. But, in the denser 20p3r scenario, the limitations of the short-sighted reward setting severely impact the final planning results. Finally, with the introduction of 'Pred' module, the co-planner achieves great CSR% in different pedestrian density task scenarios. Moreover, the obvious decrease in CIR% metric indicates that the robot has learned a more social friendly collaborative planning capability. Indeed, the slight increase in APL and NTC metrics, as the robot avoids social conflict zones by taking detours, is within expectations.

4.3.4 Qualitative analysis

The co-planning trajectories of robots driven by different algorithms in a random scenario are shown in Fig. 6. Meanwhile, we extract one critical frame from this co-planning process to visualize the effect of the pedestrian trajectory prediction module-based K-step lookahead reward function on the human–robot social interaction process in Fig. 7. It is necessary to state that these plots are generated with the same random seed.

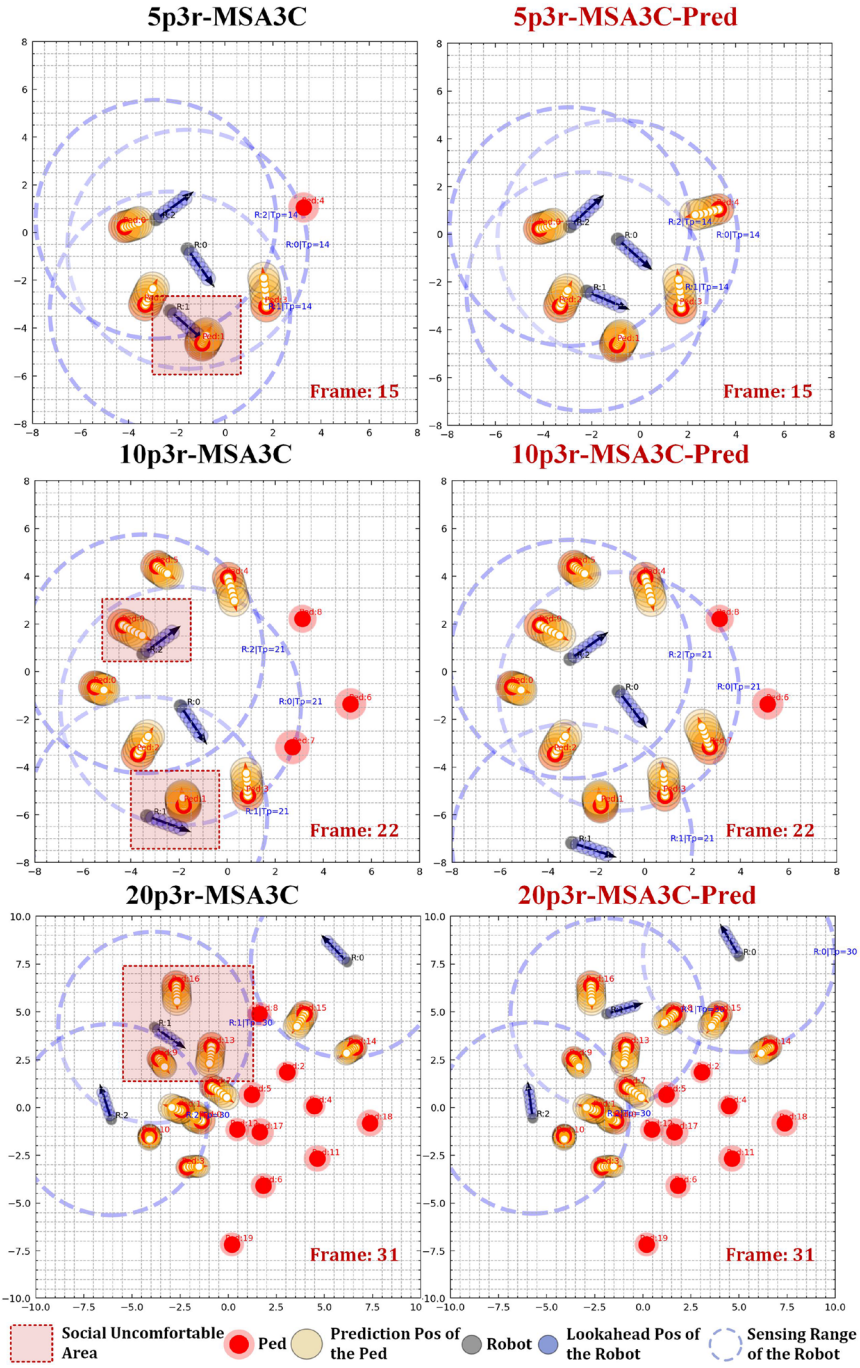

Fig. 7 The social relationship plots and the visualization of K-step prediction of human comfort zones for MSA3C and full version MSA3CPred at specific frame extracted from Fig. 6

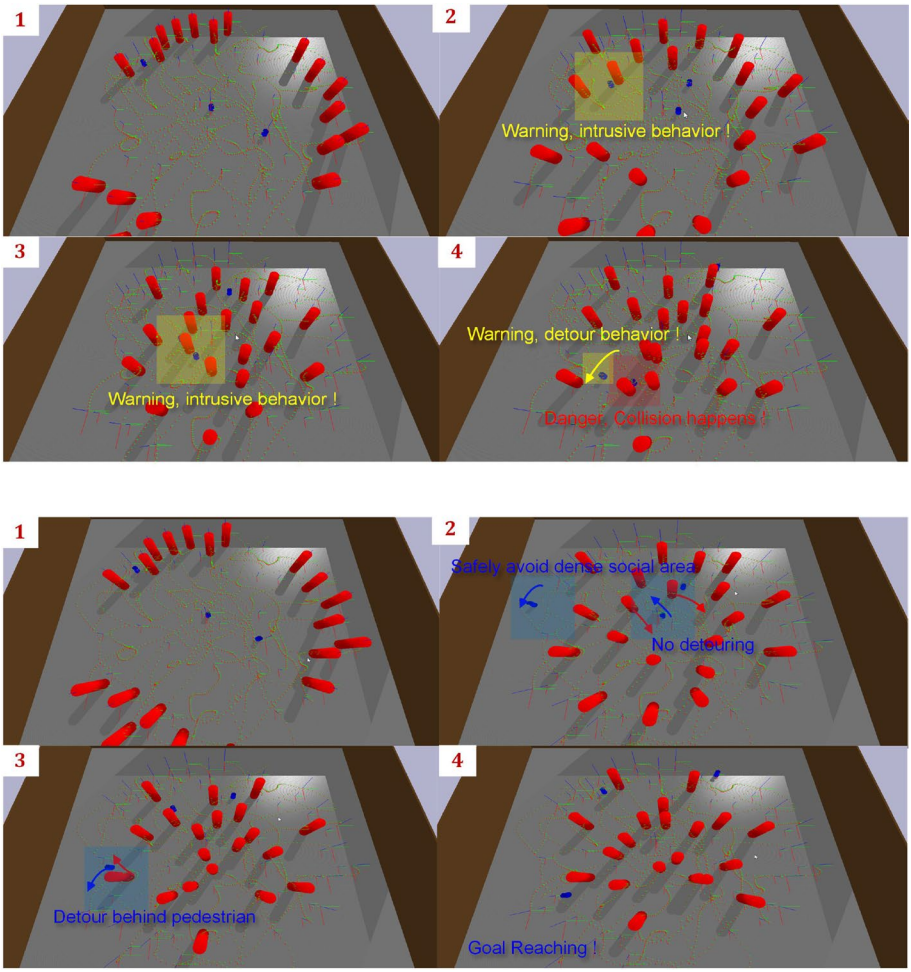

Fig. 8 Comparison results of the co-planning effects of MSA3C and MSA3CPred in 20p3r Pybullet simulation scenario

