Asymmetric Information Enhanced Mapping Framework for Multirobot Exploration based on Deep Reinforcement Learning

Jiyu Cheng, Junhui Fan, Xiaolei Li, Paul L. Rosin, Yibin Li, and Wei Zhang, *Senior Member, IEEE*

***Abstract*—Despite the great development of multirobot technologies, efficiently and collaboratively exploring an unknown environment is still a big challenge. In this paper, we propose AIM-Mapping, a Asymmetric InforMation Enhanced Mapping framework. The framework fully utilizes the privilege information in the training process to help construct the environment representation as well as the supervised signal in an asymmetric actor-critic training framework. Specifically, privilege information is used to evaluate the exploration performance through an asymmetric feature representation module and a mutual information evaluation module. The decision-making network uses the trained feature encoder to extract structure information from the environment and combines it with a topological map constructed based on geometric distance. Utilizing this kind of topological map representation, we employ topological graph matching to assign corresponding boundary points to each robot as long-term goal points. We conduct experiments in real-world-like scenarios using the Gibson simulation environments. It validates that the proposed method, when compared to existing methods, achieves great performance improvement.**

***Index Terms*—Multirobot system; multirobot exploration; multi-agent reinforcement learning; graph neural network**

I. Introduction

With the advancement of artificial intelligence technology, multirobot systems have been used in more and more applications. In tasks like search and rescue or inspection [1], robots often operate in unknown environments without prior access to the environment map. During multirobot exploration, robots need to use onboard sensors to perceive and reconstruct the environment efficiently while moving based on certain strategy. The early approach, frontier-based exploration [2], identifies boundaries of the explored area, guiding robots to ensure complete coverage. This kind of methods typically use geometric information from the robots and frontiers to assign either short or long term goals. As robots move, they update the map to facilitate future decision-making. However, achieving optimal decision is difficult considering robot mobility, sensor capabilities, and coordination. Some approaches model the environment using occupancy grids or distance-based topological maps, framing the collaborative exploration task as a combinatorial optimization problem [3, 4, 5]. Despite that optimization solvers can guarantee optimal solutions, they typically suffer from high computational complexity due to the NP-hard property of the problem. To address this challenge, some researchers apply heuristic methods with relaxed constraints to improve target point allocation among robots [6, 7, 8]. These methods, guided by manually designed heuristic functions, offer significant computational efficiency. However, since heuristic rules are based on human intuition, they usually lack generality and often lead to suboptimal and locally constrained decisions.

In recent years, deep reinforcement learning (DRL) has made significant breakthroughs in solving combinatorial optimization problems [9] and motion control challenges [10]. Building on this, multi-agent reinforcement learning (MARL) extends reinforcement learning to the multi-agent domain, demonstrating strong performance in various multirobot applications such as formation control [11], autonomous vehicle fleets [12], and intelligent warehousing [13]. After training, these strategies enable robots to perform complex real-time coordinated actions. The challenges of exploring unknown environments in current boundary-based and reinforcement learning scenarios include: (1) Short-sighted decision-making. Due to the property of the unknown environment tasks, long-term information may be unavailable, making it difficult to determine the information value at the current time step, leading to inaccurate immediate rewards. (2) Multirobot cooperative exploration can effectively improve the efficiency of exploring unknown environments, but the increased action space in multirobot scenarios makes it more complex to find solutions.

In this paper, we propose an efficient multirobot active mapping method called AIM-Mapping. During training, privilege information is used to address the inaccuracies and instability in state value estimation caused by the unknown environment in reinforcement learning. AIM-Mapping encodes observation features using a differential structured feature extraction network, generating state values by capturing the difference between privilege information and observation. The term ”asymmetric” refers to the information asymmetry during reinforcement learning, where the critic module has access to privilege information from unexplored areas that the actor module cannot obtain. Our method trained on only 9 indoor scenes, shows remarkable generalization across different indoor datasets and varying robot numbers. The experimental

Jiyu Cheng, Junhui Fan, Xiaolei Li, Yibin Li, and Wei Zhang are with the School of Control Science and Engineering, Shandong University, Shandong, China, 250061 (e-mail: jycheng@sdu.edu.cn; 202334916@mail.sdu.edu.cn; {qylxl, liyb, davidzhang}@sdu.edu.cn).

Paul L. Rosin is with the School of Computer Science and Informatics, Cardiff University, CF10 3AT Cardiff, U.K. (e-mail: rosinpl@cardiff.ac.uk).

(Corresponding author: Wei Zhang.)

results highlight its advantage over cutting-edge multirobot active mapping methods and several adapted reinforcement learning baselines. The main contributions are summarized as follows:

- We propose a novel multirobot active mapping framework of which the collaboration efficiency is greatly enhanced by privilege information based on an asymmetric actor-critic training design.
- We propose a new perspective on evaluating exploration performance in unknown environments, introducing privilege information to assess state value through feature engineering and mutual information.
- We adopt topological graph matching in the multirobot decision-making based on the asymmetric feature representation framework.
- The whole method is deployed and tested in the iGibson simulation environment to showcase its practicality and potential application in real-world scenarios.

The rest of this article is organized as follows. Section II introduces some related works as well as their advantages and disadvantages. Section III gives the problem formulation of the task. Section IV describes the details of our framework. The experiment implementation and the analysis of the results are presented in Section V. Finally, Section VI draws the conclusions and proposes our future work.

II. RELATED WORK

In this section, we discuss several kinds of multirobot exploration methods, including heuristic methods, optimization-based methods, information-theoretic methods, and DRL-based methods.

Heuristic methods. This kind of methods usually utilize the empirical hints to facilitate exploration. Yamauchi *et al.* [6] proposed the concept of the frontier for active mapping, aiming to guide the robot to the frontiers until the entire space is observed. However, these approaches lack coordination, often resulting in redundant exploration. Colares *et al.* [14] addressed this problem by introducing a collaboration factor to enhance target allocation efficiency. Bourgault *et al.* [15] utilized Voronoi partitioning to assign robots to frontier points within their subspace, avoiding redundant exploration. The Artificial Potential Field (APF) method creates virtual force fields to guide robots. Initially used in global path planning [16], Lau *et al.* [8] constructed a potential function based on distance to guide movement. However, potential fields based on Euclidean distance usually suffer from local optimum. Renzaglia *et al.* [17] applied potential fields to local navigation, while Liu *et al.* [18] designed a nonlinear potential function incorporating coverage factors. More recently, Yu *et al.* [19] introduced a wave-front distance metric and a penalty function for sensor overlap to reduce redundant exploration and improve efficiency.

Optimization-based methods. In multirobot exploration, target selection and path planning in unknown environments can be formulated as an Optimal Assignment Problem [20]. Werger *et al.* [3] introduced a cost function based on Voronoi partitioning and used the Hungarian Method for approximate solutions. Klodt *et al.* [4] proposed a Pairwise Optimization strategy for optimal frontier point allocation. For more complex scenarios, Dong *et al.* [5] applied a clustering algorithm for frontier points, modeling the problem as an Optimal Mass Transport Problem [21] with a path distance-based cost function. Faigl *et al.* [22] modeled target allocation as a multiple Traveling Salesman Problem [23]. Additionally, Clark *et al.* [24] modeled the problem as a queue stability control problem, employing Lyapunov optimization [25] to guide multirobot decision-making.

