ALPHA: Attention-based Long-horizon Pathfinding in Highly-structured Areas

Chengyang He[1], Tianze Yang[1], Tanishq Duhan[1], Yutong Wang[1], Guillaume Sartoretti[1]

***Abstract*— The multi-agent pathfinding (MAPF) problem seeks collision-free paths for a team of agents from their current positions to their pre-set goals in a known environment, and is an essential problem found at the core of many logistics, transportation, and general robotics applications. Existing learning-based MAPF approaches typically only let each agent make decisions based on a limited field-of-view (FOV) around its position, as a natural means to fix the input dimensions of its policy network. However, this often makes policies short-sighted, since agents lack the ability to perceive and plan for obstacles/agents beyond their FOV. To address this challenge, we propose ALPHA, a new framework combining the use of ground truth proximal (local) information and fuzzy distal (global) information to let agents sequence local decisions based on the full current state of the system, and avoid such myopicity. We further allow agents to make short-term predictions about each others' paths, as a means to reason about each others' path intentions, thereby enhancing the level of cooperation among agents at the whole system level. Our neural structure relies on a Graph Transformer architecture to allow agents to selectively combine these different sources of information and reason about their inter-dependencies at different spatial scales. Our simulation experiments demonstrate that ALPHA outperforms both globally-guided MAPF solvers and communication-learning based ones, showcasing its potential towards scalability in realistic deployments.**

I. INTRODUCTION

As artificial intelligence (AI) improves by leaps and bounds, robots/agents, now more than ever, can be deployed in man-made structures such as warehouses, seaports, and airports [1], [2], [3], [4], to assist with the transportation of goods and personnel. In such cases involving multiple agents within a known, static environment, an essential sub-task is to plan collision-free paths for all agents from their current positions to their pre-set goals [5]. This problem is known as Multi-Agent Pathfinding (MAPF).

The evolution of neural networks has impacted MAPF by introducing learning-based approaches, leading to a growing trend within the community to develop such methods. However, unlike traditional planners that have access to global states and give complete solutions [6], [7], [8], [9], [10], most existing learning-based planners rely on limited field-of-view (FoV) to make local plans. Although this reliance can lead to myopicity, particularly in highly structured environments where distant obstacles/agents should be taken into account for the planning, existing learning-based solutions still heavily depend on limited FoV due to the constraints posed by the input dimensions of Convolutional Neural Networks (CNNs) [11]. In order to provide agents with more information beyond the confines of their FoV, current methods either use part of expert paths within the FoV or let agents communicate with each other [12], [13]. However, these methods implicitly exploit partial slices of the global map rather than encoding the whole map. Consequently, the challenge at hand is how to encode and distill the vast global information into abstract forms that can explicitly assist agents in long-horizon planning. To address this problem, we introduce ALPHA, a novel MAPF planner which uses our proposed augmented graph representation to capture global map states and employ an attention-based network to achieve context learning by reasoning about nodes' inter-dependencies.

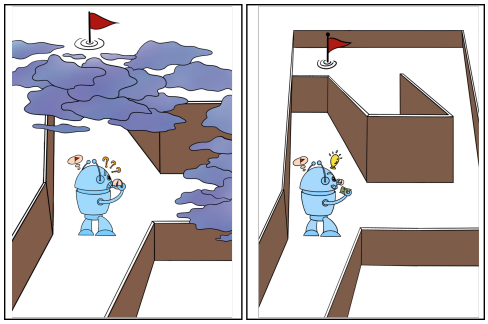

Fig. 1. Most of the existing learning-based MAPF methods rely on limited FOV, which makes the agent lack the ability to perceive and consider the environment beyond the FOV and thus leads to short-sighted policies (left example), while ALPHA enables the agent to perceive global information and consequently to make the right decisions (right figure).

Our key insight revolves around creating a low-dimensional representation of the global map to grasp the map structure and other agents' intentions, which prevents the raw data from overwhelming and confusing the network due to its dense features. We split the global information into more learnable static and dynamic channels, improving the ability of agents to understand the structure of the map and infer the intentions of other agents. To tackle static obstacles, we design a graph representation of the global map that incorporates handcrafted features for extracting structure formed by interconnected static obstacles in highly structured maps. For the dynamic side, we assume that the dynamic uncertainty in the environment solely arises from the movement of fellow agents, without the presence of unmodeled dynamic obstacles. Therefore, for dealing with other agents, we develop another graph that encapsulates the short-term intention of each agent. To process these graphs, ALPHA

[1]Authors are with the department of Mechanical Engineering, College of Design and Engineering, National University of Singapore, 21 Lower Kent Ridge Rd, Singapore {chengyanghe, yangtianze, e1280621, e0576114}@u.nus.edu, guillaume.sartoretti@nus.edu.sg

utilizes attention mechanisms to help the network prioritize critical nodes, connections, and contextual information, which enables agents to better reason about more important regions to aid decision-making. Experimental results show that with the help of these additional global graphs, ALPHA outperforms state-of-the-art non-communication learning-based MAPF solvers in both random, warehouse, and highly-structured maps (e.g. room-like environments with space-limited doors and narrow corridors). Even in congested traffic scenarios, ALPHA demonstrates comparable performance to the latest communication-based MAPF solvers.

