2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Traffic Flow Learning Enhanced Large-Scale Multi-Robot Cooperative Path Planning Under Uncertainties

Xingyao Han, Siyuan Chen, Xinye Xiong, Qiming Liu, Shunbo Zhou, Heng Zhang and Zhe Liu

Abstract— **Robotic systems with hundreds or even thousands of robots are widely implemented in logistic and industrial applications. In such systems, cooperative path planning is of great importance, as local congestion and motion conflict may greatly degrade system performance, especially in the presence of uncertainties. Our idea is to consider traffic flow equilibrium in path planning to relieve any potential congestion and increase efficiency. In this paper, we propose a hierarchical framework, which includes a traffic flow prediction layer, a sector-level planning layer, and a road-level coordination layer. In traffic flow prediction, we propose a spatio-temporal graph neural network that integrates local information to predict the evolution of future robot density distribution. In sector-level planning, we generate sector-level paths that consider travel distance and traffic flow equilibrium simultaneously. In road-level coordination, we implement the conflict-based search algorithm within each sector to ensure conflict-free local paths. In addition, we also explicitly consider motion/communication uncertainties that are unavoided in practical systems. We validate our effectiveness in simulations with over 1000 robots, what's more, real experiments are provided.**

I. INTRODUCTION

Multi-robot systems have been widely developed in various application scenarios [1]–[8]. Efficient generation of collision-free paths is the foundation for multi-robot systems to improve operational efficiency and motion coordination performance. Therefore, cooperative multi-robot path planning (MRPP) becomes a fundamental and critical problem. Many algorithms have been proposed [9]–[14] but few can be extended to large-scale scenarios. In large-scale systems with hundreds (or even thousands) of robots, motion conflict and local traffic congestion are inevitable [15]. Moreover, robots usually cannot strictly follow the planned path due to motion uncertainties (caused by human-robot interaction, control accuracy or external disturbance) and temporary communication outages (caused by bandwidth limitations, route handoff or system malfunctions) [16], leading to local conflicts and even deadlocks. The above challenges render exact optimal path planning meaningless.

A promising approach is to mitigate traffic congestion by distributing robots evenly throughout the entire space [17]. Therefore, it is critical to consider traffic flow equilibrium in path planning, especially in large-scale multi-robot systems. [15], [18] have employed probabilistic models to predict traffic flow and estimate possibly congested areas, thereby avoiding potential local conflicts. However, these models consume a significant amount of time and cannot meet the real-time requirements of practical systems. Learning-based methods that balance real-time performance and prediction accuracy have the potential to become a solution for multi-robot traffic flow prediction [19]. With the advent of graph neural networks, spatio-temporal graph neural networks (STGNNs) have emerged as the mainstream method to accurately predict urban traffic flow [20]. Nevertheless, current urban traffic flow prediction models suffer from a lack of local area awareness [21], [22], and their scalability for multi-robot systems has yet to be validated [15].

Our main idea is to use STGNNs to predict regional traffic flow and use prediction-based path planning to avoid potential conflicts. Our contributions include: 1) We propose a hierarchical framework (H3P) for solving the problem of large-scale systems with uncertainties, which predicts traffic flows, plans sector-level routes, and coordinates road-level local motions successively. Our framework contributes to enabling efficient operation and solving the real-time problems caused by large-scale systems. 2) Existing urban traffic prediction methods cannot be directly used in robotic scenarios, as they do not fully incorporate robot trajectory and environmental information within sectors. To solve these, we propose a spatio-temporal graph convolutional network for multi-robot traffic flow prediction (MR-STGCN), which integrates road structures, robot positions, and robot trajectories while consuming less time and computational resources. 3) We opensource our self-built benchmark and simulation platform which is capable of accommodating over 1000 robots simultaneously. Results show that our MR-STGCN can enhance prediction accuracy while reducing computation time compared to the state-of-arts. Finally, we verify the feasibility of our approach with real experiments.

II. RELATED WORK

A. Multi-Robot Path Planning under Uncertainties

MRPP approaches can generate sub-optimal, collision-free paths for hundreds of agents in reasonable computation time [11], [12]. However, they pay no attention to the inherent motion/communication uncertainties. To ensure practical applicability, a hierarchical planning-execution framework has been studied, taking into account kinematic constraints [23] and arbitrary dynamic constraints [24]. Nevertheless, their feasibility for large-scale problems remains unverified. A prediction-based framework is proposed for large-scale path

This work was supported by the Natural Science Foundation of China under Grant 62303307 and was also supported by Huawei Cloud Computing Technologies Co., Ltd. under the project 9450529.

X. Han, X. Xiong, and Z. Liu are with the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China. S. Chen and Q. Liu are with the Department of Automation, Shanghai Jiao Tong University, China. S. Zhou and H. Zhang are with the Edge Cloud Innovation Lab, Huawei Cloud Computing Technologies Co., Ltd. Corresponding author: Z. Liu (liuzhesjtu@sjtu.edu.cn).

planning in [15]. However, their prediction model exhibits a significant degree of computational redundancy, rendering it unsuitable for meeting the real-time demands inherent in large-scale multi-robot systems. Thus, reducing the prediction time while maintaining prediction accuracy becomes a critical issue. A detailed survey of MRPP algorithms can be found in our previous work [15], [17].