Information-theoretic methods. Some researchers are dedicated to approach this problem from the information theory perspective. Whaite and Ferrie *et al.* [26] introduced strategies for minimizing entropy during exploration of unknown environments, while Elfes *et al.* [27] focused on maximizing mutual information (MI) between sensor data and an occupancy grid map. Information-based exploration strategies aim to minimize uncertainty in robot localization and environment mapping [28, 29, 30]. In multi-agent systems, [31, 32] advanced the idea of using information gain to improve collaboration. In three-dimensional environments, the need for efficient computation of mutual information is particularly pronounced. Henderson *et al.* [33] proposed a faster method for continuous map computation based on recursive expressions of Shannon mutual information. Asgharivaskasi *et al.* [34] proposed semantic octree mapping and Shannon mutual information computation for robot exploration, deriving an efficiently computable closed-form lower bound for the mutual information between a multiclass octomap and a set of range-category measurements.

DRL-based methods. Recently, deep reinforcement learning is increasingly used to balance the tradeoff between computational efficiency and the optimality of the decision. Geng *et al.* [35] proposed a decentralized decision-making method in grid map environments using multi-agent reinforcement learning, and robots exchange observation information encoded by convolutional neural networks through a learnable network structure to achieve collaborative decision-making. Later, Geng *et al.* [36] improved this by introducing attention mechanisms for more targeted information exchange. However, the limited action space in grid maps often results in suboptimal long-term decisions. To address this problem, Tan *et al.* [37] introduced hierarchical reinforcement learning, extending the decision model to a hierarchical framework. Zhu *et al.* [38] proposed a Two-Stage Coordination (TSC) strategy, which consists of a high-level leader module and a low-level action executor. Researchers have applied reinforcement learning to more realistic scenarios, accounting for robot dynamics and environmental perception. Hu *et al.* [39] combined DRL with Voronoi segmentation for LiDAR-equipped robots. Chaplot *et al.* [40] designed a hierarchical framework for single-robot indoor exploration using RGB cameras, while Yu *et al.* [41] extended this to multirobot vision-based exploration with a Transformer-based decision network. Ye *et al.* [42] used depth cameras to reconstruct 2D maps and built topological graphs, introducing a multi-path graph neural network to predict distances between boundary points and robots. Lodel *et al.* [43] trained an information-aware policy via deep reinforcement

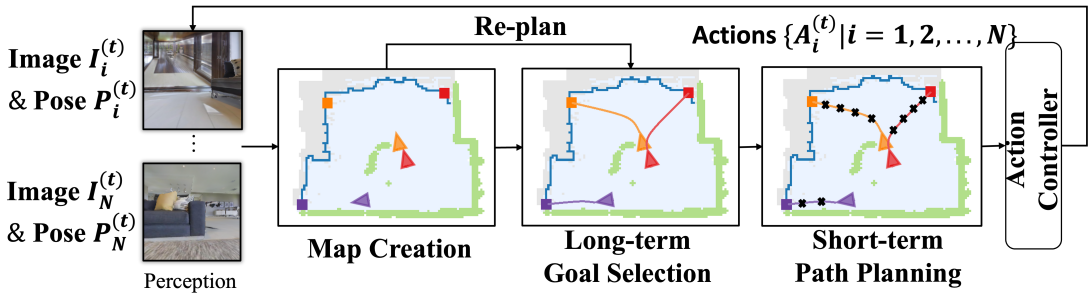

Fig. 1. Multirobot collaborative active mapping task framework contains three main sub-modules: perception and map creation, long-term goal selection, and short-term path planning.

learning, that guides a receding-horizon trajectory optimization planner. Lee *et al.* [44] proposed a spatial abstraction for multi-agent exploration, using topological graph memories with minimal image features.

Our method complements existing methods and extends the potential of Information-theoretic methods in multi-agent cooperative strategy training.

III. PROBLEM FORMULATION

A. Multirobot Active Mapping

The multirobot active mapping task can be divided into three sub-task modules: perception and map creation, long-term goal selection, and short-term path planning. For end-to-end cases, they will only output the short-term action, which are not our focus here. In the perception and map creation module, robots transform sensor information into 2D grid maps. In the long-term goal selection module, robots allocate and select long-term goal points on the grid maps. In the short-term path planning module, robots plan paths to the selected long-term goal points. The overall task framework is shown in Fig. 1. The policy network and training algorithm focus on long-term goal selection, with existing methods handling perception, map creation, and short-term path planning.

1) Perception and Map Creation: The objective of the perception and map creation module is to construct a global map based on sensor information from multiple robots. Robots often use depth cameras as distance sensors. Multiple robots transform their occupancy maps into a common world coordinate system based on their positions and orientation differences, creating a global occupancy map. At each time step, the robot obtains depth image $I_i^{(t)} \in \mathbb{R}^{h\times w}$ and the global pose $L_i^{(t)} \in \mathbb{R}^3$ from the environment. After the coordinate conversion, each pixel in the depth image $I_i^{(t)}$ is converted into a point in the 3D point cloud. The resulting point cloud can be represented as $P_i^{(t)} = (x, y, z)$, the corresponding point in the world coordinate system with the robot as the origin. Multiple robots exchange information to output their own poses $L_i^t = (x, y, \theta)$ and occupancy grid maps $M_t^{(t)}$. Each robot transforms its respective grid map to the common world coordinate system based on its pose and merges it with the previous global grid map $M^{(t-1)}$ to obtain the global grid map $M^{(t)}$ at time t.

2) Long-term Goal Selection: In long-term goal selection, a straightforward approach for each robot is to move toward the boundaries of the known and unknown map areas at each time step. Assuming that the map is closed and bounded, as long as robots continuously move toward boundary points, they will eventually complete exploration of the environment. This method based on boundary points is also adopted in this paper. Therefore, the goal is to assign a boundary point as a long-term target point for each robot when each planning cycle arrives, allowing multiple robots to explore as much of the unknown environment as possible in the shortest possible time and ultimately establish a global grid map containing all environmental information.

3) Short-term Path Planning: Short-term path planning is a discretized subtask in which robots, after receiving long-term goals, individually plan the shortest paths to reach their respective targets based on the global grid map $M^{(t)}$. In this paper, the Fast Marching Algorithm [45] is adopted to compute the shortest path from the robot to the target position. Upon obtaining the short-term path points, robots generate low-level actions through a simple heuristic method [46]: if a robot is facing the path point, it executes a forward action; otherwise, it performs rotation actions until it faces the path point.

B. Modeling of Markov Decision Process

Long-term goal selection task in the multirobot active mapping mission is crucial. In indoor scenarios, the multirobot active mapping task can be modeled as a centralized Partially Observable Markov Decision Process (POMDP). This process can be represented by a tuple $<\mathcal{N}, \mathcal{S}, \mathcal{O}, O, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{Y}>$. Where $\mathcal{N}$ is the set of N robots. $\mathcal{S}$ represents the global state space. $\mathcal{O} = \times_{i\in\mathcal{N}}$ is the joint observation space for multiple robots, where O is the observation function. $\mathcal{A} = \times_{i\in\mathcal{N}}\mathcal{A}^i$, represents the joint action space for multiple robots. $\mathcal{P} : \mathcal{S}\times\mathcal{A}\to\triangle(\mathcal{S})$ denotes the state transition probabilities. $\mathcal{R} : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$, is the reward function for all agents. $\mathcal{Y} \in [0, 1)$ is the reward discount factor. At each time step, agents receive local observations $o_i^{(t)} = O(S^{(t)}, i)$ from the global state $s^{(t)} \in \mathcal{S}$ then a controller collects observations from all agents and generates a joint action $a^{(t)} = \pi(\cdot|o_1^{(t)}, ..., o_N^{(t)})$ through a centralized policy. Each agent receives and executes the corresponding action $a_i^{(t)} \in a^{(t)}$ from the central controller. Finally, the joint actions $a^{(t)} \in \mathcal{A}$ of multiple