II. RELATED WORK

In recent years, there has been growing interest in using machine learning techniques to solve MAPF. These learning-based methods can be broadly categorized into three types: relying solely on local information, combining with global guidance, and incorporating agents' communication.

Methods relying solely on local observations, such as our previous work PRIMAL [14], utilize reinforcement learning (RL) and imitation learning (IL) to provide fully decentralized solutions. PRIMAL2 [15] extends this characteristic to the life-long MAPF problem, allowing agents to learn conventions that structure and coordinate their paths. The local observations proposed by PRIMAL have demonstrated outstanding performance, inspiring many subsequent approaches to adopt this representation method. Building upon local observations, some researchers have proposed to incorporate global guidance to enhance agent performance, as seen in MAPPER [16] and G2RL [17]. However, these methods often drive agents to follow the path generated by expert single-agent algorithms to various extents, which may reduce coordination and lead to more rigid decision-making. The third class of approaches uses communication learning to exchange information with each other and achieve better individual decision-making and group coordination [18], [19]. Although communication learning enhances the accuracy of agent dynamics, the static map structure is still indirectly accessed through other agents' FoV. Thereby, due to the challenge of directly integrating whole map information with different spatial scales into neural network frameworks, these methods still use field-of-view-based representations to capture slices of global information locally. While this means might entail information loss beyond the selected slice, it is still a general method in state-of-the-art learn-based MAPF solvers because it is a natural means to fix the neural network input dimension.

Unlike previous methods, ALPHA proposes to encode global information and combines it with local observations directly without any slicing or reduction of global information. Moreover, ALPHA does not explicitly require agents to follow expert paths, which potentially enhances the flexibility and coordination of agent decisions. This paradigm provides agents with more comprehensive and long-term information, enabling agents to make better planning decisions.

III. PROBLEM STATEMENT

A. MAPF Problem Formulation

In the MAPF problem, we have n agents $A = \{a_1, ..a_n\}$ and an undirected graph $G = (V, E)$ where E is the set of edges connecting the set of vertices V. Each agent a_i is assigned a unique start vertex ($s_i \in V$) and a unique destination/goal vertex ($d_i \in V$). We call the set of all start vertices as S and the set of all destination vertices as D. We assume time to be discretized into uniform steps. At every time step, each agent executes one of the two options: i) move to one of its adjacent vertices in the graph, or ii) wait at its current vertex. The set of actions that all agents perform at a time step t_j is referred to as a joint set J_{t_j}. A joint set of actions is considered valid if no two agents occupy the same vertex at any time step: $a_{i,t_j} \neq a_{k,t_j}$, and if no two agents swap vertices in a single time step ($a_{i,t_{j+1}} = a_{k,t_j} \iff a_{k,t_{j+1}} \neq a_{i,t_j}$), where a_{i,t_j} denotes the position of agent a_i at time t_j Our objective is to determine a series of valid joint sets that guide agents from their source vertices S to their goal vertices D in the least amount of time steps.

B. Environment Setup

Remaining consistent with the standard MAPF problem, we use the following setup: the map is a 2D discrete 4-connectivity grid world, and agents can move to the free cell adjacent to their location or stay idle at each time step. An episode terminates when all agents are on their goals at the end of a time step (success) or when the number of time steps reaches the pre-defined limit (failure). Unlike previous rigid and simple room environments [5], we also look at a new room-like map generator that offers greater flexibility in terms of the number, size, and shape of rooms in the generated maps. In particular, the new room map contains corridors of varying widths, which are highly similar to real offices, warehouses, and other environments. Unlike the timeliness impact of loose obstacles in random maps, continuous obstacles in such highly structured maps may have a significant impact on current decision-making even across substantial distances.

IV. MAPF AS AN RL PROBLEM

In this section, we elaborate on our approach to processing static and dynamic information in the global map to enable agents to learn informed policies from it.

A. Observations

In order for the agent to develop long-horizon planning capability beyond its FoV, we believe that the agent's observations should be composed of ground truth local observations and fuzzy global observations. For local observations, we follow our previous work [14] to provide grid-based local information in four separate channels, giving agents the potential ability of local obstacle avoidance and coordination.