The denser the red trajectories of pedestrians, the more timesteps robots consume for the co-planning process. First, we can find that the MLGA2C group generates co-planning trajectories with better smoothness and straightness but fails to complete these three tests without collision. Reactive-based methods are weak in generalizing across different scenarios and have inconsistent performance. The other three MSA3C-based settings help robots complete co-planning without human–robot collision. Further study of the trajectories reveals that MSA3CPred can help the robot adjust its policy to avoid dense social interaction regions during the planning process. Figure 7 can illustrate this conclusion more visually. We can find that the full version MSA3CPred helps robots produce more farsighted policies in the scenarios of 5p3r, 10p3r, and 20p3r. These motion policies not only guarantee the social safety of robots in the present timestep but also ensures robots avoid entering the K-step prediction comfort zone of pedestrians in the future prediction time domain. This mechanism also effectively motivates robots to escape from dense

social interaction areas prone to freezing issues. In addition, combing with the results in Table 2, we can find that MSA3C trained in 10p3r environment effectively generalizes to 5p3r and 20p3r scenarios and maintains better social co-planning performance compared to other baselines. This further validates the scenario scalability of MSA3C. To sum up, these above qualitative analyses are highly consistent with the conclusions drawn in quantitative experiments.

Further, we perform 20p3r co-planning inference tests for MSA3C and MSA3CPred in Pybullet physical engine simulator. The final result in Fig. 8 shows that without K-step prediction module, robots are inclined to make non social-aware decisions such as comfort zone intrusion, dense social area intrusion, and forced detour. In contrast, MSA3CPred co-planner has managed to circumvent these issues. This inference is further validated by the 2.66% CIR social performance compared to 6.77% of MSA3C in this scenario.

5 Conclusion

We present a brand new social aware multi-robot cooperative planning method MSA3C based on MARL architecture with an attention mechanism. The algorithm generally follows the CTDE paradigm and introduces the multi-head attention-based centralized critic network to better aggregate the local information from each robot. By replacing the simple proximity function with a TSG-based social encoder, our model allows each robot to better understand the social importance of each pedestrian in its FOV. Also, we introduce the K-step lookahead reward setting to alleviate the intrusion of the robot into the social comfort zone of pedestrians and avoid the short-sighted decision-making process of each robot. Through quantitative and qualitative analyses in multiple pedestrian participation co-planning scenarios, we show that our MSA3C outperforms multiple baselines.

Author contributions LD wrote the main manuscript text, ZH and CS designed the algorithmic architecture and conducted related experiments, XY and HZ performed data processing and experimental data proofreading.

Funding Funding was provided by National Science and Technology Major Project (Grant No. 2021ZD0112700), the National Natural Science Foundation of China (Grant No. 62173251), Fundamental Research Funds for the Central Universities (Grant No. 2242023K30034), and the "Zhishan" Scholars Programs of Southeast University.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Berg JVD, Guy SJ, Lin M, Manocha D (2011) Reciprocal n-body collision avoidance. In: Robotics research. Springer, Berlin, pp 3–19. https://doi.org/10.1007/978-3-642-19457-3_1

Chen C, Liu Y, Kreiss S, Alahi A (2019) 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. https://doi.org/10.1109/ICRA.2019.8794134

Desaraju VR, How JP (2011) Decentralized path planning for multi-agent teams in complex environments using rapidly-exploring random trees. In: 2011 IEEE international conference on robotics and automation, pp 4956–4961. https://doi.org/10.1109/ICRA.2011.5980392

Dong L, He Z, Song C, Sun C (2023) A review of mobile robot motion planning methods: from classical motion planning workflows to reinforcement learning-based architectures. J Syst Eng Electron 34(2):439–459

Douthwaite JA, Zhao S, Mihaylova LS (2018) A comparative study of velocity obstacle approaches for multi-agent systems. In: 2018 UKACC 12th international conference on control (CONTROL), pp 289–294. https://doi.org/10.1109/CONTROL.2018.8516848

Duan J, Guan Y, Li SE, Ren Y, Sun Q, Cheng B (2021) Distributional soft actor-critic: off-policy reinforcement learning for addressing value estimation errors. IEEE Trans Neural Netw Learn Syst 33(11):6584–6598

Everett M, Chen YF, How JP (2018) Motion planning among dynamic, decision-making agents with deep reinforcement learning. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 3052–3059. https://doi.org/10.1109/IROS.2018.8593871

