

# PC<sup>2</sup>P: Multi-Agent Path Finding via Personalized-Enhanced Communication and Crowd Perception

Guotao Li<sup>12</sup>, Shaoyun Xu<sup>1</sup>, Yuexing Hao<sup>12†</sup>, Yang Wang<sup>1</sup>, Yuhui Sun<sup>12</sup>

**Abstract**—Distributed Multi-Agent Path Finding (MAPF) integrated with Multi-Agent Reinforcement Learning (MARL) has emerged as a prominent research focus, enabling real-time cooperative decision-making in partially observable environments through inter-agent communication. However, due to insufficient collaborative and perceptual capabilities, existing methods are inadequate for scaling across diverse environmental conditions. To address these challenges, we propose PC<sup>2</sup>P, a novel distributed MAPF method derived from a Q-learning-based MARL framework. Initially, we introduce a personalized-enhanced communication mechanism based on dynamic graph topology, which ascertains the core aspects of “who” and “what” in interactive process through three-stage operations: selection, generation, and aggregation. Concurrently, we incorporate local crowd perception to enrich agents’ heuristic observation, thereby strengthening the model’s guidance for effective actions via the integration of static spatial constraints and dynamic occupancy changes. To resolve extreme deadlock issues, we propose a region-based deadlock-breaking strategy that leverages expert guidance to implement efficient coordination within confined areas. Experimental results demonstrate that PC<sup>2</sup>P achieves superior performance compared to state-of-the-art distributed MAPF methods in varied environments. Ablation studies further confirm the effectiveness of each module for overall performance.

## I. INTRODUCTION

Multi-Agent Path Finding (MAPF) aims to plan collision-free paths for multiple agents from start positions to goal locations while optimizing total path length or travel time [1], [2]. Given its extensive application in various domains including intelligent warehouses [3], autonomous vehicles [4], and industrial robotics [5], how to enhance planning capability has become a current research hotspot. Current MAPF methods can generally be categorized into centralized and distributed paradigms. Centralized algorithms are mostly expanded based on heuristic search, such as ODrM\* [6] and CBS [7]. These methods heavily rely on pre-acquired global information and necessitate replanning in dynamic environments, limiting their scalability and real-time performance.

Distributed algorithms tackle the MAPF problem using Multi-Agent Reinforcement Learning (MARL), which rely solely on observations within a limited field of view (FOV) and achieve real-time collaborative policies through agent communication. PRIMAL [8] is the first decentralized MAPF framework combining reinforcement learning (RL) and imitation learning (IL). DHC [9] introduced a multi-agent

communication network mechanism to enhance coordination among different agents without expert guidance. Subsequent studies, such as [10], [11], and [12], have further improved collaborative efficiency by optimizing communication modules. Meanwhile, several methods have been specifically developed for specialized planning scenarios, including small-scale dense environments [13], [14] and large-scale environments [15], [16]. However, current distributed MAPF algorithms exhibit instability in planning success rate and solution quality across environments with varying scales, densities, and types, especially experiencing a sharp performance decline in large-scale and high-density scenarios. The primary reason is that increased environmental density leads to higher concentration of agents in a confined area. In such scenario, collaborative ability plays a decisive role in collision-free path planning. But existing communication mechanisms are deficient in managing interaction relationships and contents among different agents, which undermines their robustness across various environments. Additionally, typical agent observations tend to be repetitive or similar across multiple time steps within dense areas, thus failing to provide effective perception information for collaboration strategies. If mutual avoidance fails, agents may frequently remain stationary or wander within narrow spaces, potentially causing complex deadlocks over time and ultimately leading to mission failure.

In this paper, we present PC<sup>2</sup>P, a novel distributed MARL-MAPF method based on coordination and perception advancement. Firstly, we devise a three-stage personalized-enhanced communication mechanism, “Selection-Generation-Aggregation”, which dynamically chooses communicating agents and acquires customized messages from them incorporating with a graph attention mechanism. Secondly, we introduce crowd-augmented local observation that integrates static spatial constraints and dynamic occupancy changes. This enhances agents’ awareness of both current and future environmental states, providing heuristic guidance for real-time policy outputs. Finally, we propose a region-based cooperative deadlock-breaking strategy that leverages expert guidance to resolve localized extreme deadlocks in limited areas. Empirical results demonstrate that PC<sup>2</sup>P consistently outperforms other distributed algorithms across various test environments.

## II. RELATED WORK

### A. Multi-Agent Reinforcement Learning for MAPF

Recent research has increasingly explored MAPF tasks in combination with MARL, developing innovative approaches

<sup>1</sup>Authors are with the Institute of Microelectronics of the Chinese Academy of Sciences, No. 3 North Tucheng West Road, Chaoyang District, Beijing, China {liguotao24, xushaoyun, haoyuexing, wangyang, sunyuhui}@ime.ac.cn

<sup>2</sup>University of Chinese Academy of Sciences

†Corresponding author

to enhance inter-agent coordination. PRIMAL [8] pioneered the integration of asynchronous advantage actor-critic (A3C) [17] with Imitation Learning (IL), leveraging expert guidance to optimize policy networks. Building on this foundation, PRIMAL2 [18] extended the approach to lifelong MAPF tasks. Afterward, extensive research has incorporated communication mechanisms into MARL frameworks to boost collaborative ability. DHC [9] introduces heuristic channel guidance to direct agent movement, stabilizes the training process through curriculum learning [19] and distributed prioritized replay buffer [20], and uses graph convolution [21] for communication. DCC [10] proposes Decision Causal Communication, which utilizes both decision causal unit and a request-response mechanism to enhance communication efficiency. PICO [13] infers implicit priority relationships among agents from expert demonstrations and constructs a dynamic communication topology. MAGAT [15] builds a distributed communication network through supervised learning. SACHA [22] leverages a heuristic-based attention mechanism to foster cooperation among agents within the FOV, employing graph convolution for effective information exchange. SCRIMP [11] implements a scalable communication module using a Transformer-like architecture [23]. HELSA [16] utilizes hierarchical reinforcement learning to decompose complex tasks, enabling efficient planning in large-scale environments. CRAMP [14] enhances agent cooperation in dense settings through environment-specific reward shaping. EPH [12] proposes three inference techniques, achieving effective integration with traditional algorithms. Nevertheless, these approaches struggle with scalability across diverse environments, particularly in large-scale, high-density scenarios.