1) Global Static Observations: The process of obtaining global static observations can be divided into two phases: firstly, the extraction of a graph from the global map, and secondly, augmenting the graph with high-level features to enhance its expressiveness.

a) Graph extraction: To prevent agents from being overwhelmed by dense grid-based global information, we extract a concise graph representation from the map. We process the maps in three steps: skeletonization, neighborhood analysis, and edge generation. The skeletonization of the map is inspired by binary image thinning in machine vision for feature extraction and topological representation. We tried the Zhang-Suen algorithm and the Medial Axis Transform method. The skeleton generated by the former has fewer branches. More branches mean more information, but also more computation. We then perform neighborhood analysis on the resulting skeleton to identify *nodes* capable of representing the map structure. Concretely, we select all branch pixels (connected to at least three other pixels) and leaf pixels (connected to at most one other pixel) of the thinning map to construct our set of nodes. Finally, we obtain the edges based on the skeleton and the nodes, which use CV methods or the A* algorithm. We employed the 8-connectivity A* algorithm to identify edges between two nodes in the skeleton by checking the presence of other nodes along their connecting paths.

After graph extraction, we obtain a two-dimensional graph representation of the map with N nodes at time step t, represented by a 2D coordinate set:

$$\begin{aligned} &\mathcal{V}_t^2 = \{v_{1,t}^2, v_{2,t}^2, \cdots, v_{N-2,t}^2, v_{N-1,t}^2, v_{N,t}^2\} \\ &\forall v_{i,t}^2 = (x_{i,t}, y_{i,t}), \end{aligned} \tag{1}$$

which can only characterize some important free cells in the map. In addition to these map-generated nodes, we also add two extra nodes into the graph to represent the agent's current position $v_{N+1,t}^2$ and goal position $v_{N+2,t}^2$.

b) Augmented Static Graph: Recognizing the limitations of 2D coordinates in capturing obstacle structures, we augment each node with high-level features to enhance the graph's ability to represent global structure. Our intuition is that a node's value to a single agent pathfinding can be reflected in three aspects: i) The amount of detour (if any) an agent requires to reach a specific node (owing to obstacles) in comparison to the Manhattan distance; ii) similarly, the amount of detour required to reach the goal from the specified node; iii) the amount of deviation from the optimal path when detoured through this specific node. For this purpose, on top of the coordinates of the nodes, we set three additional features: node accessibility, detour-to-goal, and off-route degree.

- *Node accessibility*, denoting the difficulty for the agent to get from its current position to the node, is defined as follows for node i:

$$d_{i,t}^{na} = \mathcal{A}_{len}^*(v_{N+1,t}^2, v_{i,t}^2) - \mathcal{M}_{dis}(v_{N+1,t}^2, v_{i,t}^2) \tag{2}$$

- *Detour-to-goal* is used to evaluate the difficulty of reaching the agent's goal from the node i:

$$d_{i,t}^{dg} = \mathcal{A}_{len}^*(v_{N+2,t}^2, v_{i,t}^2) - \mathcal{M}_{dis}(v_{N+2,t}^2, v_{i,t}^2) \tag{3}$$

- *Off-route degree* quantifies the extent of the node's deviation from the agent's potentially optimal path to its goal.

$$\begin{aligned} d_{i,t}^{od} = \ & \mathcal{A}_{len}^*(v_{N+1,t}^2, v_{i,t}^2) + \mathcal{A}_{len}^*(v_{N+2,t}^2, v_{i,t}^2) \\ & - \mathcal{A}_{len}^*(v_{N+1,t}^2, v_{N+2,t}^2) \end{aligned} \tag{4}$$

where $\mathcal{A}_{len}^*(\cdot,\cdot)$ is the length of a path generated by the A* algorithm between two coordinates, and $\mathcal{M}_{dis}(\cdot,\cdot)$ the Manhattan distance between two coordinates. Ultimately, we obtain an augmented static graph with high-level features $\mathcal{V}_t^5$:

$$\begin{aligned} &\mathcal{V}_t^5 = \{v_{1,t}^5, v_{2,t}^5, \cdots, v_{N,t}^5\} \quad with \\ &v_{i,t}^5 = (x_{i,t}, y_{i,t}, d_{i,t}^{na}, d_{i,t}^{dg}, d_{i,t}^{op}), \quad i = 1, \cdots, N \end{aligned} \tag{5}$$

where (x_t, y_t) is the relative coordinates of the node in the map at time t, and the number of vectors in the set depends on the number of nodes in the augmented graph. We believe that, by focusing on these 5-dimensional nodes, the agent can learn to identify valuable areas in the map and develop long-horizon planning capability in single-agent scenarios. In doing so, the additional advantage of such representation is that the agent is not required to consider any specific nodes as mandatory waypoints, thus enhancing planning flexibility. A more flexible policy could enhance the prospects of agent cooperation, as it might prioritize team coordination over its own optimal policy.