B. Spatio-Temporal Graph Neural Networks

GNNs are effective in extracting features from graph structures and achieving superior performance in various tasks [25]–[27]. To tackle sequence-related problems, [28] proposed a temporal graph convolutional network (T-GCN), which employed GCN modules to learn spatial features and GRU modules to learn temporal features, thus extracting spatio-temporal correlations. Since then, numerous studies [29]–[31] have focused on introducing various techniques and mechanisms to improve network prediction accuracy. However, none of them have introduced auxiliary networks to take into account local region information. In contrast, [21] and [22] considered internal and external factors affecting the forecast, and propose attribute-augmented STGCN and knowledge-driven STGCN. Nevertheless, these methods primarily focused on predicting urban traffic flow without fully incorporating region-specific environmental information, which may be crucial in the context of robotics.

C. Traffic Flow Prediction: Urban vs Robotic Scenarios

Urban traffic flow prediction has received significant attention in the field of data mining, in which, GNNs and STGNNs have become the mainstream solution [19], [20], [28]. Meanwhile, the emerging field of large-scale multi-robot traffic flow prediction aims to estimate robot density distribution and avoid congestion through planning. Probabilistic traffic models [18] and probabilistic robot motion models [15] have been proposed to predict congested areas, but these methods suffer from error accumulation and long computation times. *What is the gap between urban traffic flow prediction and robotic traffic flow prediction?* We summarize our observations as follows: 1) Urban traffic flow prediction typically does not incorporate local information, whereas robotic traffic flow prediction requires comprehensive consideration of factors such as robot trajectories and local roadmaps. 2) Urban traffic flow prediction may not prioritize computation time, whereas robotic traffic flow prediction serves as a module to assist in planning and must meet real-time performance requirements. 3) Urban traffic flow prediction generally does not account for vehicle actions, while robotic traffic flow prediction and multi-robot control are iterative and require a coordinated approach. Introducing STGNNs for large-scale multi-robot traffic flow prediction remains an open challenge.

III. PROBLEM FORMULATION

A. Multi-Robot Path Planning (MRPP)

Multi-robot path planning (MRPP) is described in terms of a quaternion (G,R,V_s,V_g). The roadmap $G=(V,E)$ is a topology graph that describes the connectivity of the planning space, where V and E represent the sets of vertexes and edges, respectively. The set of robots $R=\{r_1,r_2,...,r_n\}$ comprises n robots, and each robot $r_i(i=1,2,...,n)$ moves on G. The start vertex set V_s and goal vertex set V_g are subsets of V, $V_s,V_g \subset V$. At any step $k \in \mathbb{N}$, the position of r_i corresponds to a vertex v_i in G. Therefore, there is a time-variant injective map $\mathscr{F}$ from R to V. $\mathscr{F}(R)=V_s$ at $k=0$ and $\mathscr{F}(R)=V_g$ at $k=K$, where K represents the total time taken for R to move from V_s to V_g. Between two consecutive time steps, r_i can either stay in the current vertex or move to an adjacent vertex. The objective of MRPP is to minimize the total time taken by the robots to move from V_s to V_g.

In this work, both the number of vertexes $|V|$ on the roadmap G and the number of robots $|R|$ are large, making it challenging to plan in real-time. To address this issue, the environment is divided into several partitions, with each partition defined as a sector S_j. Specifically, we split G into a set of subgraphs $\{G_{xj}|j=1,2,\cdots,N_s\}$, where N_s is the number of sectors. G_{xj} represents the planning space of S_j. The connection relationship between the subgraphs is described in terms of the sector-level topology graph $G_s=(V_s,E_s),|V_s|=N_s$, and the exits and entrances between the sectors are also defined. In this way, we transform the planning problem on the large roadmap G into a planning problem on the sector-level topology graph G_s and the road-level topology graphs $\{G_{xj}\}$, reducing the consumed time.

B. Multi-Robot Traffic Flow Prediction (MR-TFP)

Traffic flow prediction (TFP) is usually defined as a spatio-temporal regression task. The inputs include S time steps and N locations. Specifically, each location represents a traffic object, which may be a road segment, intersection, or subdivision. At step k, the record of j-th traffic object is denoted as $x_j^k \in \mathbb{R}$ (e.g., speed, flow). A traffic record containing N locations can be represented as $X_N^k=[x_1^k,x_2^k,\cdots,x_N^k]^T \in \mathbb{R}^N$. The input containing S time steps can be expressed as $X_N^{(k-S+1)\sim(k)}=[X_N^{k-S+1},X_N^{k-S+2},\cdots,X_N^k] \in \mathbb{R}^{N\times S}$. The goal of TFP is to learn the map $f(\cdot)$ from the current traffic records $X^{(k-S+1)\sim(k)}$ and the topology graph G to the traffic flow after H time steps $X_N^{k+H}=f(X_N^{(k-S+1)\sim(k)};G)$.