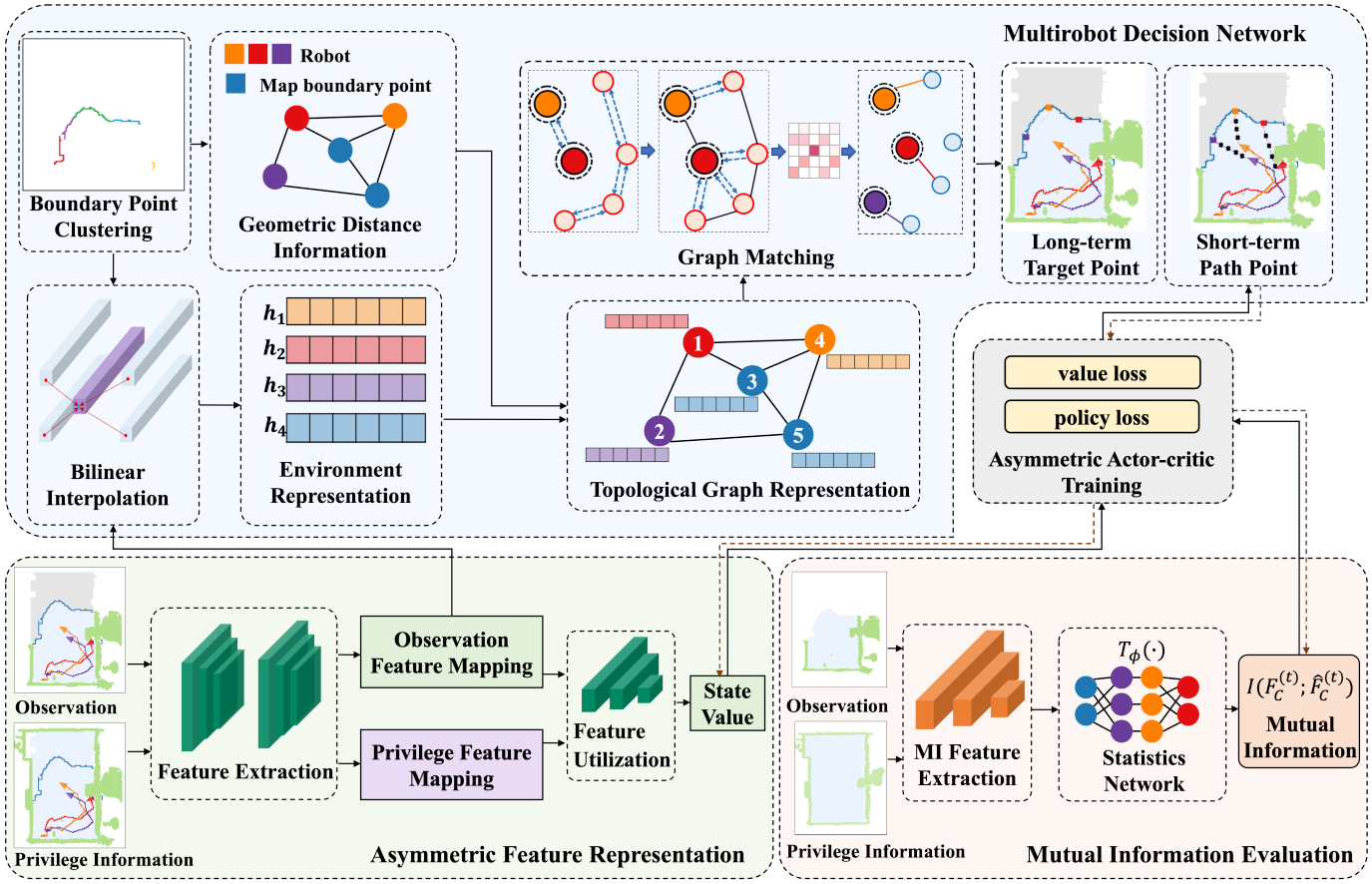

Fig. 2. The overall AIM-Mapping framework. Asymmetric Feature Representation is used to generate the state value and the observation feature mapping. Multirobot Decision Network combines the geometric distance information and structure information to formulate the topological graph representation, and adopts graph matching to generate the corresponding goal point. Mutual Information Evaluation is utilized to facilitate the training process.

agents transition the system from state $s^{(t)}$ to state $s^{(t+1)}$ based on the state transition probabilities $P(s^{(t+1)}, a^{(t)})$ and receive reward $r^{(t)} = R(s^{(t)}, a^{(t)})$. To address this problem, value networks and policy networks are designed, employing an end-to-end multi-agent deep reinforcement learning framework. This framework aims to maximize the value function $V_\pi(s) = \mathbb{E}_{s,a}[\sum_{t=0}^{T} \gamma_t r^{(t)} | s_0 = s, a\ \pi(\cdot|o_1^{(t)}, ..., o_N^{(t)})]$ to learn an optimal centralized policy $\pi * (\cdot|o_1^{(t)}, ..., o_N^{(t)})$. In the task, the action space of agents consists of a set of candidate points for long-term goals, where the action of an agent is selecting a candidate point as a long-term goal and moving towards that target point.

IV. METHODOLOGY

This paper proposes a multirobot exploration method called AIM-Mapping based on deep reinforcement learning. The core idea is to fully utilize the privilege information to help enhancing the efficiency of multirobot exploration. There are mainly three modules: *Asymmetric Feature Representation*, *Mutual Information Evaluation*, and *Multirobot Decision Network*, and the first two modules are both based on privilege information. During the training process, the Asymmetric Feature Representation leverages both observation information and privilege information to generate the value state as well as the observation feature mapping. Then, in Multirobot Decision Network, topological information based on geometric distance and structural information based on the above-mentioned observation feature mapping are both used to generate the topological graph representation. Through graph matching, the robots will be assigned the corresponding long-term target point and short-term path point. In addition, the Mutual Information Evaluation module is specially designed to facilitate the training process, utilizing the privilege information. The overall AIM-Mapping framework is illustrated in Fig.2.

A. Asymmetric Feature Representation

This section provides an overview of the Asymmetric Feature Representation, which is designed to encode the disparity of observation and privilege information. As shown in Fig. 3, the input of the module contains observation and privilege information which is obtained directly from the simulator. In order to maintain the structural properties, both types of information are grid-map-based. After that, the inputs are both expanded to 5-channel maps, including obstacle channel, passable-area channel, robot channel, boundary channel, and trajectory channel. Each channel is encoded with a binary map, where the value of a grid cell is 1 if there is corresponding entity information, otherwise the value is set as 0. Specifically, the observation is represented as $O_c^{(t)} \in \{0,1\}^{X \times Y \times 5}$, where X and Y are the dimensions of the global map. And the input

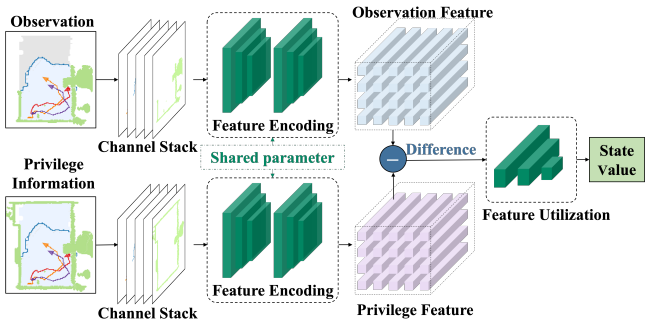

Fig. 3. The Asymmetric Feature Representation network. Given the observation and privilege information as the input, the module first expands them into 5-channel maps, and extracts the feature of the above two kinds of information. The feature disparity will be fed into a Feature Utilization network, and the output of the network is used as the state value.