2) Global Dynamic Observations: After obtaining the augmented graph used to characterize the map structure, the next step is to acquire the intentions of other agents to facilitate higher-level coordination. As mentioned earlier, given that the dynamic uncertainty in the environment derives from the movements of other agents, we interpret the intentions of other agents as their (short-term) individual-A* paths to goal. To this end, we construct a second global graph whose nodes consist of agents in the map, whose features are used to represent the corresponding agent's intention. Specifically, the features of each node consist of three components: 1) the current positions of that agent, 2) its predicted future positions, and 3) its direction of travel.

The current position of agent i is denoted by (x_{curr}^i, y_{curr}^i). For the predicted future positions of other agents, we fit a 2D Gaussian distribution with the means and variances of the short-term A* predicted paths, indicating the areas where other agents are likely to appear. Specifically, at time t, we compute the trajectories of each agent for the next f steps using A*, denoted as $tr_{t,f}^i(p_1^i, p_2^i, \cdots, p_f^i)$. Then, we calculate the mean and variance of all points in the trajectory $tr_{t,f}^i$ along the x-axis and y-axis, respectively. These means $(\mu_{f,x,t}^i, \mu_{f,y,t}^i)$ and variances $(\sigma_{f,x,t}^i, \sigma_{f,y,t}^i)$ are used to represent the likelihood of potential positions that the agent may occupy within the next f steps. Finally, we assign each agent a vector $(dx_{f,t}^i, dy_{f,t}^i, mag_{f,t}^i)$ pointing towards its predicted position after f steps, where $(dx_{f,t}^i, dy_{f,t}^i)$ is its unit direction vector and $mag_{f,t}^i$ is its the magnitude. The

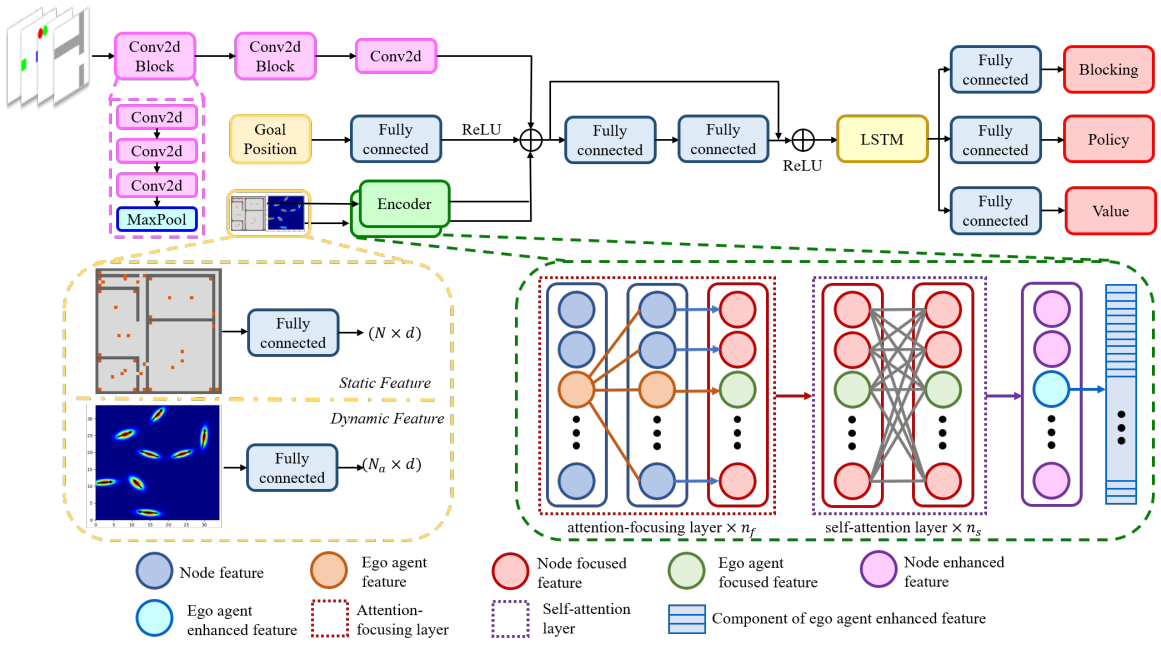

Fig. 2. ALPHA's attention-based neural network. Note that for global observation, the static features and dynamic features are passed into different encoders through different channels, and then the output features are concatenated together to form the final features of the global observation. Feature embedding: During the embedding process of nodes in the map, where the static features of any node $\forall v_{i,t}^5 \in \mathcal{V}_t^5$ need to be transformed into d-dimensional embedding vectors through a linear layer; similarly, the 9-dimensional dynamic features of all agents $\forall a_{f,t}^i \in \mathcal{D}_{f,t}$ also need to undergo a linear layer transformation to obtain d-dimensional representations. N and N_a represent the number of nodes and the number of agents in the map, respectively.