In this work, the sectors $S_1,S_2,\cdots,S_{N_s}$ are considered as traffic objects, and the connection relationship between them is described by the sector-level topology graph G_s (mentioned in Section. III-A). Unlike the traffic objects in the city described above, we can obtain not only the number of robots in each sector but also the internal information of each sector (e.g., whether each vertex on G_{x_j} is occupied by a robot). Similar to x_j^k, the internal information of the j-th sector at step k is denoted as d_j^k (Note: d_j^k can be organized in the form of raster maps or topology maps). In addition, $D_{N_s}^k$ and $D_N^{(k-S+1)\sim(k)}$ represent the internal information of N_s sectors at step k and that in S time steps, respectively. Hence, the multi-robot traffic flow prediction (MR-TFP) problem can be described as learning the map $f(\cdot)$ such that it satisfies the equation $X_{N_s}^{k+H}=f(X_N^{(k-S+1)\sim(k)};D_N^{(k-S+1)\sim(k)};G_s)$.

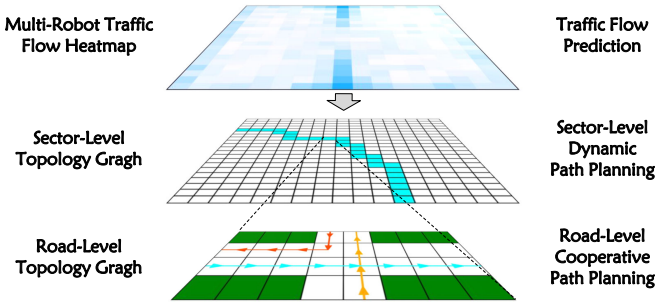

Fig. 1. The structure of our proposed framework H3P.

IV. MAIN APPROACH

A. System Framework

As shown in Fig. 1, our proposed framework H3P consists of multi-robot traffic flow prediction, centralized sector-level path planning, and distributed road-level motion collaboration: 1) The proposed MR-STGCN predicts the traffic flow in a future horizon and estimates traveling time through each sector with the predicted robot density distribution; 2) The sector-level path planner incorporates the estimated traveling time into the cost to travel through each sector and utilizes any search-based algorithm to plan sector-level paths; 3) The road-level path planner utilizes conflict-based search (CBS) [10] to generate local optimal conflict-free paths for robots within each sector. When conflicts arise due to uncertainties, an online priority-based approach is used.

B. Multi-Robot Traffic Flow Prediction

We propose a multi-robot spatio-temporal graph convolution network (MR-STGCN) as shown in Fig. 2. Unlike urban traffic prediction models, we considers not only the overall traffic flow but also the specific scenes within each sector.

Firstly, we extract the internal scene features using either CNN or GNN. Similar to [21], we divide the internal scenes of sectors into a static layer and a dynamic layer to enhance the effectiveness of the local feature extractor (LFE).

CNN-Based LFE: Each sector is represented by a binary raster map M_j with two channels. The first one is a static layer that describes static obstacles in the environment, while the second one is a dynamic layer that describes moving robots. The shape of the internal information d_j^k is $H \times W \times 2$, and we use CNN to obtain the embedding $\tilde{x}_j^k = \text{CNN}(d_j^k)$. Setting the dimension of the output F, we have $\tilde{x}_j^k \in \mathbb{R}^F$.

GNN-Based LFE: Each sector is represented by a topology graph G_{xj}, whose nodes have two attributes. The first static attribute is the node type, which describes whether the vertex is an obstacle or a movable location. The second dynamic attribute is the presence or absence of a robot. The shape of the internal information d_j^k is $N_{xj} \times 2$, where N_{xj} is the vertex number of G_{xj}. The adjacency matrix A_{xj} can be obtained from the road-level topology graph G_{xj}. We use GNN to obtain the embedding $\tilde{x}_j^k = \text{GNN}(d_j^k, A_{xj})$. Setting the dimension of the output F, we have $\tilde{x}_j^k \in \mathbb{R}^F$.

After obtaining the local features $\tilde{x}_j^k$ of step k sector S_j, we concatenate x_j^k with $\tilde{x}_j^k$ to obtain the attribute-enhanced traffic flow $e_j^k = [x_j^k, \tilde{x}_j^k], e_j^k \in \mathbb{R}^{F+1}$. Then, the MR-TFP problem is transformed to learning a map $f(\cdot)$ such that:

$$X_{N_s}^{k+H} = f(E_{N_s}^{k-S+1}, E_{N_s}^{k-S+2}, \cdots, E_{N_s}^{k} | G_s)$$

The spatio-temporal convolutional block [19] is a general framework to process structured time series. Moreover, the model consists entirely of convolutional structures and thus has fewer parameters and faster execution. Similarly, we use a fully convolutional structure for the above prediction problem to reduce the time occupied by the prediction module and to meet the real-time requirements of the system.

Secondly, graph convolutional network (GCN) [32] is used to obtain each sector's representation considering the influence of adjacent sectors:

$$Y_{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} Y_l W_l)$$

where $\sigma(\cdot)$ is the activation function, $\tilde{A} = A + I$ is the adjacency matrix with self-loops, $\tilde{D}$ is the corresponding degree matrix, $W_l \in \mathbb{R}^{C_i \times C_o}$ is the weight matrix of the l-th layer, Y_l and Y_{l+1} are the input and output representations. When $l = 0$, we have $Y_l = E_{N_s}^k$. With the enhanced traffic flow $E_{N_s}^k$ at multiple time steps and the sector adjacency matrix A_s as the input, GCN captures spatial dependencies of sectors at each time step and generates sector representations.

