AB-Mapper: Attention and BicNet based Multi-agent Path Planning for Dynamic Environment

Huifeng Guan[1*], Yuan Gao[2*], Min Zhao[2], Yong Yang[4], Fuqin Deng[1,2,4,†], Tin Lun Lam[2,3,†]

***Abstract*— Multi-agent path finding in dynamic environments is of great academic and practical value for multi-robot systems in the real world. To improve the effectiveness and efficiency of the learning process during path planning in dynamic environments, we introduce an algorithm called Attention and BicNet based Multi-agent path planning with effective reinforcement (AB-Mapper) under the actor-critic reinforcement learning framework. In this framework, on one hand, we design an actor-network that can utilize the BicNet with communication function to achieve the intra-team coordination. On the other hand, we propose a critic network that can selectively allocate attention weights to surrounding agents. This attention mechanism allows an individual agent to automatically learn a better evaluation of actions by considering the behaviours of its surrounding agents. Compared with the SOTA method Mapper in crowded environments with dynamic obstacles, our AB-Mapper is more effective (90.27±0.06% vs. 61.65±13.90% in terms of mean success rate) in solving the general multi-agent path finding problem.**

I. Introduction

The multi-agent path finding (MAPF) problem refers to solving the path planning problem for multiple agents. These agents plan to reach their goals from their starting positions while avoiding conflicts with other agents in the environment [1]. The MAPF problems in complex environments are challenging and have attracted the attention of many research groups in recent years [2].

There are mainly two classes of algorithms commonly used for solving MAPF problems. The first class includes the search-based path planning methods. Collision-based search (CBS) and its variants are the mainstream approaches [3]. These approaches generally rely on centralized planning and can be summarized into two main steps. First of all, each agent plans its path based on the single-agent path planning algorithm such as A*, while ignoring the presence of other agents. Secondly, the central planner detects if there is a collision. At the collision node, it replans an alternative path with constrained actions. However, when the number of possible collisions and the number of agents increase in the crowded environment, it leads to an exponential growth in the computational complexity of replanning [4]. This challenge leads to an increasingly long time, and it may exhaust the available computational resources and memory [5]. Due to the weak generalization capabilities, these methods in the first class are unsuitable for planning in partially observable environments. The second class includes deep reinforcement learning-based algorithms. The Path finding via Reinforcement and Imitation Multi-Agent Learning (PRIMAL) is one of the mainstream methods under the actor-critic framework [6]. This method uses the results of the search-based method as expert demonstrations to guide the learning process, and agents can reactively plan paths online in a partially observable world environment. However, it does not consider dynamic obstacles in crowded environments.

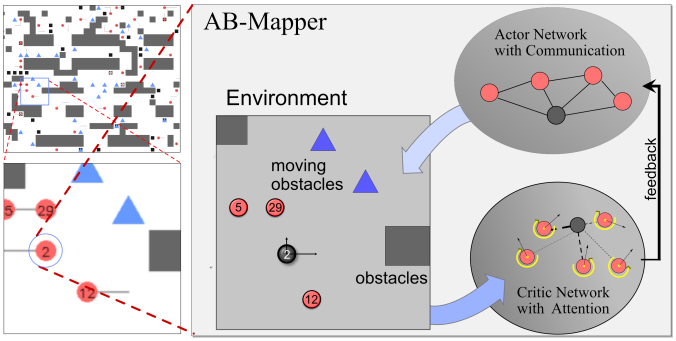

Fig. 1. Main ideas of AB-Mapper algorithm. The actor network of AB-Mapper utilizes the BicNet for enabling the agents to communicate with others, whilst the critic network uses the attention mechanism to evaluate the performance of the actor network by paying attention to related agents in the environment.

Later, Multi-agent Path Planning with Evolutionary Reinforcement learning (Mapper) was developed to model the behavior of dynamic obstacles based on the image-based representation [7]. It selects the agent with the largest reward to update the other agents at each iteration based on an evolutionary algorithm. This mechanism enables Mapper to perform path planning more effectively than PRIMAL in dynamic environments. Therefore, it becomes the state-of-the-art (SOTA) method and the baseline method of our study.

By investigating the failure cases of Mapper in various experiments, we found that the agent selected by the largest reward based on Mapper's evolutionary algorithm is sometimes not intelligent enough to handle complex environments. Since the Mapper method is a communication-free method, it can not efficiently circulate information between the agents, nor can it plan a coordinated policy in a dynamic

*Authors contributed equally, ranked alphabetically. [1]School of Intelligent Manufacturing, the Wuyi University, Jiangmen, China. [2]Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China. [3]School of Science and Engineering, the Chinese University of Hong Kong, Shenzhen, China. [4]3irobotix Co., Ltd, Shenzhen, China. †Corresponding authors are Fuqin Deng dengfuqin@cuhk.edu.cn and Tin Lun Lam tllam@cuhk.edu.cn.