### B. Communication Mechanisms in MARL

The essence of addressing MAPF with MARL lies in mastering collaboration, where efficient communication is crucial for inter-agent coordination. Effective information sharing under partial observability has become a key research focus in MARL [24]. Traditional broadcast communication, however, is hindered by high computational overhead and inefficiency. To address these limitations, researchers have proposed numerous innovative communication strategies. TarMAC [25] employs attention mechanisms to facilitate selective message transmission and enhances efficiency through multi-round communication. I2C [26] introduces a decision causal unit to enable selective communication and incorporates a request-response mechanism to reduce communication overhead. MAGIC [27] utilizes Graph Attention Networks (GAT) [28] as both a scheduler and aggregator, optimizing communication target selection and information aggregation. AC2C [29] implements a two-hop communication mechanism, allowing agents to access remote information beyond FOV. CACOM [30] proposes a two-stage communication framework that dynamically adapts the communication topology based on context-aware messages and leverages Transformer [23] for efficient information aggregation.

## III. PROBLEM FORMULATION

### A. MAPF Definition

The MAPF problem focuses on computing collision-free paths for  $k$  agents on a graph  $G = (V, E)$ , where  $V$  is the vertex set and  $E$  is the edge set. Each agent moves from a unique start position  $s_i \in S = \{s_1, s_2, \dots, s_k\}$  to a designated goal position  $g_i \in T = \{g_1, g_2, \dots, g_k\}$ . Time is discretized into time steps, during which each agent can either move to an adjacent vertex or remain at its current vertex. The total path of agent  $i$ , denoted as  $\pi_i = (v_1^i, v_2^i, \dots, v_T^i)$ , must satisfy three conditions: (1)  $v_1^i = s_i$ ; (2)  $v_T^i = g_i$ ; and (3) for each time step  $t$ ,  $(v_t^i, v_{t+1}^i)$  is either a valid edge in  $E$  or a stationary action where  $v_t^i = v_{t+1}^i$ . To ensure conflict-free planning, two types of constraints are imposed: vertex conflicts ( $v_t^i \neq v_t^j$ ) to prevent agents from occupying the same vertex simultaneously, and edge conflicts ( $(v_t^i, v_{t+1}^i) \neq (v_{t+1}^j, v_t^j)$ ) to avoid agents traversing the same edge in opposite directions. The objective is to minimize the total number of time steps  $\sum_{i=1}^k T_i$ , where  $T_i$  denotes the first time step at which agent  $i$  reaches its goal  $g_i$ .

### B. Environment Settings

According to the MAPF task definition, we model the environment as a 2D 4-neighbor grid world, where each cell is either empty or an obstacle.  $m$  agents initially occupy  $m$  empty cells, randomly selected  $m$  initial and  $m$  goal positions from empty cells when each episode starts, to ensure connectivity and non-overlapping between start and goal locations. At each time step, agents can move up, down, left, right, or stay still. During movement, conflicts are resolved by executing the action with the second-highest Q value, as proposed in EPH [12]. This paper considers a partially observable environment, where each agent can only observe environmental information within an  $l \times l$  FOV. The reward function adopts the form proposed by DHC [9], as shown in Table I, designed to encourage agents to reach their goals quickly while avoiding conflicts.

TABLE I: Reward Structure

Actions	Reward
Move (Up/Down/Left/Right)	-0.075
Stay (on goal, away goal)	0, -0.075
Collision (obstacle/agents)	-0.5
Finish	3

## IV. METHOD

PC<sup>2</sup>P addresses partially observable distributed MAPF through Q-learning-based MARL by extracting local observation information for individual agents, facilitating inter-agent communication via deep integration of personalized information, and ultimately generating real-time Q values for policy selection. Figure 1 illustrates the overall framework of PC<sup>2</sup>P. Section IV-A details the personalized-enhanced communication and Q-learning training, Section IV-B investigates crowd-augmented local observations, and Section IV-C discusses cooperative deadlock-breaking strategies for resolving extreme deadlock scenarios.