Everett M, Chen YF, How JP (2021) Collision avoidance in pedestrian-rich environments with deep reinforcement learning. IEEE Access 9:10357–10377. https://doi.org/10.1109/ACCESS.2021.3050338

Fan T, Long P, Liu W, Pan J (2020) Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. Int J Robot Res 39(7):856–892

Gu T, Chen G, Li J, Lin C, Rao Y, Zhou J, Lu J (2022) Stochastic trajectory prediction via motion indeterminacy diffusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 17113–17122

Gupta A, Johnson J, Fei-Fei L, Savarese S, Alahi A (2018) Social GAN: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)

He Z, Dong L, Sun C, Wang J (2022) Asynchronous multithreading reinforcement-learning-based path planning and tracking for unmanned underwater vehicle. IEEE Trans Syst Man Cybern Syst 52(5):2757–2769. https://doi.org/10.1109/TSMC.2021.3050960

He Z, Dong L, Song C, Sun C (2022) Multiagent soft actor-critic based hybrid motion planner for mobile robots. In: IEEE transactions on neural networks and learning systems (to be published). https://doi.org/10.1109/TNNLS.2022.3172168

Huang Y, Bi H, Li Z, Mao T, Wang Z (2019) Stgat: modeling spatial-temporal interactions for human trajectory prediction. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV)

Huang X, Zhou L, Guan Z, Li Z, Wen C, He R (2019) Generalized reciprocal collision avoidance for nonholonomic robots. In: 2019 14th IEEE conference on industrial electronics and applications (ICIEA), pp 1623–1628. https://doi.org/10.1109/ICIEA.2019.8834353

Liang Z, Cao J, Lin W, Chen J, Xu H (2021) Hierarchical deep reinforcement learning for multi-robot cooperation in partially observable environment. In: 2021 IEEE third international conference on cognitive machine intelligence (CogMI), pp 272–281. https://doi.org/10.1109/CogMI52975.2021.00042

Liu S, Chang P, Huang Z, Chakraborty N, Liang W, Geng J, Driggs-Campbell K (2022) Socially aware robot crowd navigation with interaction graphs and human trajectory prediction. arXiv preprint arXiv:2203.01821

Liu S, Chang P, Liang W, Chakraborty N, Driggs-Campbell K (2021) Decentralized structural-RNN for robot crowd navigation with deep reinforcement learning. In: 2021 IEEE international conference on robotics and automation (ICRA), pp 3517–3524. https://doi.org/10.1109/ICRA48506.2021.9561595

Matsuzaki S, Hasegawa Y (2022) Learning crowd-aware robot navigation from challenging environments via distributed deep reinforcement learning. In: 2022 International conference on robotics and automation (ICRA), pp 4730–4736. IEEE

Mehran R, Oyama A, Shah M (2009) Abnormal crowd behavior detection using social force model. In: 2009 IEEE conference on computer vision and pattern recognition, pp 935–942. IEEE

Mellinger D, Kushleyev A, Kumar V (2012) Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams. In: 2012 IEEE international conference on robotics and automation, pp 477–483. https://doi.org/10.1109/ICRA.2012.6225009