This work was supported in part by the National Key R&D Program of China (2020YFB1313300), the funding (AC01202101103) from the Shenzhen Institute of Artificial Intelligence and Robotics for Society, the special projects in key fields of Guangdong Provincial Department of Education (2019KZDZX1025), the Basic and Applied Basic Research Foundation of Guangdong under Grant (2019A1515111119, 2021A1515010926), Innovative Program for Graduate Education (503170060259) from the Wuyi University, and Shenzhen Peacock Plan of Shenzhen Science and Technology Program (KQTD2016113010470345).

crowded environment [8]. Therefore, we design the Attention and BicNet based Multi-agent Path Planning with Effective Reinforcement learning (AB-Mapper) under the actor-critic framework. As illustrated in Fig. 1, we introduce the BicNet network into the actor network, allowing the agent to pass information to other agents. Furthermore, we propose an attention mechanism into the critic network, emphasizing the importance of relative agents. Compared with the SOTA method, AB-Mapper improves the effectiveness of solving MAPF problems. Our major contributions are:

- We propose the AB-Mapper method under the actor-critic framework with a limited observation of the environment. By modeling the environment with images, we use a common convolutional neural network (CNN) to extract the features of the environment observed by all the agents.
- We illustrate the importance of integrating the BicNet to fuse the extracted features by CNN in the actor network so that the agents can plan their actions based on the state information of other agents. This design enables communication among agents, speeding up the convergence of the algorithm.
- We introduce the attention mechanism into the critic network. Each agent uses this network to pay attention to the actions and states of other agents, enabling the agent to know more precisely which agents should be more beneficial to itself in the planning process. This leads to the stability and convergence of the AB-Mapper in a dynamic environment.

This paper is organized as follows: In Section II, we review related works on MAPF. In Section III, we brief the technical background of our proposed AB-Mapper method and introduce its working principles, including the BicNet network and the attention mechanism with relevant details. In Section IV, we show our experiments and then analyze the results. Finally, we sum up our study in Section V.

II. Related works

This section first reviews previous works related to the environment modeling methods for MAPF. Then we continue the discussion regarding agent communication in multi-agent reinforcement learning (MARL), which is highly related to the BicNet used in our study. After that, we briefly go through the development of the attention mechanisms related to our critic network.

A. The environment modeling methods for MAPF

There are mainly two ways to model the environment. One is based on graph structure. For example, Jiang et al. [9] modeled the relationship within a multi-agent environment using a graph, where the nodes of the graph are the agents and the encoding of the agents' local observations are the nodes' features. Furthermore, Ma et al. [10] employed graph convolution neural network to extract the relational representations among the agents and combine the potential features from the neighboring nodes. Also, Zhang et al. [11] used the graph attention network to calculate the weight of the current agent's interaction with other agents after modeling the agent into a graph. Although graph neural network can model the unstructured environment, using graph is not efficient to process dynamic obstacles in crowded environment where the size of the graph changes all the time.

The other way to model the environment is based on the image such as the widely used occupancy grid map in robotics. It contains the positions of agents and obstacles within the pixels. With CNN for feature extraction from the objects in the image, the reinforcement learning-based path planning methods have achieved a great success [12]. In addition to characterizing environmental features, CNN can also characterize path trajectory features. Deng et al. utilized CNN on an end-to-end model to achieve complex mapping functions, including extracting dynamic environmental features from raw sensor data and planning guidance commands [13, 14]. This enables the agent to initially learn navigation strategies from experts and transfer the learned strategies between different environments [15]. Wang et al. [16] used the 3D CNN layer to extract features from the trajectory planned by the global planning method for the agent in an unknown dynamic environment, which was used to assist the agent in real-time avoidance. This improves the efficiency of path planning. Among these CNN based methods, the most classical one is the PRIMAL [6]. Firstly, it uses the global planner to plan the initial paths. Then it enters the processed path trajectory by CNN for feature extraction, merges it into the actor-critic framework for training, and utilizes imitation learning based on the expert system to repair the initial paths. However, it does not consider non-cooperative dynamic obstacles nor temporal information. Therefore, the imitation learning takes longer time to train intensively in this method. Later, Mapper is developed to improve the efficiency based on the evolutionary algorithm and then becomes the SOTA method [7]. Similar to Mapper, in this paper, we also use the image to model the local environment of the agent for efficient feature extraction. Since in a sparse environment, the combination of the individual optimal actions is the optimal joint action, there is no need to consider the communication between the agents for simplicity consideration [17]. In Mapper, each agent plans actions based only on a sequence of observations and the policy from itself. However, the interactions between the agents and the dynamic obstacles make the agents have to overcome the problem of low stability and poor robustness in the planned strategies due to the non-stationarity of the environment [18]. Without communication between agents, the Mapper method can not efficiently circulate information, nor can it plan a coordinated policy in a dynamic crowded environment. Similar to Son et al. [19], we allow each agent to learn and construct its behavioral value function after aggregating the state and action information from other agents. Therefore, the agent can learn whether the actions of other agents are beneficial or harmful to itself, which helps to mitigate the issue of non-stationarity during training in the dynamic environment [20].