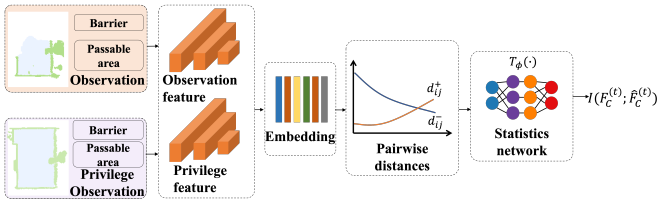

Fig. 4. The Mutual Information Evaluation network. Embedding the observation feature and privilege feature, the pair distance is calculated by merging the two sets of features. Then the mutual information is evaluated by the statistical network.

privilege information is represented as $\hat{O}_c^{(t)} \in \{0,1\}^{X\times Y\times 5}$. A feature encoding network is designed to extract the structure information of the environment using a convolutional neural network (CNN) as the backbone. The feature encoding network reduces the dimension of the input information while increasing the number of channels, extracting structure feature from the spatial domain into corresponding channel vectors. If the input size is $X \times Y \times 5$, the output size of the feature mapping is $X_h \times Y_h \times C_h$, where $X_h = X/8$, $Y_h = Y/8$, and $C_h = 32$ are selected in this method. At time t, the privilege information and observation, after feature extraction by the aforementioned feature encoding network, yield privilege feature mappings $\hat{F}_c^{(t)} \in \mathbb{R}^{X_h\times Y_h\times C_h}$ and observation feature mappings $F_c^{(t)} \in \mathbb{R}^{X_h\times Y_h\times C_h}$, respectively. The sizes of privilege feature mappings and observation feature mappings are the same, differing only in that whether the channel information representing explored areas includes privilege information. As a result, the disparity of observation feature mapping and privilege feature mapping captures the difference between explored area and the whole area, as well as the structure information. After that, the Feature Utilization network extracts the feature of the disparity information as the state value. Specifically, the differential calculation process described above can be represented as:

$$F_c^{(t)} = CNN\big(O_c^{(t)}\big), \hat{F}_c^{(t)} = CNN\big(\hat{O}_c^{(t)}\big) \tag{1}$$

$$\Delta F_c^{(t)} = Flatten\big(F_c^{(t)} - \hat{F}_c^{(t)}\big) \tag{2}$$

Where $\Delta F_c^{(t)}$ is the vectorized disparity of feature mappings. It is worth noting that in the above equation, the CNN network shares parameters. By the processing of the network structure in the terms of differences, the feature encoding network can be enhanced, thereby encoding key structural information of the environment into the feature mappings.

B. Mutual Information Evaluation

To quantify the information gain during exploration and measure the uncertainty reduction, we design a mutual information evaluation network.

In environment exploration tasks, the mutual information between observation and privilege information measures the amount of information shared between them. Given the privilege information $\hat{O}_c^{(t)} \in \{0,1\}^{X\times Y}$ and local observation information $O_c^{(t)} \in \{0,1\}^{X\times Y}$ as two channels, they are input into a shared feature extraction network. The expected information gain of a candidate control action a_i can be measured by the mutual information between the observation features $F_{N\times D}^{(t)}$ and the privilege features $\hat{F}_{N\times D}^{(t)}$, where N is the number of samples and D is the dimension.

Given a batch of observation, our proposed loss function is computed over two sets of paired embeddings. For each local observation feature map, the positive samples $\{F_i^{(t)}, \hat{F}_i^{(t)}\}$ consist of local observations and their corresponding privilege information, while the negative samples $\{F_i^{(t)}, \hat{F}_j^{(t)}\}$ consist of local observations paired with other mismatched privilege information. We assume that the joint distribution $\{F^{(t)}, \hat{F}^{(t)}\}$ of each sample can be estimated by comparing the similarity of positive and negative samples with the joint distribution of the noise contrast samples. The loss function is defined as:

$$L_{MI} = -\mathbb{E}\left[\log \frac{p_{pos}\{F_i^{(t)}, \hat{F}_i^{(t)}\}}{p_{pos}\{F_i^{(t)}, \hat{F}_i^{(t)}\} + \sum_{j\neq i} p_{neg}\{F_i^{(t)}, \hat{F}_j^{(t)}\}}\right] \tag{3}$$

Where $p_{pos}\{F_i^{(t)}, \hat{F}_i^{(t)}\}$ is the joint probability of positive sample pairs; $p_{neg}\{F_i^{(t)}, \hat{F}_j^{(t)}\}$ is the joint probability of negative sample pairs. Under the conditional independence hypothesis, the joint distribution and edge distribution are obtained from the samples.

From an information theory perspective, given the current observation, privilege information, and past information feature map, the mutual information-driven map building and exploration method aims to maximize the mutual information between observation features and privilege features:

$$\begin{aligned} I(F^{(t)}; \hat{F}^{(t)}) = &\frac{1}{\|\mathcal{P}\|} \sum_{(z_i,z_j)\in\mathcal{P}} \log(1 + e^{-T_\phi(z_i,z_j)}) \\ &-\frac{1}{\|\mathcal{N}\|} \sum_{(z_i,z_j)\in\mathcal{N}} \log(1 + e^{T_\phi(z_i,z_j)}) \end{aligned} \tag{4}$$

Where $\mathcal{P}$ represents positive pairs of samples, $\mathcal{N}$ represents negative pairs of samples, $T_\phi(x, y) : \mathbb{R} \to \mathbb{R}$ Sampling from the examples is numerically more stable, especially when using the cross-entropy loss function. The distance between embeddings (z_i, z_j) is the normalized L_2 distance d_{ij}. Cru-

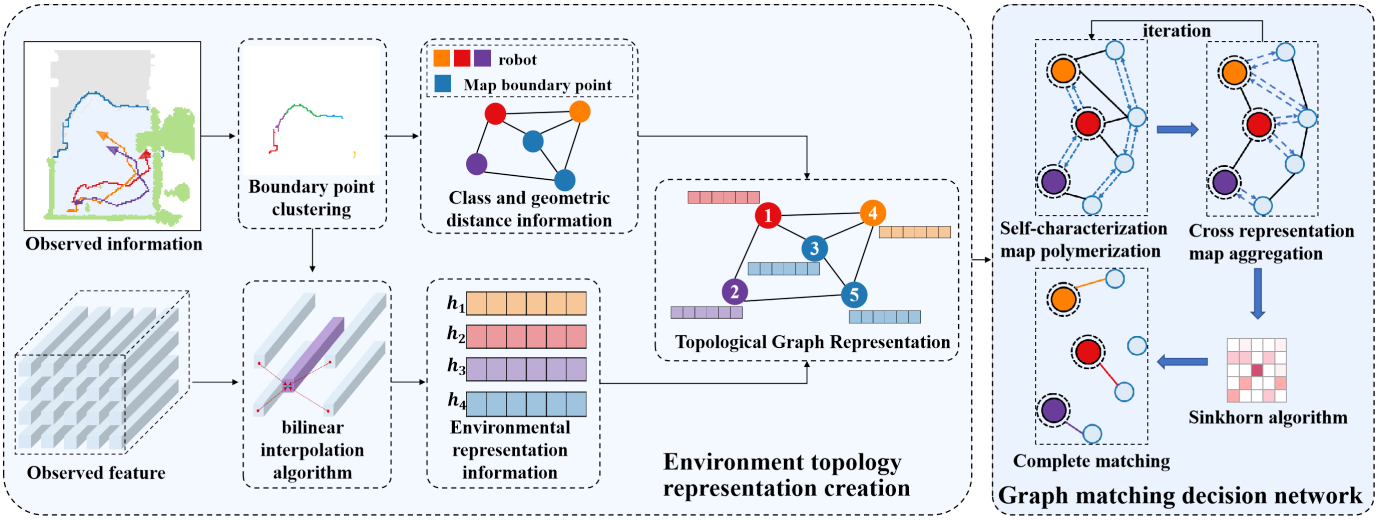

Fig. 5. Multi-agent decision-making network based on topological graph matching. This framework concatenates internal and external information fusion of the graph, completing graph matching between the representation of robots and boundary points, and assigning corresponding boundary points as long-term target points for each robot.