Thirdly, temporal convolutional network (TCN) [33] is utilized to capture the temporal dynamic trend of the traffic flow. The TCN layer consists of a 1-D causal convolution kernel Γ and a gated linear unit (GLU). For each node in G_s, the convolution kernel Γ maps Y_l to Z and the GLU is then used to implement residual connectivity between stacked temporal convolution layers, which helps to learn dependencies within long-time steps. The multi-layer TCN maps the output of the previous network to single-step prediction features $Z \in \mathbb{R}^{n \times C_o}$.

Finally, we map the learned single-step prediction features Z to the predicted traffic flow $\hat{X}_{N_s}^{k+H}$ by a multi-layer perceptron (MLP). The goal of traffic flow prediction is to make the predicted results approximate the real traffic states as much as possible. We use L2-distance to calculate the loss:

$$Loss = \sum_t ||X_{N_s}^{k+H} - \hat{X}_{N_s}^{k+H}||^2$$

where $\hat{X}_{N_s}^{k+H}$ and $X_{N_s}^{k+H}$ are the predicted and true traffic flow after H time steps, respectively.

C. Path Planning and Motion Coordination

On the centralized high level, we plan the sectors that each robot traverses from the start sector to the goal sector on G_s. The predicted traffic flow $\hat{x}_j^{t+H}$ in sector S_j in Section IV-B is considered to set the edge weights of entering sector S_j on G_s. The higher the sector traffic flow, the longer the estimated time to travel through the sector, and thus the more cost to enter the sector. Then any search-based algorithm can be employed to plan the sector-level path $[S_{i_1}, S_{i_2}, \cdots, S_{i_{N_p}}]$ for each robot (N_p is the number of sectors that the robot traverses). Supposing the global start vertex and the global goal vertex are s and d, and the exit and entrance of the

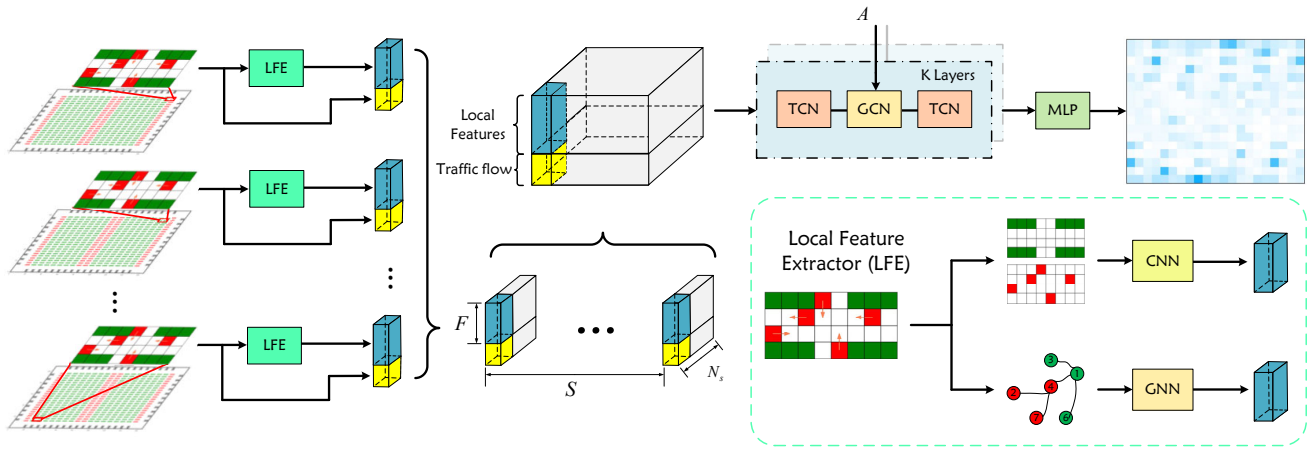

Fig. 2. The architecture of MR-STGCN. Firstly, sector information is input into Local Feature Extractor (LFE), obtaining the local sector features $\hat{x}_j^k$ (blue cuboid). We can choose either a CNN-based or a GNN-based LFE, as shown in the lower right corner. Then the local sector features $\hat{x}_j^k$ are concatenated with sector traffic flow x_j^k (yellow square), obtaining the attribute-enhanced traffic flow e_j^k. Afterward, S time steps and N_s sectors of attribute-enhanced traffic flow e_j^k are concatenated and input into K layers of ST-Conv block [19]. Finally, an MLP is used to output the predicted traffic flow of all sectors.

neighboring sector S_j to S_{j+1} are d_j and s_{j+1}, then the result of sector-level planning can be expressed as:

$$[(S_{i_1},(s,d_{i_1})),(S_{i_2},(s_{i_2},d_{i_2})),\cdots,(S_{i_{Nq}},(s_{i_q},d))]$$

Furthermore, we implement the rolling planning strategy [15] to replan the sector-level path every T_s, where T_s represents the average time to traverse the sectors. On the distributed low level, we plan the road-level paths within S_j on G_{xj}. With the sector-level planning completed, we can identify the local start vertexes (global start vertex s or sector entrance vertex s_j) and the local goal vertexes (sector exit vertex d_j or global goal vertex d) for the robots within sector S_j. Then optimal collision-free paths within S_j are generated using CBS [10].