B. Communication in MARL

In the MARL, information exchange plays a vital role in the coordinated behavior among agents. Similar to the group foraging behavior in nature, each participant in the group may have a one-sided understanding of the environment. Therefore, communicating and sharing are beneficial to coordinated task. In recent years, establishing an efficient communication protocol between agents has become the focus of researchers. For example, Sukhbaatar et al. [21] proposed the CommNet based on continuous communication in a fully observable environment. However, as the communication network was fully symmetric and embedded in the original network, it was difficult to fuse a large amount of information. Therefore, it would perform poorly in an environment with a large number of agents [22].

The long short-term memory (LSTM) is a kind of neural network that can be used to transmit information between agents. Based on the LSTM, Singh et al. [23] designed the IC3Net to control a binary gating function. By this gating mechanism, it prevented the communication between unrelated agents. However, the IC3Net is built on a structure of unidirectional information transmission, which can not efficiently use the state information of each agent to produce a coordinated strategy. Recently, for starcraft combat games, Peng et al. [24] proposed the BicNet, as the bidirectional recurrent structure that could serve as a communication channel and local memory saver. In BicNet, each agent can maintain its internal states and share the information with other agents, which improves the communication efficiency between agents. As a consequence, the BicNet is suitable for solving multi-agent cooperative tasks, such as MAPF. We are the first to introduce the BicNet in solving the MAPF problems for dynamic and crowded environments.

C. Attention Mechanisms

Many researchers have tried to apply the attention mechanism in MAPF. For example, Shah et al. [25] proposed a novel linguistic instruction attention mechanism to score the tokens of the input visual and instruction information. This method finally used a softmax function to normalize the relative importance of each token corresponding to the current instruction. Later, Rosbach et al. [26] applied the attention mechanism in inverse reinforcement learning for predicting the reward function over an extended planning horizon.

Since evaluation and optimization of policies were the key points to improving the learning ability of the agent, in recent years, Iqbal et al. [27] designed an attention-based critic network to achieve more effective learning in a multi-agent cooperative environment. Similarly, the attention-based critic network designed by Parnika can select the optimization objective and the useful state-policy information needed to satisfy the constraints, resulting in a better policy [28]. To the best of our knowledge, this paper is the first to use the attention mechanism for the MAPF problems in dynamic and crowded environments. This mechanism can indirectly influence the predictive capability of the crowded agents and reduce collision rates while navigating the ways for them.

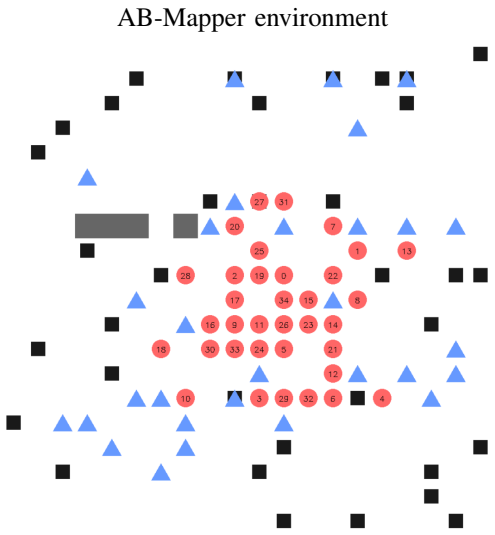

Fig. 2. The gray blocks are static obstacles; the orange circles represent the agents; the black blocks are the agents' goals; the blue triangles represent the dynamic obstacles, which move based on the A* algorithm. In every episode, the starting point is randomly initialized under the premise that the agent does not occupy the goal. If one agent occupies the position of the current agent during the initialization, this agent will wait for the previous agent to move one step before initializing.