cially, defining T_ϕ as a function of d_{ij} can relate the distance in the embedded space to mutual information. By comparing the similarity of these positive and negative sample pairs, the loss can effectively increase the distance between positive sample pairs and negative sample pairs, thereby maximizing the mutual information.

Fig. 4 provides an overview of mutual information evaluation process. Along with an embedding network for learning feature vectors, a statistical network is also trained to capture statistics on distances between vectors. To estimate the distance between them T_ϕ provides a training signal for the embedded network. This procedure requires no prior assumptions about the distance distribution, allowing us to learn variational functions to separate arbitrarily complex distributions.

The statistical network T_ϕ is designed to ensure that positive samples are ranked higher than negative samples for any given query. During gradient updates, the network should move positive sample pairs closer together while pushing negative sample pairs further apart. This process helps to effectively distinguish between positive and negative samples within the embedding space. Let $p(d^+|\theta_t)$ and $p(d^-|\theta_t)$ represent the conditional density functions corresponding to the distances of positive pairs d^+ and negative pairs d^- respectively, based on the embedding network parameters θ_t at timestep t. The feature embedding loss gradient functions are expressed as follows:

$$sgn(\nabla_\theta I(F_i^{(t)}, F_j^{(t)})) = -sgn(\nabla_\theta d^-) \tag{5}$$

$$sgn(\nabla_\theta I(F_i^{(t)}, \hat{F}_i^{(t)})) = sgn(\nabla_\theta d^+) \tag{6}$$

During the training process, the positive distance is minimized and the negative distance is maximized in a given time step. Intuitively, when formula holds, mutual information and distance d are not only related to each other, but also have a monotone relationship.

We use the mutual information between the privilege feature mapping and the observation feature mapping to evaluate the task completion. In this way, we ensure that the feature encoding network correctly captures the structured feature information reflected by the difference between the privilege feature and the observation feature.

C. Multi-Agent Decision-making Network

To maximize the utilization of environmental structure information and enable multirobot systems to make rational decisions, a topology-based graph matching decision network is designed. First, feature vectors corresponding to robot positions and boundary points in the observation feature map are extracted and combined with geometric distance features from the environment to construct a representation of the topological graph. Then, a graph neural network framework is adopted to perform graph matching, using the design from [42] as the network backbone. The overall framework process is illustrated in Fig. 5.

1) Point Feature Extraction: After obtaining feature mapping from the Asymmetric Feature Representation module, which encodes the structure information of the environment, a point feature extraction method based on nearest neighbor clustering and bilinear interpolation algorithm was adopted. The method first clusters the boundary points of the explored area, using the centroid of each cluster as the representative of corresponding boundary point cluster, and also records the number of points in each cluster as one of the inputs for decision-making. Additionally, the maximum distance between two boundary points in the same cluster cannot exceed a threshold distance r_{clus}. After clustering is completed at time t, the boundary point cluster i can be represented as $F_i^{(t)} = \{f_k^{(t)}\}_{k=1:n_i}$ where $f_k^{(t)} \in \mathbb{R}^2$ represents the two-dimension coordinates of boundary point K, and n_i indicates the number of boundary points in cluster i. And the centroid of boundary point cluster $F_i^{(t)}$ can be represented as f_k^c. The specific clustering algorithm and computational process are illustrated in the pseudocode of Algorithm 1.

Extracting the structure features of key locations such as robots and boundary points is essential for the decision-making network. In our method, bilinear interpolation is applied to the feature map to extract the relevant feature vectors corresponding to the original grid coordinates in the observation feature map. Given an input observation map with the size of $X \times Y \times 5$ and a feature map with the size of $X_l \times Y_l \times C_h$, for a point p_i with coordinates (x, y) in the input observation map, its projected coordinates in the feature map space should be $p_i' = (x', y')$, where $x' = x \cdot \frac{x_l}{x}, y' = y \cdot \frac{y_l}{y}$. The value corresponding to point (x', y') in the feature map $F_C^{(t)} \in \mathbb{R}^{X_l \times Y_l \times C_h}$ can be obtained using bilinear interpolation along the x-axis and y-axis directions. The bilinear interpolation process described above is denoted as $I_i^f = Interp(p_i, F_c^{(t)})$ in this paper.

2) Topological Graph Representation: After point feature extraction, the feature vectors need to be combined with geometric distance information from the environment to construct self-representation graphs $G_r = V_r, E_r$ and $G_f = V_f, E_f$, which only contain information about robots or boundary points, as well as a cross-representation graph $G_r = V_r, V_f, E_{rf}$, which contains both robot and boundary point information. For a node i in the self-representation or cross-representation graph, its initial node feature vector $v_i \in \mathbb{R}^{5+C_h}$ is composed of three parts: category information $v_i^{cla} \in \{0, 1\}^2$, geometric information $v_i^{geo} \in \mathbb{R}^3$, and environmental representation information $v_i^{cla} \in \mathbb{R}^{C_h}$. The category information v_i^{cla} is a one-hot encoded label representing two categories: whether the node is a robot node or a boundary point. The first two dimensions of geometric information v_i^{geo} represent position $p_i = (x, y)$ of the node, while the last dimension represents the geometric information of the node. Additionally, the environmental representation information is obtained by interpolating the node position information in the observation feature map: $v_i^{rep} = Interp(p_i, F_c^{(t)})$.

The environment is partially observable, and utilizing historical observation information helps robots avoid redundant exploration. The self-representation graph $G_r^h = \{V_r^h, E_r^h\}$ is constructed to represent the historical trajectory information of robots, and $G_g^h = \{V_g^h, E_g^h\}$ is constructed to represent historical boundary point information. To establish connections between current robot and boundary point information and historical information, we construct the cross-representation graph $G_{r_r}^h = \{V_r^h, V_r, E_{r_r}^h\}$, as well as the cross-representation graph $G_{f_g}^h = \{V_g^h, V_f, E_{f_g}^h\}$ between current boundary points and historical target points. By constructing these topological graphs, both structural feature information and geometric distance information in the environment can be adequately represented, laying the foundation for efficient decision-making by subsequent graph matching decision networks.

3) Graph Matching Decision Network: Graph matching decision network utilizes a graph attention mechanism to sequentially aggregate and extract features from the self-representation graph and the cross-representation graph, updating the features of corresponding edges and nodes in the topological graph. In the cross-representation graph G_{rf} after feature updates, the feature value of each edge in the edge set E_{rf} represents the matching degree of each robot to the boundary point node. Therefore, by extracting the feature values from the updated edge E_{rf} set and using the Sinkhorn algorithm for linear assignment computation, the graph matching can be completed, assigning long-term target points to each robot.