To handle conflicts caused by uncertainties, we utilize a local priority-based approach. Specifically, the robot with the highest priority (depending on the task emergency, robot energy level, and so on) continues to move along the path, while the others adjust their paths or wait in place. To avoid conflicts caused by sector switching, we implemented an entryway reservation system to manage the authority [15].

V. IMPLEMENTATION

We develop a simulator and corresponding benchmarks that support thousands of robots, our simulator is available at *Warehousing_Robots_Simulator* [1].

A. Simulator and Datasets

In this paper, the simulation environment is a grid map of 110×166, containing 1056 robot stations, 3696 pickup stations and 672 work stations. In the beginning, each robot stays in its own station, and its task is to deliver goods from the pickup station to the assigned work station. Tasks are published at a settable frequency f_{pub}, defined as TaskPubRate, and are assigned to the nearest free robots.

Meanwhile, two typical kinds of uncertainties are considered: 1) **Motion Delay Uncertainty**: When motion delay occurs, the robot stays in the current position for one step;

[1]https://github.com/IRMVLab/Warehousing_Robots_Simulator

TABLE I
UNCERTAINTY LEVEL DEFINITION

	p_d	p_c	p_r
$\mathscr{L}_1$	0.5%	0.5%	35%
$\mathscr{L}_2$	1%	1%	30%
$\mathscr{L}_3$	1.5%	1.5%	25%
$\mathscr{L}_4$	2%	2%	20%
$\mathscr{L}_5$	2.5%	2.5%	15%

2) **Communication Failure Uncertainty**: When communication interruption occurs, the robot won't move after I steps until it reconnects again. Uncertainties are categorized into five levels, as in Table I. For each robot, p_d and p_c represent the probability of motion delay and communication interruption, respectively. For each disconnected robot, p_r denotes the probability of resuming communication. Based on the above simulator, we generate 10000 tasks, set $f_{pub} = 3$, and run the thousand-warehousing-robot system to complete the delivery tasks under varying uncertainty levels denoted as $\mathscr{L}_1, \mathscr{L}_3, \mathscr{L}_5$. At each time step, the traffic flow and the robot positions are saved separately according to sectors. Then, three datasets denoted by $\mathscr{D}_1, \mathscr{D}_2, \mathscr{D}_3$ are generated from the records, corresponding to uncertainty levels 1,3,5. The time steps of $\mathscr{D}_1, \mathscr{D}_2, \mathscr{D}_3$ are 3575, 3620, and 3616, respectively.

B. Evaluation Metrics and Comparison Models

We use the following metrics:

(1) Mean Absolute Error(MAE)

$$MAE = \frac{1}{KN_s}\sum_{k=1}^{K}\sum_{j=1}^{N_s}|x_j^k - \hat{x}_j^k|$$

(2) Root Mean Square Error(RMSE)

$$RMSE = \sqrt{\frac{1}{KN_s}\sum_{k=1}^{K}\sum_{j=1}^{N_s}(x_j^k - \hat{x}_j^k)^2}$$

(3) Weighted Mean Absolute Percentage Error(WMAPE)

$$WMAPE = \frac{\sum_{k=1}^{K}\sum_{j=1}^{N_s}|x_j^k - \hat{x}_j^k|}{\sum_{k=1}^{K}\sum_{j=1}^{N_s}|x_j^k|}$$

TABLE II

ACCURACY PERFORMANCE OF MULTI-ROBOT TRAFFIC FLOW PREDICTION WITH VARIOUS HORIZONS AND UNCERTAINTY LEVELS

Dataset	Methods	Prediction Horizon											
		3			5			10			20		
		MAE	RMSE	WMAPE	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE
$\mathscr{D}_1$	PRMM [15]	**0.4043**	1.0030	**0.1972**	0.7333	1.3636	0.3576	1.0665	1.6517	0.5203	1.5234	2.4516	0.7437
	STGCN [19]	0.6108	0.8387	0.3076	0.7303	0.9828	0.3677	0.8703	1.1467	0.4380	0.9569	1.2651	0.4812
	MR-STGCN w/C	0.5415	**0.7585**	0.2726	**0.6759**	**0.9067**	**0.3403**	**0.8334**	**1.0727**	**0.4194**	**0.9153**	**1.1888**	**0.4603**
	MR-STGCN w/G	0.5442	0.7658	0.2740	0.6876	0.9237	0.3461	0.8430	1.0901	0.4243	0.9389	1.2357	0.4722
$\mathscr{D}_2$	PRMM [15]	**0.4095**	1.0132	**0.2009**	0.7350	1.3601	0.3607	1.0638	1.6543	0.5222	1.4850	2.3915	0.7293
	STGCN [19]	0.6215	0.8382	0.3089	0.7561	0.9854	0.3758	0.8617	1.1117	0.4282	0.9517	1.2274	0.4728
	MR-STGCN w/C	0.5188	**0.7374**	0.2579	**0.6609**	**0.8837**	**0.3285**	**0.8101**	**1.0433**	**0.4025**	**0.9092**	**1.1692**	**0.4517**
	MR-STGCN w/G	0.5215	0.7435	0.2593	0.6644	0.9037	0.3303	0.8448	1.0982	0.4198	0.9453	1.2473	0.4697
$\mathscr{D}_3$	PRMM [15]	**0.4097**	1.0018	**0.2010**	0.7421	1.3643	0.3640	1.0892	1.6871	0.5345	1.5174	2.4518	0.7451
	STGCN [19]	0.5798	0.8084	0.2879	0.6950	0.9404	0.3451	0.8530	1.1249	0.4235	0.9658	1.2862	0.4794
	MR-STGCN w/C	0.5211	**0.7360**	0.2588	**0.6558**	**0.8775**	**0.3256**	**0.8158**	**1.0438**	**0.4050**	**0.9122**	**1.1710**	**0.4528**
	MR-STGCN w/G	0.5268	0.7527	0.2616	0.6589	0.8934	0.3272	0.8322	1.0837	0.4132	0.9244	1.2240	0.4589