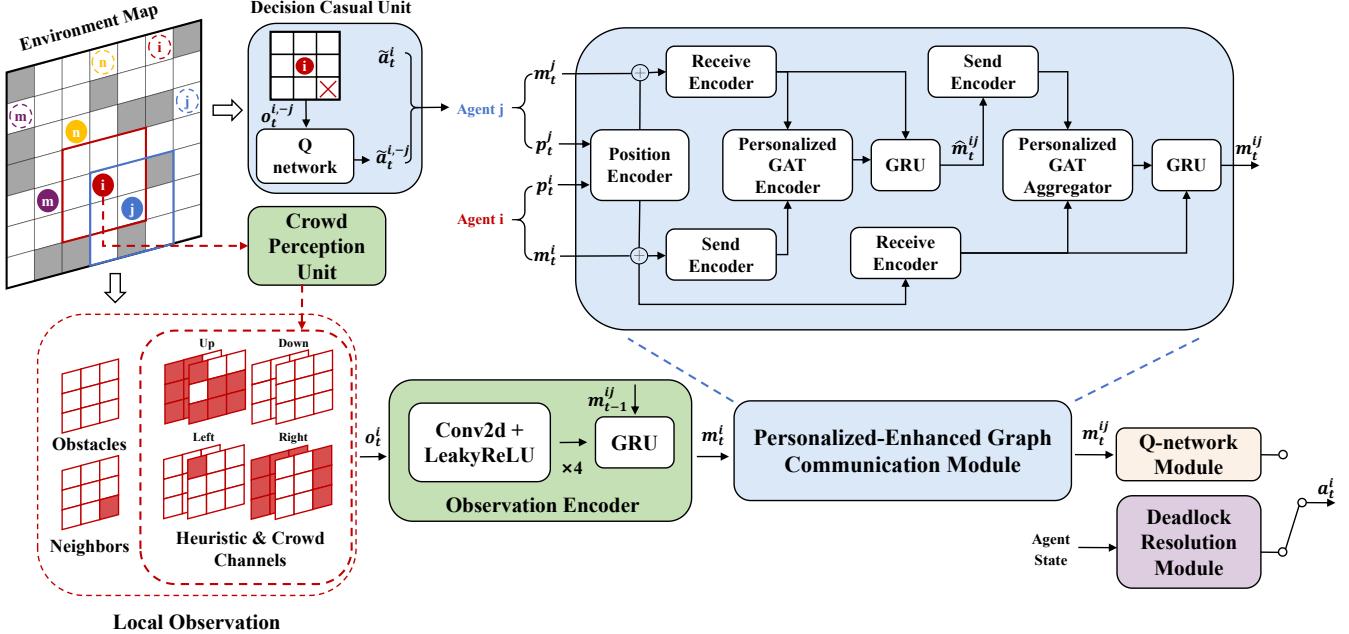


Fig. 1: The PC<sup>2</sup>P framework comprises four components: Observation Module (green), Communication Module (blue), Q-network Module (pink), and Deadlock Resolution Module (purple). Agents gather local observations (red dashed box), include basic and crowd-augmented information, which are then encoded by the Observation Encoder. The Communication Module selects relevant agents and aggregates personalized information (detailed above), enabling the Q-network Module to generate policies. Finally, upon detecting extreme deadlocks, PC<sup>2</sup>P activates the Deadlock Resolution Module to output alternative strategies.

### A. Personalized-Enhanced Communication for MAPF

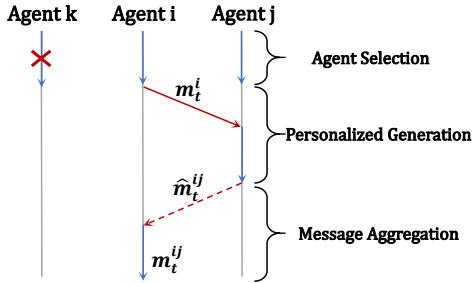


Fig. 2: Illustration of the Personalized-Enhanced Communication Mechanism. (1) Agent  $i$  selects Agent  $j$  for communication via the Decision Causal Unit. (2) Agent  $i$  sends information  $m_t^i$  to agent  $j$  to generate the personalized message  $\hat{m}_t^{ij}$ . (3) Agent  $i$  aggregates the received  $\hat{m}_t^{ij}$  with its own information  $m_t^i$  to complete the communication process.

Due to variations in agents' states (e.g., positions, goals, and local environments), agents' communication requirements differ significantly. Existing methods typically utilize either broadcast or selective communication mechanisms, determining "who" to interact with. However, these approaches neglect to address the issue of "what" — the content of communication. We propose a personalized communication mechanism based on graph topology enhancement. Build-

ing on causal decision-based selective communication, each agent sends or receives personalized messages based on its own information, thereby enabling customized communication. The mechanism comprises three stages: agent selection, personalized message generation, and message aggregation, as illustrated in Figure 2. In the initial stage, agent  $i$  identifies its communication partners through Decision Causal Unit, following the methodology proposed by DCC [10]. The communication partner set, denoted as  $\mathcal{C}_i$ , is defined as:

$$\mathcal{C}_i = \{j \mid \tilde{a}_t^i \neq \tilde{a}_t^{i-j}\}_{j \in \mathcal{B}_i}, \quad (1)$$

where  $\mathcal{B}_i$  represents the agents within the local observation neighborhood,  $\tilde{a}_t^i$  denotes agent  $i$ 's action at time  $t$  based on its observation, and  $\tilde{a}_t^{i-j}$  denotes the action when the influence of agent  $j$  is excluded. If the action variation occurs, agent  $j$  is selected for communication. In the second stage, agent  $i$  sends information  $m_t^i$  to agent  $j$  to generate the message  $\hat{m}_t^{ij}$  that is personalized for agent  $i$ . In the third stage, the personalized message  $\hat{m}_t^{ij}$  is returned and aggregated with agent  $i$ 's original information  $m_t^i$  to obtain the post-communication message  $m_t^{ij}$ . Ultimately,  $m_t^{ij}$  is used to compute the Q value through the Q-network, with the final policy  $a_t^i$  determined by argmax operation.

1) *Personalized-Enhanced Graph Communication Module:* Given the dynamic nature of inter-agent communication relationships, we adopt a graph-based topology to establish the real-time communication network among agents. Notably, we utilize Graph Attention Networks (GAT) [28]

to accomplish message generation and aggregation, which adaptively assigns distinct weights to time-varying communication partners, as illustrated in Figure 1. Agent information consists of local observation  $m_t$  and positional information  $p_t$ . Using agent  $i$  and agent  $j$  as examples,  $p_t^i$  and  $p_t^j$  are transformed into position vectors  $l_t^i$  and  $l_t^j$  via position encoder, and concatenated with the respective local observations  $m_t^i$  and  $m_t^j$ . The concatenated representations are preprocessed through the send encoder  $f_s$  and the receive encoder  $f_r$  respectively. Subsequently, Personalized GAT Encoder computes the personalized message  $\hat{m}_t^{ij}$  based on observations  $m_t^i$  and  $m_t^j$ . Personalized GAT Aggregator integrates the agent  $i$ 's information  $m_t^i$  with the personalized message  $\hat{m}_t^{ij}$ , ultimately generating the post-communication message  $m_t^{ij}$ . The above two network components share the same structure based on GATv2 [31]. For clarity, we illustrate them using Personalized GAT Encoder as the example. Initially, attention coefficient  $e_{ij}$  is computed between agents  $i$  and  $j$ :

$$e_{ij} = \mathbf{a}^\top \text{LeakyReLU} ([f_s(m_t^i) \| f_r(m_t^j)]), \quad (2)$$

where  $\mathbf{a}$  represents the linear transformation weight vector,  $f_s(m_t^i)$  denotes the sending feature vector,  $f_r(m_t^j)$  represents the receiving feature vector, and LeakyReLU introduces non-linear characteristics. The attention coefficient is then normalized to  $\alpha_{ij}$ :