III. Proposed Approach For Path Planning

A. Preliminaries

We model the interaction process between agent and environment as a partially observable Markov decision process, with the tuple ($\mathcal{S}$, $\mathcal{A}$, P, R, $\mathbf{o}$, ϑ, γ), where $\mathcal{S}$ represents the state space, $\mathcal{A}$ represents the action space, $P : S \times A \times S \rightarrow [0, 1]$ denotes the transition probability, $R : S \times A \rightarrow \mathbb{R}$ is a reward function, $\mathbf{o}$ represents local observation, ϑ denotes conditional local observation probability and $\gamma \in [0, 1]$ is a discount factor [29]. As the example environments shown in Fig. 2, this paper divides the local observations of the environment into three observation images with limited field of view (7×7 grid size), similar to the environment modeling method in PRIMAL [6] and Mapper [7]. The first image stores the positions of current observed static obstacles, surrounding agents and dynamic obstacles, which are represented by different values. The second image records the trajectory of surrounding agents and dynamic obstacles, which encodes the time sequence information. The third image represents the reference path planned by the A* method, based on the static environment map. Since the actor-critic framework can help the reinforcement learning algorithm to update the policy efficiently through the gradient of the current policy, it is suitable for the real-time decision-making task in path planning [30]. We also present our AB-Mapper method under the actor-critic framework. Moreover, our AB-Mapper method estimates the $Q_j\left(\mathbf{o_j}, a_j\right)$ for the agent j through off-policy temporary difference learning and minimizing the loss. The loss function of actor network is defined as

$$\mathcal{L}_A = -\sum_{j=1}^{n}\left(\log \pi_\theta\left(a_j \mid \mathbf{o_j}\right) * y_j\right), \tag{1}$$

$$y_j = r_j + \gamma Q_j\left(\mathbf{o_j'}, a_j'\right), \tag{2}$$

where the $\mathbf{o_j'}$ and a_j' are the local observation and action of the agent j at the next moment respectively. The critic network's loss function is

$$\mathcal{L}_C = \sum_{j=1}^{n}\left(Q_j\left(\mathbf{o_j}, a_j\right) - y_j\right)^2. \tag{3}$$

B. Method

The flow of the AB-Mapper algorithm is mainly composed of three steps: firstly, a CNN is used to extract the features from the state; the features are then fed into the BicNet to predict the actions; finally, the observations and actions are fused into the critic network for joint optimization of the policy.

Similar to Mapper, we use a 6-layer CNN to extract features from the observations. In Mapper, each individual agent uses a CNN to extract features from its own observations of the environment, without communication. For efficient fusion of the information, we first collect the local observations of all the agents in the actor network as shown in Fig. 3. Then, all the output features are fed into the BicNet to plan the actions.

The BicNet has the structure of a bidirectional LSTM, which can transmit information from the forward direction and the reverse direction simultaneously [24]. This structure enables the actor network to obtain more contextual information to plan the actions. And each output node at every timestep contains other agents' state contextual information. Therefore, at each step of sequential decision-making, each agent can maintain its internal state and share information with its collaborators, thus coordinating with other agents' actions. To the best of our knowledge, we are the first to apply the BicNet for MAPF in dynamic and crowded environments.

In Mapper, each agent has an independent critic network, but it lacks of information interaction between critic networks, making it impossible for a single agent to know which agent's state-action information may have an impact on itself. Applying attention mechanism in the critic network is an effective method to solve the above problem. The main idea of the attention mechanism is to calculate the attention weight through the query-key system [31]. It is a mainstream practice to influence the prediction of Q values based on the attention weights. Similar ideas have been used for the tracking task of a tiny number of agents in some multi-agent tasks [27, 28]. As shown in Fig. 4, we adapt this idea to the critic network for the evaluation of Q value. This centralized learning method allows each agent to learn the state-action information of other agents that affect the current agent in solving the MAPF problems. With the attention mechanism in the centralized critic network, we can assign the attention weight to the embedding representation of the state-action information for each agent, and then calculate the Q value. Specifically speaking, we encode the state-action information $(\mathbf{o_1}, a_1, \ldots, \mathbf{o}_j, a_j, \ldots, \mathbf{o}_n, a_n)$ of all agents through their respective encoders $(g_1, \ldots, g_j, \ldots, g_n)$ and then fed it to the

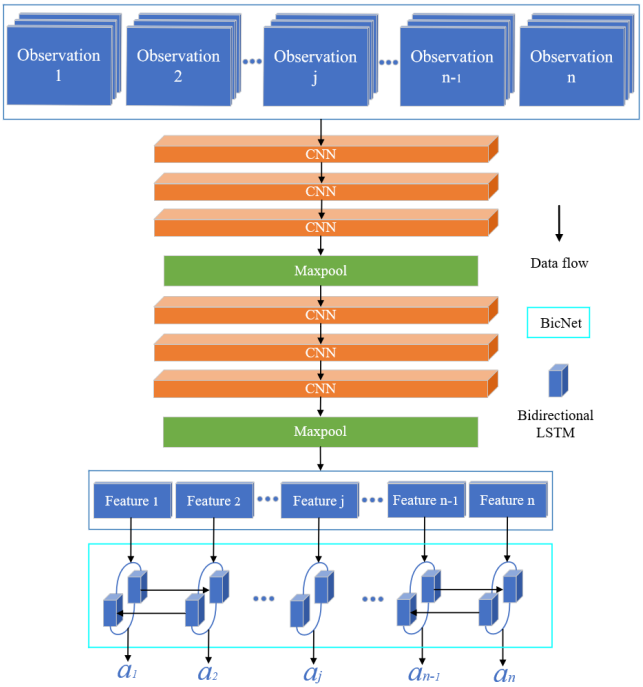

Fig. 3. AB-Mapper's actor network. The local observations of each agent as a batch are fed to CNN. The output state features of CNN are introduced into the BicNet to plan coordinated actions.