TABLE I
REWARD STRUCTURE

Action	Reward
Move to cardinal directions	-0.3
Stay idle (not on goal)	-0.3
Stay idle (on goal)	0.0
Collision with others/obstacle	-2.0
Blocking other agents	-1.0 × η

global dynamic graph used to describe the agents' intention consists of a 9-dimensional vector set $\mathcal{D}_{f,t}$:

$$\begin{aligned}\mathcal{D}_{f,t} =&\{a_{f,t}^1, a_{f,t}^2, \cdots, a_{f,t}^{N_a}\}\\ a_{f,t}^i =&(x_{curr}^i, y_{curr}^i, \mu_{f,x,t}^i, \sigma_{f,x,t}^i, \mu_{f,y,t}^i, \sigma_{f,y,t}^i,\\ &dx_{f,t}^i, dy_{f,t}^i, mag_{f,t}^i) \quad i = 1, 2, \cdots, N_a\end{aligned} \tag{6}$$

where N_a is the number of agents. It is crucial to emphasize that due to the inherent uncertainty of learning-based agents, anticipating their strict adherence to A* paths and formulating policies based on A*-informed long-term predictions can have adverse effects on the performance of the MAPF planner. But in the short term, A* paths can provide valuable insights into the agent's immediate actions, particularly in highly structured scenarios involving narrow doors and corridors.

B. Action Space and Reward

The agent's action space contains five elements, namely four cardinal directions (up, down, left, right) and stay idle. Our reward structure is shown in Table I. If the ego agent blocks other agents, it incurs a penalty of $blocking\ penalty \times \eta$, where η represents the number of blocked agents [14].

V. Attention-Based Neural Network

For local observations derived from precise grid states, we employ convolutional layers and pooling layers (inspired by VGGnet [20]) to condense the information; meanwhile, for global observations structured as graphs, we utilize encoders based on the graph transformer [21] to integrate the information. The most interesting part of this network architecture is its novel encoder, utilized to infer the inter-dependencies among nodes in both the static and dynamic augmented graphs generated from global observations. These inter-dependencies are commonly referred to as *context* [22]. Through context learning, the agent infers which of the global context-aware nodes is more important to its decision-making and constructs its policy based on this. Next, all features from different channels are concatenated and then passed through a residual block, consisting of two linear layers and a residual shortcut [23], and fed into a long-short-term memory (LSTM) cell. Finally, the agent can obtain the policy, value, and blocking through three linear layers.

1) Attention-focusing Layer: Our motivation for designing the attention-focusing layer is to enable the agent to discern the varying importance of different regions on a map, which can improve the agent's policy. In this layer, the agent infers the dependencies between itself and all other nodes, further augmenting the extent of this discrimination.

Specifically, the agent computes correlations with all other nodes in both the static and dynamic graphs, assigns attention weights, and subsequently applies them to the feature vectors. As an illustration, using the augmented static graph, at time step t, all input 5-dimensional feature vectors $v_{i,t}^5$

are transformed into d-dimensional embedding vectors u_i via a linear layer. The ego agent embedding vector u_{N+1} is extracted to calculate the query vector q_a through the weight matrix W_{fa}^Q. Similarly, the key vector k_i is calculated by passing all embedding vectors through the weight matrix W_{fa}^K, allowing us to compute the ego agent's attention for all nodes using the following formulas:

$$q_a = W_{fa}^Q u_{N+1}, \quad k_i = W_{fa}^K u_i, \quad s_{a,i} = \frac{q_a \cdot k_i^T}{\sqrt{d}},$$
$$\alpha_{a,i} = \frac{e^{s_{a,i}}}{\sum_{i=1}^{N+2} e^{s_{a,i}}}, \quad i = 1 \cdots N+2 \tag{7}$$

where $W_Q^{fa}, W_K^{fa} \in \mathbb{R}^{d \times d}$ consist of learnable parameters of the neural network. The obtained attention $\alpha_{a,i}$ is used to represent the agent's dependency for node i. We use attention for strengthening the embedding vectors u_i of interest and for weakening the embedding vectors that are not of interest:

$$u_i' = \alpha_{a,i} u_i, \quad i = 1 \cdots N+2 \tag{8}$$

By stacking several attention-focusing layers (n_f in Fig. 2), the agent's interest in different nodes can be significantly different. Agents will focus their attention on those regions that are already potentially of interest rather than viewing all nodes equally. Our encoder uses a self-attention layer [24], [25] after the attention focusing layer to reason about the inter-dependencies of nodes in the augmented graphs and perform context learning.

Our supplementary material details the computation of self-attention layers, training specifics, hardware setups, loss functions (RL and IL), and hyper-parameter selections[1][2].