Bold: the best value; Underline: the second-best value.

where K is the total time of time steps, and N_s is the number of sectors, while x_j^k and x_j^k respectively represent the real and predicted robot traffic flow within sector S_j at time step k.

(4) Mean Task Accomplish Time(MTAT)

$$MTAT = \frac{1}{N_{task}} \sum_{i=1}^{N_{task}} (t_{acc}^i - t_{pub}^i)$$

where N_{task} is task number, t_{pub}^i and t_{acc}^i are the published and accomplished time of $i-$th task. MTAT describes the average time step needed for each task.

(5) Mean Maximum Traffic Flow(MMTF)

$$MMTF = \frac{1}{K} \sum_{k=1}^{K} \max_{j=1}^{N_s} \frac{x_j^k}{C_j}$$

where K and N_s are the total time of accomplishing all tasks and the number of sectors, respectively. x_j^k is the number of robots within sector S_j at time step k while C_j refers to the maximum capacity of robots within the given sector. MMTF describes the average of maximum traffic flow among all sectors, A small value implies a balanced traffic flow.

We implemented the following models for comparison:

- **PRMM** [15] (Probabilistic Robot Motion Model) is the state-of-art multi-robot traffic flow prediction method based on position probability distribution.
- **STGCN** [19] (Spatio-Temporal Graph Convolutional Network) is a fully convolutional structured spatio-temporal graph network, whose effectiveness has been verified in urban traffic flow prediction.
- **MR-STGCN w/C** is our proposed STGCN with fused local information extracted by CNN.
- **MR-STGCN w/G** is our proposed STGCN with fused local information extracted by GNN.

C. Parameter Settings

The above models are trained and evaluated based on the Pytorch framework. We use the Adam optimizer, while the learning rate and weight decay are set to 0.001 and 0.0005. The batch size and proportion of the training set are manually set to 32 and 0.7. We use the data of the previous 12 frames to predict the multi-robot traffic flow in the next 3, 5, 10, and 20 frames, catering to short(3), medium(5 & 10), and long-term (20) traffic flow predictions. The output dimension F of the local feature extractor is set to 8. During the execution, the prediction horizon H is configured to be 10. All experiments are conducted on a GTX 3070 GPU with 32GB RAM.

VI. EXPERIMENTAL RESULTS

A. Multi-Robot Traffic Flow Prediction Results

1) Prediction Accuracy: Table II presents the prediction accuracy of MR-STGCN and other methods with prediction horizons of 3, 5, 10, and 20 on the datasets introduced in Section V-A. The following observations can be made:

a) In short-term prediction (with a prediction horizon of 3), PRMM achieves the best performance in terms of MAE and WMAPE. As the horizon extends, the prediction accuracy of all methods decreases. However, the magnitude of the decrease in PRMM is the greatest due to error accumulation.

b) In medium-to-long-term prediction (with prediction horizons of 5, 10, and 20), learning-based prediction methods demonstrate higher prediction accuracy in terms of all metrics compared to the state-of-art probability-based prediction method. For instance, in the case of dataset $\mathscr{D}_1$ and a prediction horizon of 10, STGCN outperforms PRMM by 30.57% in terms of RMSE.

c) In medium-to-long-term prediction (with prediction horizons of 5, 10, and 20), MR-STGCN w/C achieves the highest prediction accuracy across all metrics, while MR-STGCN w/G ranks second. In MR-TFP, MR-STGCN showed better performance than STGCN. For example, in the case of dataset $\mathscr{D}_2$ and a prediction horizon of 5, MR-STGCN w/C outperformed MR-STGCN w/G by 10.32% and 8.29% in terms of the RMSE metric, respectively.

2) Computation Time: Table III presents the computation time of MR-STGCN and other methods for 6000 time steps with prediction horizons of 5, 10, 20, and 40. As shown in the table, the computation time of the traditional probabilistic robot prediction model increases rapidly as the prediction horizon extends. This is the main motivation to introduce STGNNs into the MR-TFP problem. With the addition of local feature extractor, our MR-STGCN has an increased computational time compared to STGCN. In comparison to

TABLE III

TIME CONSUMPTION WITH VARIOUS HORIZONS

Methods \ Total Time(s)	Prediction Horizon			
	5	10	20	40
PRMM [15]	207.9	639.6	2556.8	12319.3
STGCN [19]	**70.7**	**72.9**	**71.8**	**72.1**
MR-STGCN w/C	76.7	76.9	74.9	78.8
MR-STGCN w/G	71.7	73.7	72.2	73.3

TABLE IV

MTAT AND MMTF UNDER DIFFERENT TASKPUBRATES WITH PLANNING TIME

Methods	TaskPubRate $f_{pub}(/frame)$						PT(s)
	3		4		5		
	MTAT	MMTF	MTAT	MMTF	MTAT	MMTF	
HCA [14]	144.33	0.2628	*	*	*	*	1.2967
H2P	147.05	0.2121	150.38	0.2576	*	*	**0.0348**
H3P	**142.43**	**0.1984**	**148.30**	**0.2178**	**157.05**	**0.2511**	0.1026

1) PT means Planning Time (s).
2) * means deadlocks occurred and tasks were not entirely accomplished.