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{j \in \mathcal{C}_i} \exp(e_{ij})}. \quad (3)$$

For each attention head, we compute the personalized information  $\hat{m}_t^{ij}$  using normalized coefficients  $\alpha_{ij}$  via linear mapping  $f_p$ :

$$\hat{m}_t^{ij} = f_p \left( \text{concat} \left( \sum_{j \in \mathcal{C}_i} \alpha_{ij}^h f_r(m_t^j), \forall h \in \mathcal{H} \right) \right), \quad (4)$$

where  $\mathcal{H}$  represents the attention head set, and  $\alpha_{ij}^h$  denotes the normalized attention coefficient for the  $h$ -th head. Finally, Personalized GAT Encoder returns the personalized communication information  $\hat{m}_t^{ij}$  to agent  $i$  via a Gated Recurrent Unit (GRU).

2) *Q-network*: We utilize the Double Dueling Deep Q Network (D3QN) for model training, which combines the advantages of Dueling DQN [32] and Double DQN [33]. This approach effectively stabilizes the training process and mitigates Q value overestimation. The action value function for agent  $i$  at time  $t$  is defined as:

$$Q_{s,a}^i = V_s(m_t^{ij}) + \left[ A(m_t^{ij})_a - \frac{1}{|\mathcal{A}|} \sum_{a'} A(m_t^{ij})_{a'} \right], \quad (5)$$

Where  $\mathcal{A}$  represents the action space, and  $m_t^{ij}$  represents the post-communication message of agent  $i$ . After obtaining the Q value for agent  $i$ , we employ the Temporal Difference (TD) error as the loss function for model training. Calculate the multi-step TD error using Mean Squared Error (MSE):

$$\mathcal{L}(\theta) = \text{MSE} (R_t^i - Q_{s_t, a_t}^i(\theta)), \quad (6)$$

where  $R_t^i = r_t^i + \gamma r_{t+1}^i + \dots + \gamma^n Q_{s_{t+n}, a_{t+n}}^i(\bar{\theta})$  represents the multi-step expected return,  $r_t^i$  represents the reward obtained by agent  $i$  at time  $t$ .  $\bar{\theta}$  represents the target network parameters, while  $\theta$  corresponds to the current network parameters.

## B. Crowd-Augmented Local Observation

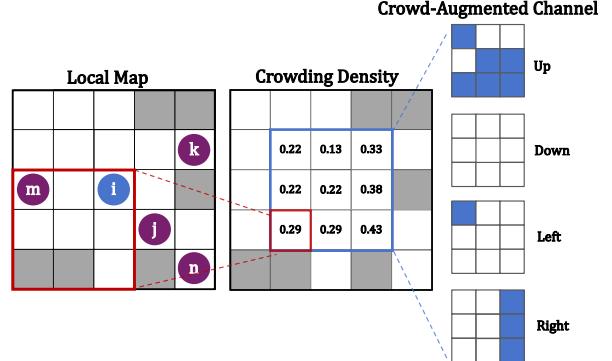


Fig. 3: Illustration of how agent  $i$  acquires the crowd-augmented channel. To enhance clarity, both FOV (blue bounding box) and the crowd perception region (red bounding box) in the figure are set to a  $3 \times 3$  area as an example. Crowding density is calculated based on the number of agents and obstacles within the  $3 \times 3$  region surrounding  $i$ . Each directional channel (Up, Down, Left, Right) is marked as 1 if the corresponding action reduces the comprehensive congestion degree for  $i$ ; otherwise, it is 0.

Agent's local observations in our method consist of two components. One part is basic information including obstacle positions, agent locations, and distance-based heuristic information. The other part is crowd perception information, which dynamically assesses current and future congestion situations within the local region. As illustrated in Figure 3, each agent computes four crowd-augmented channels by evaluating congestion degree within its local region. Specifically, the congestion degree at each position is determined through a comprehensive assessment of the surrounding  $5 \times 5$  region, composed of static congestion  $\rho_{\text{static}}$  and dynamic congestion  $\rho_{\text{dynamic}}$ . Static congestion  $\rho_{\text{static}}(x, y)$  quantifies spatial constraints arising from the distribution of obstacles, whose metric is computed as:

$$\rho_{\text{static}}(x, y) = \frac{O(x, y)}{N(x, y)}, \quad (7)$$

where  $O(x, y)$  denotes the number of obstacles in the local region at position  $(x, y)$ , and  $N(x, y)$  represents the total number of grid cells in the local region at position  $(x, y)$ . Dynamic congestion  $\rho_{\text{dynamic}}(x, y, t)$  measures the spatial occupancy caused by agent movement, whose metric is calculated as:

$$\rho_{\text{dynamic}}(x, y, t) = \frac{A(x, y, t)}{N(x, y) - A(x, y, t) - O(x, y)}, \quad (8)$$

where  $A(x, y, t)$  denotes the number of agents in the local region around position  $(x, y)$  at time  $t$ . Comprehensive

congestion degree  $\rho(x, y, t)$  is obtained by the weighted summation of static and dynamic congestion, as follows:

$$\rho(x, y, t) = \alpha \cdot \rho_{\text{static}}(x, y) + (1 - \alpha) \cdot \rho_{\text{dynamic}}(x, y, t), \quad (9)$$

where  $\alpha$  is used to balance the relative impacts of static and dynamic congestion. By integrating local crowd perception information, agents can not only monitor traversability of the current area, but also predict spatial states of future potential activity regions, thereby enhancing overall environmental awareness ability.