VI. Experiments

In this section, we extensively test ALPHA through simulation experiments, compare its performance with state-of-the-art baselines, and conduct ablation studies to validate each proposed technique. Additionally, we validate ALPHA by deploying a trained model in various simulation environments (Gazebo) and real-world scenarios.

A. Comparison and Analysis

During training, we randomly select the size of structured environments from a uniform distribution ranging from 10 to 40, while maintaining a consistent configuration of 8 agents. While testing, we chose environments of sizes 20, 40, and 60, deploying 4 to 128 agents.

We compare our method with five other state-of-the-art MAPF solutions, namely our previous work PRIMAL [14], MAPPER [16] with global guidance, G2RL [17] with relaxed global guidance, graph neural network-based communication method DHC [18], and transformer-based communication method SCRIMP [19]. We also present results of the search-based bounded-optimal centralized planner ODrM* [6] (with inflation factor $\epsilon = 2.0$). For each test, we generated a fixed set of 100 randomly-generated environments to evaluate these MAPF planners. In other words, all planners were tested on exactly the same set of environments, and the test results are presented in Table II. We evaluate performance using three metrics: Makespan (MS) for solution efficiency via action counts, Success Rate (SR) for task completion, and Arrival Rate (AR) to gauge the percentage of goal-reaching agents, ensuring fair assessment of nearly completed episodes and avoiding their classification as complete failures.

Compared to ODrM*, all learning-based planners exhibit significant advantages in terms of AR, which is crucial for life-long MAPF. ALPHA consistently outperforms PRIMAL in nearly all cases, likely due to PRIMAL's sole reliance on local information. As the agent team size increases, the effectiveness of rigid global guidance provided by MAPPER and G2RL diminishes, leading to a decline in their performance. Notably, even in our more complex tasks (128 agents, 60×60 map size, 0.2 obstacle density) demanding high agent cooperation, which MAPPER and G2RL were completely unable to solve, ALPHA achieved a 25% SR. This is likely due to the fact that MAPPER and G2RL, to some degree, force agents to follow paths/waypoints computed by A*, and this possibly reduces their effectiveness in denser environments.

Another notable finding is that DHC and SCRIMP policies, at times, lead to certain agents getting stuck in corners because of the other agents. We believe that the same heuristic map used by DHC and SCRIMP causes this issue. ALPHA, with its comprehensive grasp of the global map and policy flexibility, does not face this limitation. Additionally, SCRIMP employs a tie-breaking strategy to enhance coordination, which is particularly effective in crowded cases. However, this also results in SCRIMP needing much more time to compute a solution. For example, in a MAPF problem with 64 agents in a map of size 20×20, ten episodes of SCRIMP take $12.29s$, while those of ALPHA take only $7.21s$ (41% less time).

While the global information furnished by ALPHA offers enhanced flexibility, a limitation compared to other global information methods is the necessity to compute the static augmented graph at each step. Generating an augmented graph for a 40×40 grid map requires approximately $0.35ms$. For ALPHA, the time needed to solve a MAPF task linearly correlates with this graph generation process.

B. Ablation Study

Our architecture is based upon two important ideas: global information and attention focusing, where global information includes static features for encoding the environment's structure and dynamic features for describing agent's intention. To analyze the importance of these three elements, we experimented with three ablation variants of ALPHA: 1) we removed the second graph and its encoder, using only static features for environment structure encoding, 2) incorporating dynamic features to predict agents' intentions, and 3) adding an attention focusing layer to enable the agent to differentiate the importance of different areas in the map. Furthermore,

[1]https://github.com/marmotlab/ALPHA

[2]https://arxiv.org/abs/2310.08350

TABLE II

EXPERIMENTAL RESULTS. THE NOTATION "↑" IMPLIES THAT A LARGER VALUE IS PREFERABLE, AND VICE VERSA.