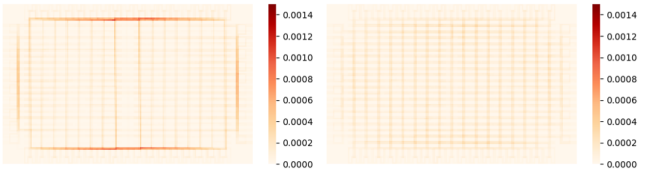

Fig. 3. Roadmap occupancy map [34] of HCA (left) and H3P (right).

MR-STGCN w/C, MR-STGCN w/G has a shorter computation time. Notably, MR-STGCN w/G has a similar computational time to STGCN but achieves greater performance improvement. For instance, with a prediction horizon of 5, MR-STGCN w/G only consumes 1.44% more time than STGCN but achieves an 8.29% increase in accuracy.

B. Multi-Robot Path Planning Results

1) Transportation Efficiency: Table IV shows the MTAT (Mean Task Accomplish Time) of each method in completing 2000 tasks at different TaskPubRates. It can be observed that our H3P has the shortest average task completion time in all cases, with H2P (H3P w/o prediction) ranked second. In addition, HCA and H2P experience deadlocks at TaskPubRates of 4 and 5, respectively. H3P, however, can successfully complete all tasks at all TaskPubRates.

2) Congestion Relief: Table IV displays the MMTF (Mean Maximum Traffic Flow) achieved by H3P and other methods in completing 2000 tasks under different TaskPubRates. From the table, we can observe that as the TaskPubRate increases, the congestion in the environment is more severe. In all cases, H3P with MR-STGCN enhancement can result in a noticeable reduction in congestion. In the best case (without considering deadlocks), H3P with MR-STGCN enhancement can reduce MMTF by approximately 24.51% compared to HCA, and by 15.56% compared to H2P.

Meanwhile, Fig. 3 illustrates the roadmap occupancy while executing 2000 tasks, with the left and right subfigures representing HCA [14] and H3P, respectively. From the figure, it is evident that HCA lacks estimation and prediction of potential congestion areas, resulting in pronounced congestion at the periphery (deep red areas). In contrast, H3P integrates considerations of environmental roadmap equilibrium, ensuring efficient utilization of the roadmap.

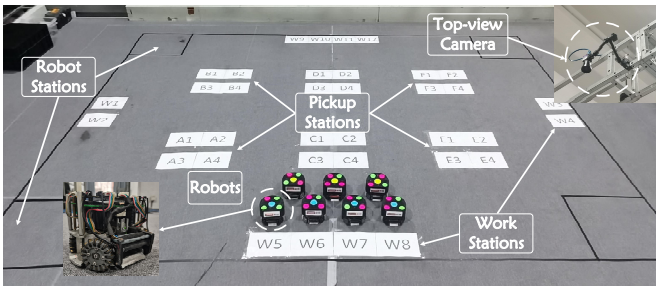

Fig. 4. Experiment System.

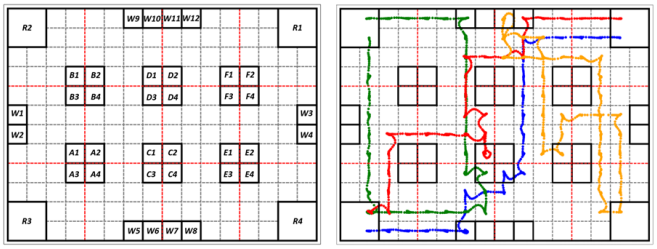

Fig. 5. Real robot trajectories recorded by top-view cameras.

3) Planning Time: Table IV also presents the Planning Time (PT) for HCA, H2P, and H3P. As observed in the table, H2P consumes only 2.6837% of the time as compared to HCA. This discrepancy stems from the fact that while HCA plans global cooperative execution paths across the entire roadmap G, H2P focuses solely on crafting global sector-level paths within the high-level graph G_s and cooperative execution paths within the local roadmap G_{xj}. Building upon H2P, H3P incorporates the prediction module and hence the time consumption increases to 7.9124% of HCA. This highlights the significant efficiency gains of the proposed hierarchical framework in reducing planning time.

C. Real Experiments

Real experiments are conducted as shown in Fig. 4. Four top-view cameras are used to localize robots through color markers. As shown in Fig. 5, the environment is divided into 12 sectors, and the movable area of each sector is discretized into grids ($0.3m \times 0.3m$, slightly larger than robot size). Each grid can only be occupied by one robot. We randomly generate tasks, randomly disable one or two robots temporarily, and randomly add control noises to simulate uncertainties. Fig. 5 shows that robots can coordinate with each other and avoid collisions, even in situations with motion uncertainties. More details can be found in *Warehousing_Robots_Simulator*.