At time  $t$ , agent  $i$  receives a binary observation matrix  $o_t^i$  with dimensions  $H \times W \times 10$ , where  $H \times W$  denotes the dimensions of the FOV. Six channels encode obstacle positions, agent positions, and distance-based heuristic information. Four channels representing crowd-augmented features. Observation Encoder includes four convolutional layers and a GRU module. The observation  $o_t^i$  is first pre-processed through the convolutional layers, then combined with the communication information from the previous time step  $m_{t-1}^{ij}$ , and finally passed through the GRU module to generate the current feature  $m_t^i$ .

### C. Cooperative Deadlock-Breaking

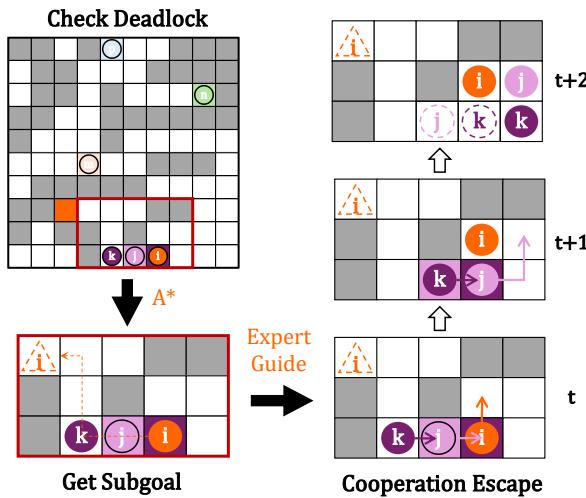


Fig. 4: Illustration of a typical deadlock-breaking example. First, agents  $j, i, k$  are identified as a deadlocked group. Next, a Local Map is constructed, and a subgoal is generated using  $A^*$  for agent  $i$ . With expert guidance, the agents collaborate to resolve the deadlock. Finally, at time step  $t+2$ , the task is continued by the RL planner.

Deadlock is an unavoidable challenge in MAPF, especially in dense environments where confined spaces can trap agents, leading to task failure. Existing methods employ random actions or single-agent  $A^*$  algorithms for resolution, yet they focus on individual-level solutions and neglect the root cause: insufficient collaborative ability hinders effective coordination in complex local environments. Therefore, we propose a region-based, expert-guided cooperative deadlock-breaking strategy. Figure 4 illustrates a typical example of

the deadlock-breaking process. We firstly identify the set of deadlocked agents  $A_{\text{deadlocked}}$  and deadlocked agent groups  $\mathcal{D}$  separately to clarify the execution scenario. The deadlocked agent set  $A_{\text{deadlocked}}$  consists of active agents (have not yet reached goal positions) which remain stationary or wander for a period. Formally, they are defined as:

$$A_{\text{deadlocked}} = \{i \in A_{\text{active}} \mid v_i^{t-1} = v_i^{t-3} \wedge v_i^{t-2} = v_i^{t-4}\}, \quad (10)$$

where  $A_{\text{active}}$  represents the set of active agents, and  $v_i^{t-k}$  denotes the position of agent  $i$  at time  $t - k$ . The set of deadlocked agent groups  $\mathcal{D}$  is defined as follows:

$$\mathcal{D} = \{\{i, \mathcal{B}_i\} \mid i \in A_{\text{deadlocked}}, \\ \forall j \in \mathcal{B}_i, j \notin A_{\text{active}} \vee j \in A_{\text{deadlocked}}\}, \quad (11)$$

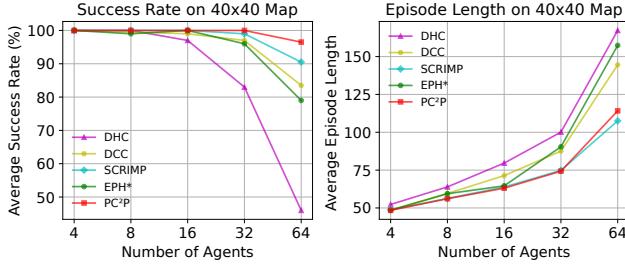
where  $\mathcal{B}_i$  denotes the set of neighboring agents of agent  $i$ . In each deadlocked agent group, all agents are either deadlocked or inactive, as their inability to move leads to the emergence of complex deadlocks. For each deadlocked agent group, a minimized Local Map is constructed. If an agent's goal lies beyond this map,  $A^*$  is applied to set a subgoal, aligning local movement with the global path. At this point, a localized planning task is defined, and ODrM\* [6] is utilized to resolve deadlocks within the Local Map. This strategy effectively assists the distributed algorithm in solving complex deadlock problems with limited FOV. Figure 4 illustrates the specific process of agents  $j, i, k$  breaking a deadlock under expert guidance. At time step  $t + 2$ , the deadlock is successfully resolved, enabling agents to continue their subsequent tasks according to the MARL method.

## V. EXPERIMENTS

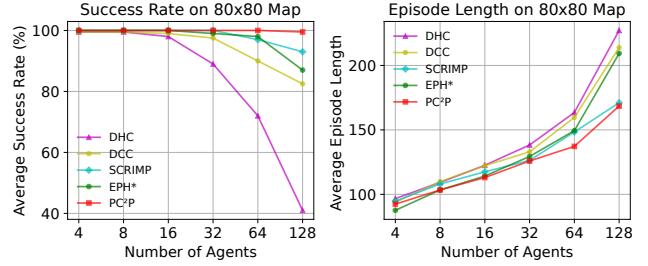
### A. Experimental Settings

1) *Training*: PC<sup>2</sup>P is trained using the Q-learning-based MARL approach, enhanced by distributed prioritized replay buffer to optimize training efficiency. The Q-network is updated using a two-step Temporal Difference (TD) error with a discount factor of 0.99. Task difficulty is progressively increased through a curriculum learning strategy. Initially, the environment is configured as a  $10 \times 10$  grid with a single agent and obstacles following a triangular distribution ranging from 0 to 0.5, with a peak at 0.33. Once the success rate surpasses 0.9, the size of the map gradually increases to  $40 \times 40$ , and the number of agents increases to 16. Each agent has a FOV size of  $9 \times 9$ , with a crowd perception area of  $5 \times 5$ . The maximum number of steps per episode is set to 256. For efficient exploration, 32 actors are run in parallel, with each actor updating its network every 1,000 training steps. The experiments were performed on a single NVIDIA 4090 GPU and an AMD EPYC 9654 (96-core) server. The total training step count was 30k steps, with a training duration of approximately 24 hours.