Model	MS↓						AR↑						SR↑					
	20 × 20 room-liked environment with 4, 8, 16, 32, 64, 128 agents																	
ODrM*	30.58	43.19	97.25	292.93	512.00	512.00	100%	98.00%	88.00%	47.00%	0.00%	0.00%	100%	98%	88%	47%	0%	0%
PRIMAL	201.32	275.93	439.95	506.83	512.00	512.00	93.50%	90.88%	88.63%	81.72%	66.79%	**35.74%**	79%	67%	30%	2%	0%	0%
MAPPER	79.81	101.05	246.69	427.33	512.00	512.00	98.75%	97.53%	95.75%	89.71%	53.92%	6.96%	97%	97%	82%	41%	0%	0%
G2RL	42.22	65.46	159.12	356.94	511.45	512.00	98.00%	98.75%	98.00%	94.43%	68.50%	18.07%	99%	97%	87%	54%	1%	0%
DHC	45.50	73.86	175.22	354.43	509.69	512.00	99.00%	98.62%	96.56%	90.69%	69.70%	20.09%	98%	93%	77%	45%	1%	0%
SCRIMP	43.42	61.56	186.34	**214.32**	**488.98**	–	99.25%	99.37%	98.87%	97.53%	**82.32%**	–	98%	96%	93%	75%	**15%**	–
ALPHA	**37.39**	**52.26**	**120.65**	310.21	503.87	512.00	**100%**	**100%**	**99.75%**	**97.69%**	70.12%	25.71%	**100%**	**100%**	**96%**	**78%**	8%	0%
	40 × 40 room-like environment with 4, 8, 16, 32, 64, 128 agents																	
ODrM*	56.73	69.34	91.89	146.88	375.37	512.00	100%	100%	97.00%	85.00%	32.00%	0.00%	100%	100%	97%	85%	32%	0%
PRIMAL	285.55	384.93	463.86	492.82	511.80	512.00	91.00%	87.62%	85.56%	82.69%	73.14%	61.71%	73%	47%	23%	11%	1%	0%
MAPPER	104.82	157.12	218.35	348.95	491.58	512.00	**100%**	99.37%	98.00%	93.71%	76.60%	51.02%	**100%**	96%	91%	66%	16%	0%
G2RL	**57.60**	93.31	166.92	241.39	433.13	512.00	**100%**	99.75%	99.06%	98.65%	93.53%	70.50%	**100%**	98%	89%	81%	33%	0%
DHC	104.19	127.78	188.62	263.81	427.02	512.00	97.75%	98.00%	97.88%	95.94%	91.17%	72.53%	92%	91%	80%	65%	28%	0%
SCRIMP	58.53	91.84	**116.05**	**183.54**	396.93	**484.76**	**100%**	99.62%	99.56%	99.21%	**94.10%**	**85.09%**	**100%**	97%	95%	84%	42%	**12%**
ALPHA	64.04	**88.75**	140.96	206.85	**392.23**	506.48	**100%**	**100%**	**99.75%**	**99.34%**	93.46%	73.99%	**100%**	**100%**	**97%**	**93%**	**60%**	7%
	60 × 60 room-liked environment with 4, 8, 16, 32, 64, 128 agents																	
ODrM*	84.71	98.43	106.46	163.53	228.95	457.17	100%	100%	99.00%	88.00%	72.00%	14.00%	100%	100%	99%	88%	72%	14%
PRIMAL	363.45	465.35	495.85	508.17	512.00	512.00	84.75%	78.37%	79.75%	73.62%	71.51%	62.83%	54%	25%	11%	3%	0%	0%
MAPPER	177.61	241.31	280.69	388.55	490.02	512.00	**99.50%**	97.75%	98.31%	93.87%	85.96%	62.47%	97%	89%	90%	61%	17%	0%
G2RL	**104.40**	**140.85**	168.70	280.04	431.21	512.00	99.00%	98.62%	97.36%	95.62%	93.76%	86.01%	96%	94%	91%	68%	34%	0%
DHC	131.59	203.71	186.66	323.19	406.40	496.70	97.75%	96.75%	98.88%	95.16%	93.30%	87.79%	91%	77%	86%	54%	35%	7%
SCRIMP	106.79	166.37	**125.50**	**211.03**	421.65	498.72	**99.50%**	**99.25%**	99.61%	98.73%	96.79%	88.08%	**98%**	95%	**97%**	81%	31%	8%
ALPHA	110.82	158.59	173.02	263.74	**357.17**	**485.23**	**99.50%**	**99.25%**	**99.75%**	**99.03%**	**97.91%**	**89.16%**	**98%**	**97%**	**97%**	**86%**	**67%**	**25%**

TABLE III

ABLATION STUDY

Structural Encoding	Intent Prediction	Attention Focusing	Short-term Intention	Total Reward↑	Episode Length ↓
–	–	–	–	-249.760	230.754
✓	–	–	–	-104.412	95.229
✓	✓	–	–	-99.567	78.931
✓	✓	✓	–	-92.320	77.481
✓	✓	✓	✓	**-87.878**	**70.071**

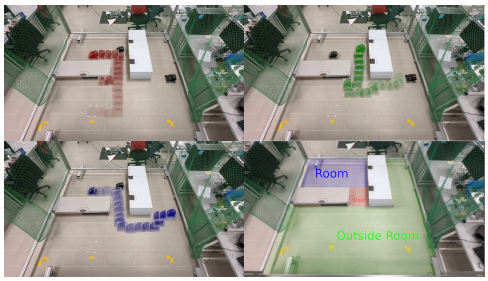

Fig. 3. Illustration of the environment and the paths of the three agents.

we attempted a fourth ablation variant of ALPHA by making shorter-term predictions (10 in practice) of agents' intentions.