Algorithm 1 Adjacent Neighbor Clustering of Boundary Points

Input: A set of boundary points $F = \{f_k\}_{k=1:n}$ containing n boundary point.

Output: Boundary point cluster and corresponding cluster center point set $F_{cluster} = \{F_i, f_i^c\}_{i=1:n^c}$,where $F_i = \{F_i, f_i^c\}_i$.

```
Initializes the clustering of boundary points F_cluster;
while not F = ∅ do
    Initialize a cluster, called F_i;
    Take a boundary point from the boundary point set F and add it to the cluster F_i
    for p in set of boundary points do
        if f is any neighboring boundary point of a cluster, and the distance to any boundary point within cluster F_i is less than r_clus. then
            Remove f from the set F and add it to the cluster F_i.
        end if
    end for
    Compute the average distance between each boundary point in the cluster F_i and the remaining boundary points in the cluster.
    Select the boundary point with the smallest average distance as the cluster center point f_i^c.
    Add the cluster F_i and the cluster center point f_i^c to the set of boundary point clusters F_cluster.
end while
```

We used an encoder based on a multi-layer perceptron network to encode the category information and geometric information $[V_i^{cla}, V_i^{geo}]$ of each node i in the graph, obtaining a feature vector of length C_h. This feature vector is concatenated with the environmental representation information V_i^{rep} to form the node feature $V_i^0 \in \mathbb{R}^{2C_h}$ for subsequent feature aggregation. The core idea of the graph attention network is to utilize an attention mechanism to aggregate features between neighboring nodes in the topological graph. Therefore, for the nodes $V_i^l \in \mathbb{R}^{h^l}$ in the $l-th$ layer of the graph network, trainable weight parameters $W_k^l \in \mathbb{R}^{h_k^l \times h_l}$,$W_q^l \in \mathbb{R}^{h_q^l \times h_l}$,and $W_u^l \in \mathbb{R}^{h_u^l \times h_l}$are introduced to generate the key $k_i^l \in \mathbb{R}^{h_k^l}$, query $q_i^l \in \mathbb{R}^{h_q^l}$, and value $u_i^l \in \mathbb{R}^{h_u^l}$ in the attention mechanism:

$$k_i^l = W_k^l \cdot V_i^l, q_i^l = W_q^l \cdot V_i^l, u_i^l = W_u^l \cdot V_i^l \tag{7}$$

The attention coefficient a_{ij}^l between node i and its neighboring node $j \in N_i$ can be calculated by the following equation:

$$a_{ij}^{l} = \frac{exp(k_j^{l^T} \cdot q_i^l)}{\sum_{m \in N_i} exp(k_m^{l^T} \cdot q_i^l)} \tag{8}$$

Additionally, the attention coefficient $a_i j^l$ will also serve as the edge feature value between node i and its neighboring node j in the topological graph. Therefore, for node i, the aggregation of neighboring node feature values $V_{N_i}^l \in \mathbb{R}^{h_u^l}$ can be expressed as:

$$V_{N_i}^l = \sum_{m \in N_i} a_{im}^l \cdot u_m^l \tag{9}$$

The final feature value of node i will be updated as the aggregation of neighboring node features and the fusion with its own node feature:

$$V_i^{l+1} = v_i^l + \rho([V_i^l || V_{N_i}^l]) \tag{10}$$

In the above equation, $\rho(\cdot)$ represents the feature fusion function implemented using a multilayer perceptron, and $[\cdot||\cdot]$ denotes the concatenation of two feature values. The feature of each node in the self-representation graph will be updated to the aggregation of its own feature and the features of its neighboring nodes. This operation will be applied to the self-representation graphs G_r, G_f, G_r^h, and G_g^h in this method. After completing the feature updates in the self-representation graphs, the node features will serve as the initial features for the corresponding nodes in the cross-representation graph, which will then be input into the subsequent graph attention network. Unlike that in the self-representation graphs, the feature extraction process in the cross-representation graph uses a non-linear mapping method to generate attention coefficients. Additionally, the distance d_{ij} calculated by the fast marching algorithm is incorporated as an input to the non-linear mapping function $\varphi(\cdot)$:

$$a_{ij}^{l} = \frac{exp(\varphi([k_j^l || q_i^l || d_{ij}]))}{\sum_{m \in N_i} exp(\varphi([k_m^i || q_i^l || d_{im}]))} \tag{11}$$

The notation $[\cdot|| \cdot ||\cdot]$ denotes the concatenation of three vectors. Additionally, the non-linear mapping function $\varphi(\cdot)$ is implemented using a multi-layer perceptron, which outputs a one-dimensional real number. In this study, the cross-representation graphs $G_{r\_r}^h$, $G_{f\_g}^h$, and G_{rf} will undergo sequential feature extraction using graph attention networks. The edge features in the cross-representation graph G_{rf} will be extracted and used as the affinity matrix in graph matching, denoted as $A_M \in \mathbb{R}^{n_r \times n_f}$. Here, n_r and n_f represent the numbers of robots and boundary points, respectively. Each element in the matrix represents the degree of matching between the corresponding robot and boundary point. Ultimately, we employ the Sinkhorn algorithm to iteratively normalize the rows and columns of the affinity matrix alternately, gradually transforming it into a probability matrix to accomplish graph matching. Each robot will select the boundary point with the highest probability value from the probability matrix as its long-term target point.

D. Asymmetric Actor-Critic Training Framework

The asymmetric actor-critic training framework combines the mutual information from the exploration process with the value loss function in reinforcement learning. This ensures that the robot follows the optimal exploration path under the guidance of mutual information. The asymmetric actors-critic in training framework is mainly built on PPO training framework and incorporates the supervised learning loss function training method. Specifically, during training, multiple agents will interact with the environment based on policies and collect data such as status, actions, and rewards into a data pool. After completing an interaction cycle, the algorithm extracts a portion of data from the data pool for training. In the training process, the loss function of PPO algorithm can be expressed as:

$$L^{PPO} = L^{CLIP}(\theta) - c_1 L^{VF}(\vartheta) + c_2 S[\pi_\theta] \tag{12}$$

Where $L^{CLIP}(\theta)$ represents the policy loss function, $L^{VF}(\vartheta)$ represents the value loss function, and $S[\pi_\theta]$ represents the entropy of the policy π_θ.

On this basis, we introduce the mutual information evaluation loss function L_ϕ as an additional loss function to supervise the training of the structured feature extraction network. The specific calculation method of mutual information is described above. The final loss function during training can be expressed as the weighted summation of the mutual information evaluation loss function and the PPO loss function:

$$L = L^{CLIP}(\theta) - c_1 L^{VF}(\vartheta) + c_2 S[\pi_\vartheta] - c_3 L_\phi \tag{13}$$

Where c_1, c_2, c_3 are weighting coefficients that balance the contribution of each term during training. These weights can be tuned depending on the specific task and the importance of each loss component.

After obtaining the loss function L, we train the network based on this loss function through gradient ascent. By incorporating the mutual information loss L_ϕ, the model is guided to extract structured and informative features, potentially improving its ability to generalize and perform complex tasks. This synergistic combination of mutual information and PPO loss helps strike a balance between learning an effective policy and retaining useful feature representations for decision-making.