2) *Testing*: This experiment employs the same test maps as [9], [10], [12] to evaluate the performance of PC<sup>2</sup>P, including random and structured maps. For random maps, we test on  $40 \times 40$  and  $80 \times 80$  maps with 0.3 obstacle density

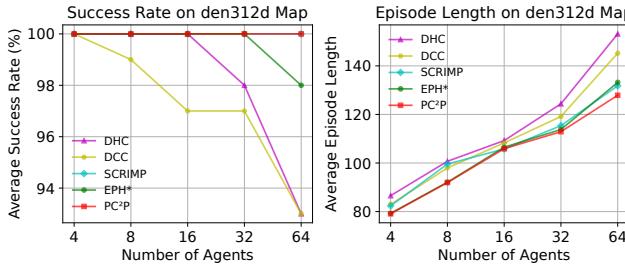


(a) Results on 40x40 Map

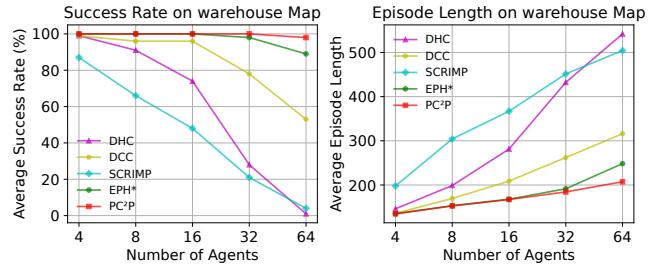


(b) Results on 80x80 Map

Fig. 5: Experimental Results of SR and EL on random maps.



(a) Results on den312d Map



(b) Results on warehouse Map

Fig. 6: Experimental Results of SR and EL on structured maps.

and maximum step lengths of 256 and 386 respectively. The structured maps, den312d ( $65 \times 81$ ) and warehouse ( $161 \times 63$ ), are sourced from the Moving AI Lab dataset [1], with maximum step lengths of 256 and 512, respectively. To test the performance in larger-scale and high-density environments,  $120 \times 120$  and  $240 \times 240$  random maps are applied, maintaining 0.3 obstacle density and maximum step lengths of 512 and 1,024. Experiment maps consist of 100 instances for each random map and 300 instances for each structured map. Evaluation metrics include the success rate (SR), which represents the proportion of completed tasks, and the episode length (EL), which reflects the quality of the solutions. Four state-of-the-art distributed MAPF models, including DHC [9], DCC [10], SCRIMP [11], and EPH [12], are selected as baseline algorithms for comparative analysis. Their publicly available source codes and pre-trained model weights are used for evaluation. EPH ensembles 22 different planners<sup>1</sup> and employs parallel computation to select the optimal result for each task. Due to its high computational cost and time consumption, it is not suitable for real-time MAPF tasks. Therefore, we adopt the default planner provided in the open-source code for testing, referring to it as EPH\*.

### B. Comparison Results

1) *Random Maps*: As shown in Figure 5, PC<sup>2</sup>P achieves the highest success rate on both  $40 \times 40$  and  $80 \times 80$  maps. Especially when the number of agents increases to 128 (80-size), PC<sup>2</sup>P maintains the success rate consistently close to 100%, while other models exhibit a significant decline.

Additionally, PC<sup>2</sup>P demonstrates nearly the lowest average step on two maps. The exception occurs when the number of agents is 64 (40-sized), where SCRIMP performs better. This can be attributed to its effective collision resolution strategy, which reduces the path lengths of colliding agents and further lowers the overall average step, yet the success rate of SCRIMP is approximately 5% lower than that of PC<sup>2</sup>P in this situation.

2) *Structured Maps*: Figure 6 demonstrates the performance of PC<sup>2</sup>P in structured maps, achieving optimal results on both test maps. Particularly in the warehouse map, which closely simulates real-world conditions, PC<sup>2</sup>P achieves a 98% success rate even with 64 agents, whereas the performance of most other models declines significantly. This highlights its effectiveness in addressing planning tasks across diverse environments.

3) *Large-Scale Maps*: Table II presents the experimental results of PC<sup>2</sup>P and other models in  $120 \times 120$  and  $240 \times 240$  random maps. PC<sup>2</sup>P achieves the highest success rate and the lowest average step across all large-scale maps. As the number of agents increases, the demand for collaboration and the occurrence of deadlocks increase significantly, leading to a sharp decline in other models' performance. In contrast, the success rates of PC<sup>2</sup>P remain consistently above 90%. This showcases the scalability and coordination ability of PC<sup>2</sup>P in large-scale, high-density environments.

### C. Ablation Studies

Table III evaluates the contribution of each module in PC<sup>2</sup>P to model performance, with PGC, CA, and CDB denoting Personalized Graph-Enhanced Communication Mod-

<sup>1</sup> Available at <https://github.com/ai4co/eph-mapf>

TABLE II: Experimental Results of SR and EL on large-scale maps.