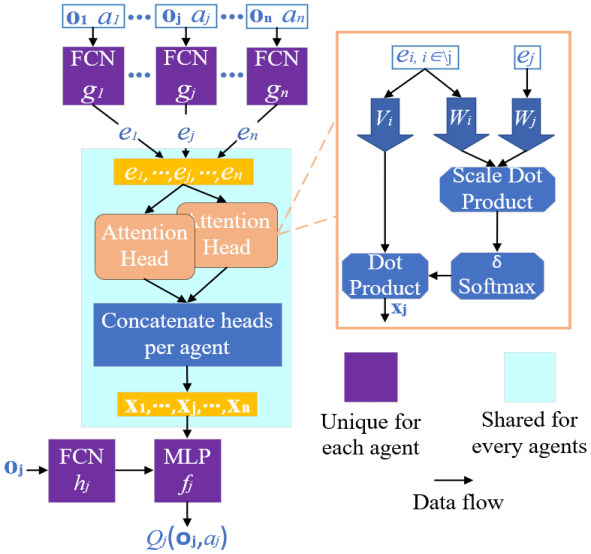

Fig. 4. AB-Mapper's critic network. In the attention module, the W_j transmutes e_j into a query $\mathbf{q}_j$ and W_i transmutes e_i into a key $\mathbf{k}_i$. The V_i transmutes e_i into a value $\mathbf{v}_i$.

critic network. For each agent j, the encoder is a one-layer fully connected network (FCN), specified as

$$\mathbf{e}_j = g_j(\mathbf{o}_j, a_j). \tag{4}$$

At every timestep, we input the encoded information of $\mathbf{o}_j$ by another FCN $h_j(\cdot)$, and $\mathbf{x}_j$ into a MLP $f_j(\cdot)$ to obtain the Q value for the agent j,

$$Q_j(\mathbf{o}_j, a_j) = f_j(h_j(\mathbf{o}_j), \mathbf{x}_j), \tag{5}$$

where $\mathbf{x}_j = \sum_{i \in \setminus j} \alpha_i \mathbf{v}_i$ is the weighed sum of $\mathbf{v}_i$, the state-action encodings of the remaining agents with contributions to the agent j, and α_i is the associated attention weight,

defined as

$$\alpha_i = \delta((\mathbf{q}_j \mathbf{k}_i^T)/\sqrt{d_k}). \quad (6)$$

$\delta(\cdot)$ is a softmax function, d_k is the dimension of $\mathbf{k}_i$, and $\mathbf{q}_j$ denotes state encoding of the agent j.

IV. Experiments

A. Experimental setup

To verify the performance of the proposed AB-Mapper method on solving the MAPF problems, we have conducted various comparison experiments with the SOTA methods. Since Mapper has been proved to outperform the benchmark methods such as the search-based local repair A* and learning-based PRIMAL [6], we mainly focus on the comparison of the performance with Mapper at the same environments and illustrate the improvement of the proposed method based two sets of experimental results.

In the first set of experiments, there are 6 Mapper environments provided by the author. In comparison experiments, we use the source code provided by the original author. Also, the environmental setting such as the number of agents and the number of dynamic obstacles, are exactly the same. In the second set of experiments, for better illustration of the performance at dynamic crowded environments, we construct a novel AB-Mapper environment for experiments, where the agents start in a clustered region, as shown in Fig. 2. Furthermore, we conduct the ablation study and compare the performance of the Mapper, Mapper(+BicNet), Mapper(+Attention) and our AB-Mapper to investigate the contributions of the BicNet and the attention parts in solving the MAPF problems.

For quantitative comparison, similar to the SOTA methods, the success rate, defined as the number of agents successfully reaching their goals over the total number of agents, is the main metric to evaluate the performance in all the experiments. Since the Mapper method can converge when the number of training before 2000 [7], we use the same setup and set the training episode to 2000 in this paper. The resultant success rates for AB-Mapper are above 80% in all the environments. The trend and conclusion are similar when we use more episodes during training in the experiments. In each episode, the maximum moving step of the agent can be dynamically adjusted, which is four times the length of the path planned and the upper limit is 50. In this paper, we independently conduct experiments three times for each environment, where the random seed for each experiment is set differently. After the convergence of the algorithms, we test each environment 90 times to obtain the mean and variance of the success rate, as demonstrated in Table II. All experiments were completed on a computer equipped with Intel i9-9900k CPU and NVIDIA 3090 graphics card with 24GB memory.