We adopt the training paradigm of "centralized training, centralized execution", where multiple agents can be regarded as a centralized single agent with a multidimensional action space. In the reinforcement learning design, the observation space for the agents is a 5-channel map $O \in \{0,1\}^{X \times Y \times 5}$. The action space for multiple agents is denoted as $A \in \{0,1\}^{X \times Y \times n_r}$, where action of each agent involves selecting n_r grid points as long-term targets from a grid map of size $X \times Y$ and executing them in the next planning cycle. To address this problem, we employ an improved asymmetric actor-critic training framework based on the PPO algorithm. For reward design, this study adopts the concept of information theory combined with mutual information, using the reduction in uncertainty about the environment, achieved by multiple robots during the exploration process, as the task evaluation metric. Additionally, multiple robots will receive

a fixed amount of penalty at each time step to encourage them to improve exploration efficiency. If the area explored by multiple robots at decision time t is denoted as $A_e^{(t)}$, where $A_e^{(t)} = 0$, then the reward $r^{(t)}$ that robots can receive at time step $t(t > 0)$ can be represented as:

$$r^{(t)} = a_1(A_e^{(t)} - A_e^{(t-1)}) - a_2 \tag{14}$$

Where a_1 and a_2 are adjustable reward coefficients. In the reinforcement learning process the time required to execute one low-level action is referred to as a time step. If multiple agents complete full exploration of the environment within this time frame, the episode ends; otherwise, it ends when the maximum time step is reached.

V. Experiments

A. Experimental Setup

1) Experimental Environment: To validate the proposed framework, experiments were conducted using the iGibson physics simulation engine. iGibson is a virtual environment tool for robotics and AI research, providing realistic indoor scenes for the development and testing of robot perception, navigation, and task planning. The iGibson simulation engine supports various map scene datasets and realistic physics-based interactions between robots and environments. In these experiments, TurtleBot robots equipped with depth cameras were used within the iGibson simulation engine to closely simulate real-world scenarios. The TurtleBot robots can move using a differential drive method and perceive the environment through depth cameras, with realistic collision interactions with the environment.

For the experiments, publicly available Gibson and MatterPort3D datasets were used for training and testing, respectively. The Gibson dataset offers large-scale 3D data of real indoor environments, while the MatterPort3D dataset provides a larger scale and more diverse set of indoor scenes. Nine scenes from the Gibson dataset were selected for training, and the trained model was then tested on the MatterPort3D dataset. Some scenes with small areas or disconnected regions that were impassable for the TurtleBot robots were excluded, resulting in 51 scenes for performance testing. These scenes were further divided into three subsets based on their area sizes: large, moderate, and small area scenes. During testing, each scene was evaluated over 100 trials, and the average results were recorded. The initial positions and orientations of the robots were randomly generated within the scene, with multiple robots initially concentrated in a small area.

2) Parameter Settings and Training Details: In this study, the grid map size was set to 480×480, where each grid cell represents an area of 0.01 square meters in the real world. The maximum field of view radius for the robot was set to 3 meters, and the maximum robot movement speed was set to 1 meter per second. During training, the maximum time steps per episode were set to 1800, and the planning horizon for long-term goal planning was set to 15 time steps. The coefficients α and β in the reward function were set to 0.005 and 0.225, respectively. The exploration rate prediction error ratio coefficients γ and δ were set to 0.05 and 0.01, respectively. The structure feature encoding network consisted of a 5-layer convolutional neural network, with the output observation feature mapping channel set to 32. The feature utilization network was a 3-layer multilayer perceptron network. Additionally, the multilayer perceptron network for encoding node categories and geometric information also had an output layer size of 32. The mutual information evaluation network first encoded the input feature information using a 4-layer convolutional neural network, followed by a 3-layer multilayer perceptron for feature output, where the output layer size of the perceptron was 64.

TABLE I
Training hyperparameter settings.

Hyperparameter Name	Value
Training rounds	1800
Learning rate	1×10^{-5}
Incentive discount factor	0.99
GAE discount factor	0.95
Value loss function coefficient c_1	3.0
Strategy entropy coefficient c_2	1.0
MI evaluation loss function coefficient c_3	1.0

Furthermore, in the graph matching decision network based on graph attention network, the vector lengths corresponding to the keys, values, and queries in the attention mechanism were set to 32. The code framework used in this study was the widely used PyTorch framework in academia. The asymmetric actor-critic training framework was an improvement based on the Proximal Policy Optimization (PPO) algorithm, with training hyperparameters set as shown in Table I. The above training parameters were determined through experimental comparisons to obtain the optimal values. The code was deployed and trained on a workstation equipped with an Intel i9-13900k central processing unit and an NVIDIA GeForce RTX 4090 graphics card, with the complete training process taking approximately 72 hours.

3) Evaluation Metrics: For multirobot active mapping tasks in indoor environments, task completion effectiveness is evaluated based on time efficiency and mapping completeness. Time efficiency refers to the time required for robots to complete the exploration task, while mapping completeness reflects the exploration speed within a given time frame. Therefore, time steps and exploration rate are used as evaluation metrics. Time steps indicate the time required for the robots to finish the exploration, while the exploration rate is the ratio of the area explored by the robots to the total explorable area of the environment within the maximum episode length.

B. Comparison Experiment

1) Baseline Methods: To thoroughly validate the effectiveness of the proposed method, several high-performance baseline methods were introduced for comparison. These include four traditional planning methods (Utility [7], mTSP [22], Voronoi [14], CoScan [5]) and two reinforcement learning-based methods (Ans-Merge [40], NCM [42]). To ensure fair

TABLE II
PERFORMANCE COMPARISON IN MATTERPORT3D TEST DATASET.

Methods	Small Scene ($< 60m^2$)		Medium Scene ($60 - 100m^2$)		Large Scene ($> 100m^2$)	
	Time (step)	Explo (%)	Time (step)	Explo (%)	Time (step)	Explo (%)
Utility[7]	1111.84	97.28	1779.78	95.41	3056.00	83.84
mTSP[22]	893.26	**97.85**	1120.22	96.76	1764.36	95.93
Voronoi[14]	904.53	97.68	1226.72	96.72	1520.07	**96.19**
CoScan[5]	716.63	98.09	1070.11	96.72	1601.43	96.07
Ans-Merge[40]	1529.58	96.08	2425.67	85.98	3827.21	81.19
NCM[42]	690.16	97.78	987.44	96.74	1492.07	96.09
AIM-Mapping	**542.35**	97.67	**803.37**	**96.76**	**1341.59**	96.03

comparison, for the aforementioned baseline methods, only their top-level decision modules, which allocate long-term target points to robots, were utilized. The bottom-level action execution modules for all methods were uniformly processed using a fast traversal algorithm, and the planning horizons for long-term goals were kept the same for all methods. The following provides detailed introductions to the aforementioned baseline methods:

Utility introduces the concept of information gain, where each robot selects the boundary point with the maximum information gain as the long-term target point. The information gain of a boundary point is defined as the area of unexplored regions within a circle centered at that boundary point with the perception distance limit as the radius.

mTSP transforms the multirobot active mapping problem into a multiple Traveling Salesman Problem, which requires multiple robots to cooperatively traverse all boundary point nodes starting from their current node positions. This is achieved by establishing a boundary point-robot passable topological graph containing distance information.

Voronoi segments the entire map using the Voronoi partitioning method, with the robot location as seed points. Each resulting map sub-block ensures that any point within it is closer to its corresponding seed point than to any other seed points. Each robot then selects the nearest boundary point within its map sub-block as the long-term target point.

CoScan first performs K-means clustering on all boundary points and models the multirobot active mapping task as an Optimal Mass Transport Problem, allocating boundary point clusters based on distances between robots and boundary point clusters.