Methods	120-sized map, 0.3 density						240-sized map, 0.3 density					
	n = 128		n = 256		n = 384		n = 256		n = 512		n = 768	
	SR↑	EL↓	SR↑	EL↓	SR↑	EL↓	SR↑	EL↓	SR↑	EL↓	SR↑	EL↓
DHC	53	368.1	13	487.9	1	511.8	44	778.9	8	983.2	1	1019.9
DCC	87	260.4	79	329.6	52	436.3	88	495.9	61	696.4	57	749.7
SCRIMP	35	416.1	0	—	0	—	0	—	0	—	0	—
EPH*	93	240.3	82	320.3	29	470.4	94	451.2	66	659.0	29	879.7
PC <sup>2</sup> P	<b>100</b>	<b>210.0</b>	<b>99</b>	<b>252.1</b>	<b>91</b>	<b>338.6</b>	<b>98</b>	<b>423.3</b>	<b>96</b>	<b>477.1</b>	<b>97</b>	<b>510.0</b>

TABLE III: Ablation Study Results of Different Modules on SR and EL.

Methods	80-sized map, 0.3 density						120-sized map, 0.3 density					
	n = 128		n = 192		n = 256		n = 256		n = 384		n = 512	
	SR↑	EL↓	SR↑	EL↓	SR↑	EL↓	SR↑	EL↓	SR↑	EL↓	SR↑	EL↓
w/o PGC + DHC	80	219.8	50	310.7	11	373.9	64	386.6	20	480.8	2	508.5
w/o PGC + DCC	95.5	184.5	81	245.0	65	305.3	95	268.1	66	389.3	21	485.1
w/o PGC + SCRIMP	96	179.9	85	241.8	51	326.2	94	273.6	58	407.5	20	489.1
w/o CA	94.5	176.3	82	232.6	50	319.9	89	282.2	67	383.7	18	489.9
w/o CDB	96.5	176.9	90	231.9	67	301.1	95	274.1	82	354.3	42	460.5
PC <sup>2</sup> P	<b>99.5</b>	<b>168.5</b>	<b>94</b>	<b>215.1</b>	<b>77</b>	<b>281.6</b>	<b>99</b>	<b>252.1</b>	<b>91</b>	<b>338.6</b>	<b>56</b>	<b>440.9</b>

ule, Crowd-Augmented Local Observation, and Cooperative Deadlock-Breaking Module, respectively. The experiments are conducted on random maps of size  $80 \times 80$  and  $120 \times 120$ , with 0.3 obstacle density. Since the success rate consistently reached 100% with a smaller number of agents, the agent count increased incrementally from 128 to 512 to accurately assess the impact of each module.

1) *Personalized Graph-Enhanced Communication Module:* To evaluate the effectiveness of the PGC module, we replaced it with the communication modules from DHC [9], DCC [10], and SCRIMP [11]. Since EPH [12] employs the same communication module as SCRIMP, it was not included separately in the experiments. All of the above models are retrained and tested. As shown in Table III, the PGC module outperforms all alternatives, and its advantages become more pronounced as the number of agents increases. Notably, in the 512-agent (120-sized) setting, PC<sup>2</sup>P achieves a 35% higher success rate compared to the best-performing alternative communication module. We can observe that as the scene density increases, higher demands are placed on collaborative ability. The PGC module dynamically adjusts inter-agent relationships and generates personalized information tailored to individual needs, thereby enabling more efficient collaborative communication and information fusion.

2) *Crowd-Augmented Local Observation:* The experimental results after removing the CA module are presented in Table III. As the agent density increases, the effect of CA module becomes increasingly significant. Specifically, in 256-agent (80-sized) and 512-agent (120-sized) environments, the success rate decreases by 27% and 38%, and the average step increases by 38.3 and 49. These dramatic changes indicate that as environmental density increases, local congestion information becomes increasingly critical for effective path planning. Heuristic-based channels guide the agents' movement towards destination, while crowd-augmented channels potentially lead them to avoid congested areas. Their combination provides valuable directional infor-

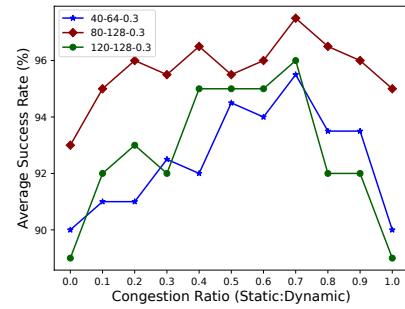


Fig. 7: Impact of congestion ratio on success rate.

mation for real-time policy optimization.

To further investigate the effect on the proportion of static congestion  $\rho_{static}$  and dynamic congestion  $\rho_{dynamic}$ , we evaluated various ratios, as illustrated in Figure 7. When the ratio of static to dynamic congestion coefficients is set to 0.7 : 0.3, the success rates across all three random maps reach optimal levels, significantly outperforming the success rates achieved using either parameter alone.

3) *Cooperative Deadlock-Breaking Module:* The experimental results after removing the CDB module are presented in Table III. In high-density environments, the impact of the CDB module becomes appreciably more substantial. The success rate increases by 10% and 14% in 256-agent (80-sized) and 512-agent (120-sized) environments, respectively. The improvement can be attributed to the increased frequency of complex deadlocks in denser environments. The CDB module addresses this by enabling effective avoidance through local agent coordination.

Notably, the performance of the model w/o CDB ranks second only to PC<sup>2</sup>P in all ablation experiments. This is because the PGC and CA modules have already enhanced deadlock resolution capabilities to some extent. Only when encountering extremely complex deadlock problems is it necessary to leverage traditional algorithms for small-scale

local coordination. These scenarios often represent critical bottlenecks that hinder overall task completion. Thus, the CDB module is essential for improving both success rate and generalizability of the model in diverse environments.

## VI. CONCLUSIONS

In this paper, we propose a novel distributed real-time planner, PC<sup>2</sup>P, for solving MAPF problems. This method employs a personalized graph-enhanced communication module, leveraging graph attention mechanisms to dynamically capture inter-agent relationships and adaptively assign personalized messages. Additionally, we introduce crowd perception information that integrates static and dynamic congestion states, enhancing agents' situational awareness in complex scenarios. A cooperative deadlock-breaking strategy is designed to address severe deadlock situations using expert guidance. Experimental results demonstrate that PC<sup>2</sup>P exhibits significant advantages across test environments with varying scales, densities, and types, particularly showcasing superior collaboration and scalability in high-density settings. Future work will concentrate on addressing the instability inherent in the Independent Q-Learning (IQL) framework by exploring the integration of the Centralized Training with Decentralized Execution (CTDE) framework to mitigate conflicts and deadlocks in MAPF tasks.