B. Experimental parameters

We use the same reward mechanism with Mapper. The total reward is $R = r_s + r_c + r_o + \lambda r_f + r_g$, where r_s, r_c and r_o are the penalties for each action, collision with an obstacle, oscillatory movement, respectively. Similar to Mapper, we use the global planner A*, ignoring the dynamic agents and obstacles, to generate the reference path S and guide the training process. An additional off-route penalty r_f is introduced to penalize the agent when it's current position p_a deviates from the reference path S. The λ is 0.3 for the relative weight of the off-route penalty. When the agent arrives at the goal position, a reward r_g is given to the agent. Table I summarizes all the values for the reward design and the reward discount factor γ is 0.99.

In all the experiments, the training parameters of Mapper are also consistent with the optimized parameters in the original paper, with the learning rate 0.0003, and the evolutionary iteration is updated once every 100 episodes. In addition to these common experimental parameters, in both Mapper(+BicNet) and AB-Mapper methods, the learning rate for the BicNet part is 0.000075. In both Mapper (+Attention) and AB-Mapper methods, the learning rate of the critic network is 0.0001, and the soft update parameter τ for the critic network is 0.001.

TABLE I

Reward mechanism

Reward	Value
Step penalty r_s	-0.1 (move) or -0.5 (wait)
Collision penalty r_c	-5
Oscillation penalty r_o	-0.3
Off-route penalty r_f	$-\min_{\boldsymbol{p}\in\mathcal{S}} \|\boldsymbol{p}_a - \boldsymbol{p}\|_2$
Goal-reaching reward r_g	30

C. Experimental results of Mapper environments

Table II summaries the statistical results in various Mapper environments. The first three columns record the dimension of the map, the numbers of agents and dynamic obstacles, respectively. From the mean success rate in all these environments, we see that the AB-Mapper outperforms the SOTA Mapper. Moreover, for the same size of the environment, such as 20×20, when the numbers of agents and obstacles are small, there are not significant differences between these two methods as shown in the first row of the data. As the numbers of agents and obstacles increase, the performance gap also increases as shown in the third row of the data. Similar trend can be seen from the fourth and fifth rows on the 65×65 environment. Also, in these two complex environment, the performances of Mapper degrade a lot, with less than 60% mean success rate, while the AB-Mapper can still maintain the mean success rate over 80%. These results illustrate that the AB-Mapper is more suitable for solving MAPF problems in complex environments, with more agents and obstacles.

TABLE II

Comparison of success rate for different algorithms.

Mapper environment			Success Rate: Mean±Std	
Size	Agent	Dynamic obstacle	Mapper	AB-Mapper Ours
20×20	15	10	86.33±0.67%	88.00±1.17%
20×20	35	30	89.92±1.64%	**90.36±0.21%**
20×20	45	30	78.56±3.22%	**89.72±0.61%**
60×65	70	100	57.28±39.75%	**83.59±7.37%**
60×65	130	140	45.04±28.45%	**88.75±0.24%**
120×130	150	40	94.03±1.7%	**99.05±0.25%**

In order to investigate the reason why the mean success rate of the Mapper decreases dramatically and its standard deviation increases dramatically on the crowded environment 60×65, with 70 agents and 100 obstacles, we log the details of evolution algorithm part at each iteration in Mapper, and visualize the success rates within the training process, for both the successful case and the failure case, as shown in Fig. 5(a). Since the evolutionary algorithm in Mapper selects the agent, with the largest accumulated reward as the candidate agent after every 100 episodes, to replace other agents with low rewards, the agent with largest reward may be the one reaching the goal position, or the one with less penalty, but not reaching the goal position in complex environments. We display the success rate, collision rate and the numbers of episodes for the candidates not reaching the goal positions in Fig. 5(a), (b) and (c), respectively. For the failure case displayed by the blue line in (a), the success rate decreases after 200 training episode, although the collision rate is relative low as shown by the blue line in (b). This is because the numbers of episodes for the candidates not reaching the goal positions are large as shown by the blue bar chart in (c) and these candidates dominate later training process. Actually, in the complex environment, it is difficult to identify the best candidate by simple rules. It lacks of diversity to update other agents by the candidate selection mechanism in Mapper. For the successful case, both the collision rate and the numbers of episodes for the candidates not reaching the goal positions are relative small as displayed by red color in (b) and (c), resulting in the ever increasing success rate in (a) during the experiment. Therefore, there may exist fluctuation in results for the Mapper method. Instead, in the proposed AB-Mapper method, we add communication mechanism to fuse the information from others, and design the attention mechanism to emphasize the relative weight of related agents to update the policy of the current agent adaptively.