The results of the ablation variants are shown in Table III. We observe that static features, which contribute to encoding the environmental structure, lead to significant performance improvements. Predicting agent intentions greatly enhances coordination and significantly reduces makespan. This aligns with our assertion that enabling the agent to understand global information facilitates better planning. Our attention focusing mechanism enhances the expressive power of the network. As mentioned in previous sections, short-term agent intention prediction outperforms long-term agent prediction, as it mitigates the inherent inaccuracies in A* prediction.

C. Experimental Validation

To illustrate the ability of ALPHA to be deployed in the real world, we test it in three standard gazebo simulation environments provided by AWS robotics[3].

Additionally, we conduct real-world experiments using three mecanum-wheeled robots ($0.2m \times 0.23m$) in an environment measuring $3.25m \times 3.25m$ containing two long, wall-like obstacles arranged to create a room with a single entrance (see Fig. 3). The agents follow policies generated by our trained ALPHA model and successfully reach their goals without collision. For more simulation and real-world experiment details, refer to the additional video.

VII. CONCLUSION

In this paper, we propose ALPHA, a novel MAPF planner designed to encode and utilize more comprehensive global information for long-horizon planning, to go beyond the use of limited FoVs in most existing learning-based MAPF planners. Our proposed augmented graph-based global observations capture both obstacle structure and agent intentions, facilitating better representation learning through these abstractions. We further utilize an attention-focusing mechanism to enhance map understanding and context learning, aiding agents in recognizing different regions' importance. Through a comprehensive range of experiments conducted on highly structured maps of varying team and world sizes, ALPHA consistently demonstrates superior performance compared to state-of-the-art learning-based MAPF baselines and a bounded-optimal search-based planner across the majority of scenarios. In doing so, we aim to offer a fresh perspective to the MAPF research community by highlighting that utilizing more effective methods for encoding a broader spectrum of global information can yield performance surpassing communication learning.

ACKNOWLEDGMENTS

This work was supported by the Singapore Ministry of Education Academic Research Fund Tier 1.

[3]https://github.com/aws-robotics

REFERENCES

[1] W. Hönig, S. Kiesel, A. Tinka, J. W. Durham, and N. Ayanian, "Persistent and robust execution of mapf schedules in warehouses," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1125–1131, 2019.

[2] 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.

[3] I. Rekik and S. Elkosantini, "A multi agent system for the online container stacking in seaport terminals," *Journal of Computational Science*, vol. 35, pp. 12–24, 2019.

[4] S. Polydorou, "A learning-based approach for distributed planning and coordination of airport surface movement operations," 2021.

[5] R. Stern, N. R. Sturtevant, A. Felner, S. Koenig, H. Ma, T. T. Walker, J. Li, D. Atzmon, L. Cohen, T. S. Kumar *et al.*, "Multi-agent pathfinding: Definitions, variants, and benchmarks," in *Twelfth Annual Symposium on Combinatorial Search*, 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] H. Ma, D. Harabor, P. J. Stuckey, J. Li, and S. Koenig, "Searching with consistent prioritization for multi-agent path finding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 7643–7650.

[9] J. Li, G. Gange, D. Harabor, P. J. Stuckey, H. Ma, and S. Koenig, "New techniques for pairwise symmetry breaking in multi-agent path finding," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 30, 2020, pp. 193–201.

[10] J. Li, Z. Chen, D. Harabor, P. J. Stuckey, and S. Koenig, "Mapf-lns2: fast repairing for multi-agent path finding via large neighborhood search," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 9, 2022, pp. 10 256–10 265.

[11] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, 2021.

[12] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 785–11 792.

[13] 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.

[14] 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.

[15] M. Damani, Z. Luo, E. Wenzel, and G. Sartoretti, "Primal _2: Pathfinding via reinforcement and imitation multi-agent learning-lifelong," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2666–2673, 2021.

[16] Z. Liu, B. Chen, H. Zhou, G. Koushik, M. Hebert, and D. Zhao, "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. 11 748–11 754.

[17] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6932–6939, 2020.

[18] 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.

[19] Y. Wang, B. Xiang, S. Huang, and G. Sartoretti, "Scrimp: Scalable communication for reinforcement-and imitation-learning-based multi-agent pathfinding," *arXiv preprint arXiv:2303.00605*, 2023.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," *Advances in neural information processing systems*, vol. 32, 2019.

[22] Y. Cao, Y. Wang, A. Vashisth, H. Fan, and G. A. Sartoretti, "Catnipp: Context-aware attention-based network for informative path planning," in *Conference on Robot Learning*. PMLR, 2023, pp. 1928–1937.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[25] W. Kool, H. Van Hoof, and M. Welling, "Attention, learn to solve routing problems!" *arXiv preprint arXiv:1803.08475*, 2018.