## REFERENCES

- [1] R. Stern, N. Sturtevant, A. Felner, S. Koenig, H. Ma, T. Walker, J. Li, D. Atzman, L. Cohen, T. Kumar *et al.*, “Multi-agent pathfinding: Definitions, variants, and benchmarks,” in *Proceedings of the International Symposium on Combinatorial Search*, vol. 10, no. 1, 2019, pp. 151–158.
- [2] J. Chung, J. Fayyad, Y. A. Younes, and H. Najjaran, “Learning team-based navigation: a review of deep reinforcement learning techniques for multi-agent pathfinding,” *Artificial Intelligence Review*, vol. 57, no. 2, p. 41, 2024.
- [3] P. R. Wurman, R. D’Andrea, and M. Mountz, “Coordinating hundreds of cooperative, autonomous vehicles in warehouses,” *AI magazine*, vol. 29, no. 1, pp. 9–9, 2008.
- [4] J. Li, E. Lin, H. L. Vu, S. Koenig *et al.*, “Intersection coordination with priority-based search for autonomous vehicles,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 10, 2023, pp. 11 578–11 585.
- [5] F. Ho, A. Goncalves, A. Salta, M. Cavazza, R. Geraldes, and H. Prendinger, “Multi-agent path finding for uav traffic management: Robotics track,” 2019.
- [6] C. Ferner, G. Wagner, and H. Choset, “Odrm\*: optimal multirobot path planning in low dimensional search spaces,” in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 3854–3859.
- [7] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, “Conflict-based search for optimal multi-agent pathfinding,” *Artificial intelligence*, vol. 219, pp. 40–66, 2015.
- [8] G. Sartoretti, J. Kerr, Y. Shi, G. Wagner, T. S. Kumar, S. Koenig, and H. Choset, “Primal: Pathfinding via reinforcement and imitation multi-agent learning,” *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2378–2385, 2019.
- [9] Z. Ma, Y. Luo, and H. Ma, “Distributed heuristic multi-agent path finding with communication,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8699–8705.
- [10] Z. Ma, Y. Luo, and J. Pan, “Learning selective communication for multi-agent path finding,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1455–1462, 2021.
- [11] Y. Wang, B. Xiang, S. Huang, and G. Sartoretti, “Scrimp: Scalable communication for reinforcement-and imitation-learning-based multi-agent pathfinding,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 9301–9308.
- [12] H. Tang, F. Berto, and J. Park, “Ensembling prioritized hybrid policies for multi-agent pathfinding,” *arXiv preprint arXiv:2403.07559*, 2024.
- [13] W. Li, H. Chen, B. Jin, W. Tan, H. Zha, and X. Wang, “Multi-agent path finding with prioritized communication learning,” 2022. [Online]. Available: <https://arxiv.org/abs/2202.03634>
- [14] P. Pham and A. Bera, “Optimizing crowd-aware multi-agent path finding through local communication with graph neural networks,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 8913–8913.
- [15] Q. Li, W. Lin, Z. Liu, and A. Prorok, “Message-aware graph attention networks for large-scale multi-robot path planning,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5533–5540, 2021.
- [16] Z. Song, R. Zhang, and X. Cheng, “Helsa: Hierarchical reinforcement learning with spatiotemporal abstraction for large-scale multi-agent path finding,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 7318–7325.
- [17] V. Mnih, “Asynchronous methods for deep reinforcement learning,” *arXiv preprint arXiv:1602.01783*, 2016.
- [18] J. Li, A. Tinka, S. Kiesel, J. W. Durham, T. S. Kumar, and S. Koenig, “Lifelong multi-agent path finding in large-scale warehouses,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, 2021, pp. 11 272–11 281.
- [19] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.
- [20] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized experience replay,” *arXiv preprint arXiv:1511.05952*, 2015.
- [21] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: <http://arxiv.org/abs/1609.02907>
- [22] Q. Lin and H. Ma, “Sacha: Soft actor-critic with heuristic-based attention for partially observable multi-agent path finding,” *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5100–5107, 2023.
- [23] A. Vaswani, “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.
- [24] C. Zhu, M. Dastani, and S. Wang, “A survey of multi-agent deep reinforcement learning with communication,” 2024. [Online]. Available: <https://arxiv.org/abs/2203.08975>
- [25] A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, and J. Pineau, “TarMAC: Targeted multi-agent communication,” in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 1538–1546.
- [26] Z. Ding, T. Huang, and Z. Lu, “Learning individually inferred communication for multi-agent cooperation,” 2021. [Online]. Available: <https://arxiv.org/abs/2006.06455>
- [27] Y. Niu, R. R. Paleja, and M. C. Gombolay, “Multi-agent graph-attention communication and teaming,” in *AAMAS*, vol. 21, 2021, p. 20th.
- [28] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” 2018. [Online]. Available: <https://arxiv.org/abs/1710.10903>
- [29] X. Wang, X. Li, J. Shao, and J. Zhang, “Ac2c: Adaptively controlled two-hop communication for multi-agent reinforcement learning,” 2023. [Online]. Available: <https://arxiv.org/abs/2302.12515>
- [30] X. Li and J. Zhang, “Context-aware communication for multi-agent reinforcement learning,” *arXiv preprint arXiv:2312.15600*, 2023.
- [31] S. Brody, U. Alon, and E. Yahav, “How attentive are graph attention networks?” in *International Conference on Learning Representations*, 2022. [Online]. Available: <https://openreview.net/forum?id=F72ximsx7C1>
- [32] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, “Dueling network architectures for deep reinforcement learning,” in *International conference on machine learning*. PMLR, 2016, pp. 1995–2003.
- [33] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.