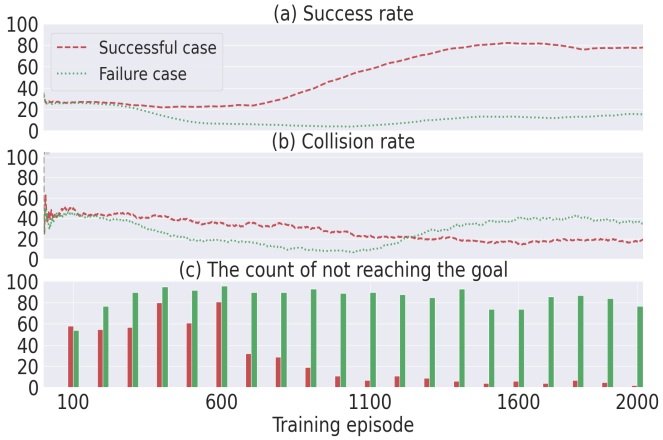

Fig. 5. Detail statistical results of the Mapper algorithm for the successful case by red color and the failure case by green color, respectively. (a) shows the success rate of the agents; (b) shows the collision rate of candidate agents; (c) counts the number of times that the candidate agent fails to reach the goal in every 100 episodes.

D. Experimental results of AB-Mapper environment

In order to quantitatively verify the improvement contributed by the BicNet and the attention mechanism, we setup the AB-Mapper environment, as shown in Fig. 2 and conduct the ablation studies. Fig. 6 displays the curves of the success rate of Mapper, Mapper(+Attention) and AB-Mapper respectively. We can see that the success rates for both Mapper(+BicNet) and AB-Mapper have grown steadily before the first 1100 episodes, and outperform the other two methods. This illustrates that the information interaction by BicNet can help the actor network to plan a coordinated strategy efficiently. After 1100 episodes, the success rate curve of Mapper(+BicNet) has a slower growth. This phenomenon occurs because the information interaction in the BicNet is continuous, and the information from the first agent flows to the last agent which cannot provide the current agent with accurate information from the highly relevant agents. The attention mechanism can help to improve the performance further by assigning more weights to highly relevant agents, as shown by AB-Mapper result. Similar to the previous failure case analysis for Mapper, due to evolutionary mechanism, the Mapper(+ Attention) has a large variance than AB-Mapper and Mapper (+BicNet).

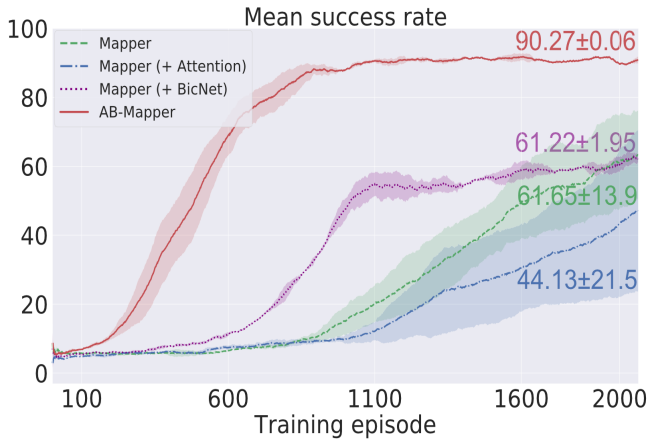

Fig. 6. The comparison result of mean success rate in AB-Mapper environment. It can be seen that when the environment becomes crowded, compared with AB-Mapper, Mapper and Mapper (+Attention) have a low success rate and large variance.

In summary, the proposed AB-Mapper uses the BicNet to transmit the state contextual information between agents to improve the communication efficiency. Furthermore, it adds an attention mechanism to combine the state and policy information of other relevant agents to refine the training process. With the communication mechanism and attention mechanism in AB-Mapper, the agents can reach the goal with a higher success rate than the SOTA method in dynamic environments.

V. Conclusion

To solve the MAPF problems in dynamic and crowded environments, we propose the AB-Mapper method under the actor-critic reinforcement learning framework after investigating the SOTA method in detail. On one hand, in the actor network, we introduce the BicNet as a communication module to improve the efficiency of communication between agents in the partially observable environment; On the other hand, we add the attention mechanism to improve the effectiveness of evaluation on an agent's policy by emphasizing

the weights of the relevant agents. Experimental results demonstrate that the AB-Mapper outperforms the SOTA method in various dynamic and crowded environments. Compared with the SOTA method, the AB-Mapper has better stability, but its scalability is relatively poor. Therefore, improving its scalability is our future work direction.