Nishimura M, Yonetani R (2020) L2B: learning to balance the safety-efficiency trade-off in interactive crowd-aware robot navigation. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 11004–11010. https://doi.org/10.1109/IROS45743.2020.9341519
Phillips M, Likhachev M (2011) SIPP: safe interval path planning for dynamic environments. In: 2011 IEEE international conference on robotics and automation, pp 5628–5635. IEEE
Qiu Q, Yao S, Wang J, Ma J, Chen G, Ji J (2022) Learning to socially navigate in pedestrian-rich environments with interaction capacity. arXiv preprint arXiv:2203.16154
Qureshi AH, Miao Y, Simeonov A, Yip MC (2021) Motion planning networks: bridging the gap between learning-based and classical motion planners. IEEE Trans Robot 37(1):48–66. https://doi.org/10.1109/TRO.2020.3006716
Rivière B, Hönig W, Yue Y, Chung S-J (2020) Glas: global-to-local safe autonomy synthesis for multi-robot motion planning with end-to-end learning. IEEE Robot Autom Lett 5(3):4249–4256. https://doi.org/10.1109/LRA.2020.2994035
Sartoretti G, Kerr J, Shi Y, Wagner G, Kumar TKS, Koenig S, Choset H (2019) Primal: pathfinding via reinforcement and imitation multi-agent learning. IEEE Robot Autom Lett 4(3):2378–2385. https://doi.org/10.1109/LRA.2019.2903261
Semnani SH, Liu H, Everett M, de Ruiter A, How JP (2020) Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning. IEEE Robot Autom Lett 5(2):3221–3226. https://doi.org/10.1109/LRA.2020.2974695
Snape J, Berg JVD, Guy SJ, Manocha D (2011) The hybrid reciprocal velocity obstacle. IEEE Trans Robot 27(4):696–706. https://doi.org/10.1109/TRO.2011.2120810
Song C, He Z, Dong L (2022) A local-and-global attention reinforcement learning algorithm for multiagent cooperative navigation. In: IEEE transactions on neural networks and learning systems (to be published). https://doi.org/10.1109/TNNLS.2022.3220798
Tang S, Thomas J, Kumar V (2018) Hold or take optimal plan (hoop): a quadratic programming approach to multi-robot trajectory generation. Int J Robot Res 37(9):1062–1084
Vemula A, Muelling K, Oh J (2018) Social attention: modeling attention in human crowds. In: 2018 IEEE international conference on robotics and automation (ICRA), pp 4601–4607. https://doi.org/10.1109/ICRA.2018.8460504
Wang L, Li Z, Wen C, He R, Guo F (2018) Reciprocal collision avoidance for nonholonomic mobile robots. In: 2018 15th International conference on control, automation, robotics and vision (ICARCV), pp 371–376. https://doi.org/10.1109/ICARCV.2018.8581239
Wang RE, Everett M, How JP (2020) R-MADDPG for partially observable environments and limited communication. arXiv preprint arXiv:2002.06684
Wang B, Liu Z, Li Q, Prorok A (2020) Mobile robot path planning in dynamic environments through globally guided reinforcement learning. IEEE Robot Autom Lett 5(4):6932–6939. https://doi.org/10.1109/LRA.2020.3026638
Wang M, Zeng B, Wang Q (2021) Research on motion planning based on flocking control and reinforcement learning for multi-robot systems. Machines. https://doi.org/10.3390/machines9040077
Yu J, LaValle SM (2016) Optimal multirobot path planning on graphs: complete algorithms and effective heuristics. IEEE Trans Robot 32(5):1163–1177. https://doi.org/10.1109/TRO.2016.2593448
Yu C, Velu A, Vinitsky E, Wang Y, Bayen A, Wu Y (2021) The surprising effectiveness of PPO in cooperative, multi-agent games. arXiv preprint arXiv:2103.01955
Zhou Y, Li S, Garcke J (2021) R-SARL: crowd-aware navigation based deep reinforcement learning for nonholonomic robot in complex environments. arXiv preprint arXiv:2105.13409

Authors and Affiliations

Lu Dong[1,5] · Zichen He[2,6] · Chunwei Song[3] · Xin Yuan[4] · Haichao Zhang[2]

✉ Zichen He
irvinghe1518@gmail.com

Lu Dong
ldong90@seu.edu.cn

Chunwei Song
scw03180522@163.com

Xin Yuan
xinyuan@seu.edu.cn

Haichao Zhang
dhu_zhc@mail.dhu.edu.cn

[1] School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China

[2] Shanghai Aerospace Electronic Technology Institute, Shanghai 201109, China

[3] Huawei Shanghai Research Institute, Shanghai 201206, China

[4] School of Automation, Southeast University, Nanjing 210096, China

[5] Engineering Research Center of Blockchain Application, Supervision And Management, Ministry of Education, Southeast University, Nanjing 211189, China

[6] College of Electronics and Information Engineering, Tongji University, Shanghai, China