VII. CONCLUSION

We propose MR-STGCN to predict multi-robot traffic flow and H3P framework to resolve the large-scale path planning problem under uncertainties. We validate our effectiveness and robustness by comprehensive simulations and comparisons, and conduct real experiments to verify our practical feasibility. In future work, we will demonstrate our effectiveness with large-scale real-world robotic systems.

REFERENCES

[1] H. Gao, J. Zhu, T. Zhang, G. Xie, Z. Kan, Z. Hao, and K. Liu, "Situational assessment for intelligent vehicles based on stochastic model and gaussian distributions in typical traffic scenarios," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 3, pp. 1426–1436, 2020.

[2] V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, "Ensemble coordination approach in multi-agv systems applied to industrial warehouses," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 922–934, 2015.

[3] Z. Liu, C. Suo, Y. Liu, Y. Shen, Z. Qiao, H. Wei, S. Zhou, H. Li, X. Liang, H. Wang, *et al.*, "Deep learning-based localization and perception systems: approaches for autonomous cargo transportation vehicles in large-scale, semiclosed environments," *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, pp. 139–150, 2020.

[4] K. Zafar, S. B. Qazi, and A. R. Baig, "Mine detection and route planning in military warfare using multi agent system," in *30th Annual International Computer Software and Applications Conference (COMPSAC'06)*, vol. 2, pp. 327–332, IEEE, 2006.

[5] X. Xiong, X. Han, Z. Liu, and H. Wang, "Exhaustiveness does not necessarily mean better: Selective task planning for multi-robot systems," in *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1–6, IEEE, 2023.

[6] M. Wang, M. Cong, Y. Du, D. Liu, and X. Tian, "Multi-robot raster map fusion without initial relative position," *Robotic Intelligence and Automation*, vol. 43, no. 5, pp. 498–508, 2023.

[7] 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.

[8] 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.

[9] R. J. Luna and K. E. Bekris, "Push and swap: Fast cooperative path-finding with completeness guarantees," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[10] 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.

[11] K. Okumura, M. Machida, X. Défago, and Y. Tamura, "Priority inheritance with backtracking for iterative multi-agent path finding," *Artificial Intelligence*, vol. 310, p. 103752, 2022.

[12] S. D. Han and J. Yu, "Ddm: Fast near-optimal multi-robot path planning using diversified-path and optimal sub-problem solution database heuristics," *IEEE Robotics and Automation Letters*, 2020.

[13] 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.

[14] D. Silver, "Cooperative pathfinding," in *Proceedings of the aaai conference on artificial intelligence and interactive digital entertainment*, vol. 1, pp. 117–122, 2005.

[15] Z. Liu, H. Wang, H. Wei, M. Liu, and Y.-H. Liu, "Prediction, planning, and coordination of thousand-warehousing-robot networks with motion and communication uncertainties," *IEEE Transactions on Automation Science and Engineering*, pp. 1705–1717, 2020.

[16] H. Ma, T. S. Kumar, and S. Koenig, "Multi-agent path finding with delay probabilities," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[17] Z. Liu, H. Wei, H. Wang, H. Li, and H. Wang, "Integrated task allocation and path coordination for large-scale robot networks with uncertainties," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 2750–2761, 2021.

[18] V. Digani, L. Sabattini, and C. Secchi, "A probabilistic eulerian traffic model for the coordination of multiple agvs in automatic warehouses," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 26–32, 2015.

[19] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[20] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis, and future directions," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[21] J. Zhu, Q. Wang, C. Tao, H. Deng, L. Zhao, and H. Li, "Ast-gcn: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting," *IEEE Access*, vol. 9, pp. 35973–35983, 2021.

[22] J. Zhu, X. Han, H. Deng, C. Tao, L. Zhao, P. Wang, T. Lin, and H. Li, "Kst-gcn: A knowledge-driven spatial-temporal graph convolutional network for traffic forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15055–15065, 2022.

[23] H. Ma, W. Hönig, L. Cohen, T. Uras, H. Xu, T. S. Kumar, N. Ayanian, and S. Koenig, "Overview: A hierarchical framework for plan generation and execution in multirobot systems," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 6–12, 2017.

[24] 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*, 2019.

[25] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," *Advances in neural information processing systems*, 2018.

[26] M. Prates, P. H. Avelar, H. Lemos, L. C. Lamb, and M. Y. Vardi, "Learning to solve np-complete problems: A graph neural network for decision tsp," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4731–4738, 2019.

[27] F. Errica, M. Podda, D. Bacciu, and A. Micheli, "A fair comparison of graph neural networks for graph classification," *arXiv preprint arXiv:1912.09893*, 2019.

[28] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[29] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2017.

[30] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *The 28th International Joint Conference on Artificial Intelligence (IJCAI)*, International Joint Conferences on Artificial Intelligence Organization, 2019.

[31] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 1234–1241, 2020.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2016.

[33] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[34] Y. Zhang, M. C. Fontaine, V. Bhatt, S. Nikolaidis, and J. Li, "Multi-robot coordination and layout design for automated warehousing," *arXiv preprint arXiv:2305.06436*, 2023.