VI. Acknowledgement

We thank Zuxin Liu for providing the Mapper source code in experiments.

References

[1] Roni Stern et al. "Multi-agent pathfinding: Definitions, variants, and benchmarks". In: *Twelfth Annual Symposium on Combinatorial Search*. 2019.

[2] Jiri Švancara et al. "Online multi-agent pathfinding". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 1. 2019, pp. 7732–7739.

[3] Jiaoyang Li et al. "Symmetry-breaking constraints for grid-based multi-agent path finding". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 6087–6095.

[4] Ariel Felner et al. "Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges". In: *Tenth Annual Symposium on Combinatorial Search*. 2017.

[5] Guni Sharon et al. "Conflict-based search for optimal multi-agent pathfinding". In: *Artificial Intelligence* 219 (2015), pp. 40–66.

[6] Guillaume Sartoretti et al. "Primal: Pathfinding via reinforcement and imitation multi-agent learning". In: *IEEE Robotics and Automation Letters* 4.3 (2019), pp. 2378–2385.

[7] Zuxin Liu et al. "Mapper: Multi-agent path planning with evolutionary reinforcement learning in mixed dynamic environments". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 11748–11754.

[8] Hongda Qiu. "Multi-agent navigation based on deep reinforcement learning and traditional pathfinding algorithm". In: *arXiv preprint arXiv:2012.09134* (2020).

[9] Jiechuan Jiang et al. "Graph convolutional reinforcement learning". In: *arXiv preprint arXiv:1810.09202* (2018).

[10] Ziyuan Ma, Yudong Luo, and Hang Ma. "Distributed Heuristic Multi-Agent Path Finding with Communication". In: *arXiv preprint arXiv:2106.11365* (2021).

[11] Tianle Zhang et al. "Robot Navigation among External Autonomous Agents through Deep Reinforcement Learning using Graph Attention Network". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 9465–9470.

[12] DA Yudin et al. "Object detection with deep neural networks for reinforcement learning in the task of autonomous vehicles path planning at the intersection". In: *Optical Memory and Neural Networks* 28.4 (2019), pp. 283–295.

[13] Fuqin Deng et al. "Abnormal Occupancy Grid Map Recognition using Attention Network". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 8666–8672.

[14] Fuqin Deng et al. "FEANet: Feature-enhanced attention network for RGB-thermal real-time semantic segmentation". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 4467–4473.

[15] Mark Pfeiffer et al. "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots". In: *2017 ieee international conference on robotics and automation (icra)*. IEEE. 2017, pp. 1527–1533.

[16] Binyu Wang et al. "Mobile robot path planning in dynamic environments through globally guided reinforcement learning". In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 6932–6939.

[17] Yu Fan Chen et al. "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning". In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 285–292.

[18] Shayegan Omidshafiei et al. "Deep decentralized multi-task multi-agent reinforcement learning under partial observability". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2681–2690.

[19] Kyunghwan Son et al. "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5887–5896.

[20] Samir Wadhwania et al. "Policy distillation and value matching in multiagent reinforcement learning". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 8193–8200.

[21] Sainbayar Sukhbaatar, Rob Fergus, et al. "Learning multiagent communication with backpropagation". In: *Advances in neural information processing systems* 29 (2016), pp. 2244–2252.

[22] Jiechuan Jiang and Zongqing Lu. "Learning attentional communication for multi-agent cooperation". In: *arXiv preprint arXiv:1805.07733* (2018).

[23] Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. "Learning when to communicate at scale in multiagent cooperative and competitive tasks". In: *arXiv preprint arXiv:1812.09755* (2018).

[24] Peng Peng et al. "Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games". In: *arXiv preprint arXiv:1703.10069* (2017).

[25] Pararth Shah et al. "Follownet: Robot navigation by following natural language directions with deep reinforcement learning". In: *arXiv preprint arXiv:1805.06150* (2018).

[26] Sascha Rosbach et al. "Planning on the fast lane: Learning to interact using attention mechanisms in

path integral inverse reinforcement learning". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 5187–5193.

[27] Shariq Iqbal and Fei Sha. "Actor-attention-critic for multi-agent reinforcement learning". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2961–2970.

[28] P Parnika et al. "Attention Actor-Critic algorithm for Multi-Agent Constrained Co-operative Reinforcement Learning". In: *arXiv preprint arXiv:2101.02349* (2021).

[29] Jinyoung Choi et al. "Fast adaptation of deep reinforcement learning-based navigation skills to human preference". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 3363–3370.

[30] Can Xu et al. "An actor-critic based learning method for decision-making and planning of autonomous vehicles". In: *Science China Technological Sciences* 64.5 (2021), pp. 984–994.

[31] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.