Ans-Merge extends the ANS [40] method, which is a reinforcement learning-based algorithm for single-robot exploration in unknown environments. It overlays the local grid map centered on itself and the global grid map as decision inputs and selects long-term target points for robots through regression.

NCM builds a topological graph between boundary points and robots based solely on geometric distance information and introduces a multi-graph neural network to predict the neural distance between boundary points and robots. It then matches boundary points with robots based on neural distance and assigns long-term target points to each robot.

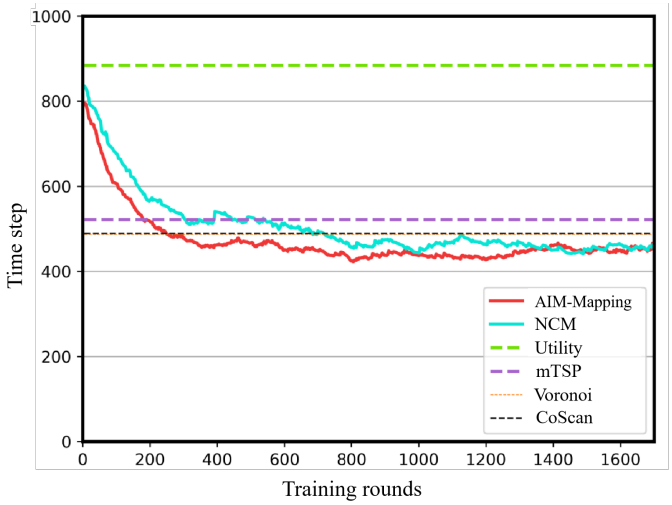

Fig. 6. Training performance comparison. The training results of the proposed AIM-Mapping and the baseline method NCM, as well as the comparison of the performance of the planning-based baseline methods (Utility, mTSP, Voronoi, and CoScan) on the training set.

2) Performance Comparison Experiment: We recorded the number of time steps required for multiple robots to completely explore the environment during training as the training result. From the Fig. 6, it can be observed that with the increase in training epochs, the number of time steps required for the AIM-Mapping algorithm and the NCM algorithm to complete the exploration task gradually decreases and eventually outperforms traditional planning methods. Among them, the AIM-Mapping algorithm slightly outperforms the NCM algorithm in terms of convergence speed and performance after convergence. The Voronoi algorithm and the CoScan algorithm perform very similarly on the training set. The average step lengths for completing the exploration task are 486.55 and 488.89, respectively, so the corresponding two dashed lines in Fig. 7 are very close.

To further validate the effectiveness of the proposed method, we tested the trained models on the test set and compared them with baseline methods. During testing, the maximum episode length was set to 5000, and other settings remained the same as those during training. The comparative experimental results are shown in Table II. From the table, it can be observed

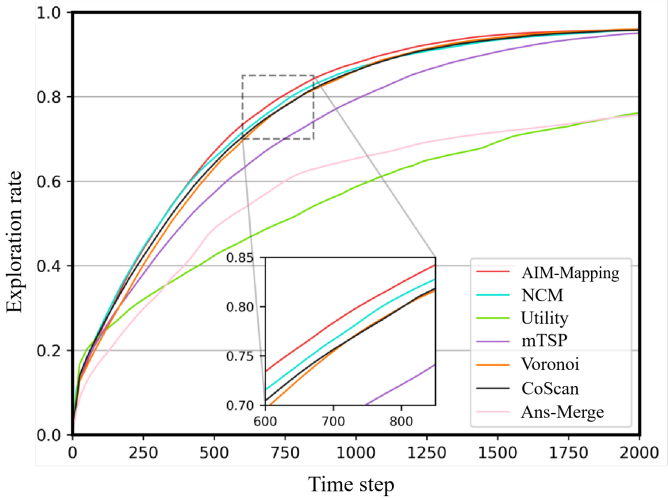

Fig. 7. Illustration of average exploration rate variation during test episodes.

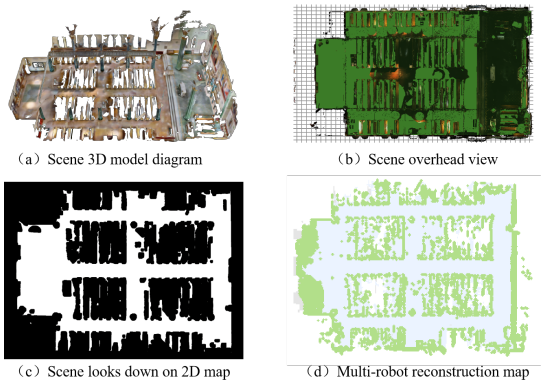

Fig. 8. Visualization and map reconstruction effect of the scene with the ID 'gYvKGZ5eRqb'

that except for the Ans-Merge and Utility methods, all other methods achieve an exploration rate of over 95% in scenes of various area scales, indicating successful completion of full environment exploration. The Ans-Merge method, using the original grid map as input under the scenes and reward settings of this study, exhibited instability during training, resulting in poor model performance. The Utility method only considers information gain and ignores distance information, leading to significant path redundancy and difficulty achieving high exploration rates in large-scale scenes. However, in terms of time efficiency, the proposed AIM-Mapping outperforms various baseline methods by achieving relatively optimal efficiency at the same exploration rate. In moderate-sized and large-scale scenes, AIM-Mapping reduces the number of time steps required for mapping compared to the best-performing baseline method NCM by approximately 10%. The performance improvement is due to the proposed AIM-Mapping, which not only extracts distance information and structural information from the environment to establish a topological representation but also considers the information value acquired during the map exploration process using mutual information. This results in more effective long-term goal planning, thereby improving the time efficiency of task completion.

C. Exploration Visualization Experiments

To visually demonstrate the effectiveness of the algorithm, we visualized the testing process in the simulation environment. As shown in Fig. 8, the scenario labeled "gYvKGZ5eRqb" from the test set was used for visualization. Based on the 3D model diagram, the scenario is identified as an indoor auditorium. In this scenario, three robots were deployed using the proposed AIM-Mapping algorithm to reconstruct the map, with the reconstruction result shown in Fig. 8(d). In Fig. 8(d), the green areas represent obstacles, while the light blue areas represent the explored and navigable regions. Since the dataset simulates a real-world scenario, where the ground may be uneven (e.g., protrusions or depressions), there is a slight difference between the left side of the map in Fig. 8(d) and Fig. 8(c). For areas that are inaccessible, the robots mark them as obstacle zones. In Fig. 9, the paper presents the first-person RGB-D observations of the three robots at different time steps, their movements in a third-person top-down view, and the global map reconstructed by the multiple robots. In the global map, green areas indicate obstacles, light blue areas indicate explored navigable regions, gray areas represent the true values of navigable areas in the map, and dark blue points denote boundary points. The elements related to the robots are distinguished using different colors: the three robots are represented by red, yellow, and purple, respectively. The arrows in corresponding colors indicate the current positions of the robots, and the curves connected to the arrows show the robots historical trajectories. In the boundary regions between known and unknown maps, dots in corresponding colors represent the long-term goal points assigned to each robot. It can be seen that although the robots were initialized in a small area, they quickly moved in different directions after the exploration began. During the exploration process, their trajectories seldom overlapped, indicating that the area of repeated exploration was minimal and that the long-term decision-making of the robots was efficient.

To provide a clear comparison with various baseline methods, this paper tested each baseline method and recorded the mapping progress and robot trajectories at different time steps during the testing process. Taking the scenario "JmbYfDe2QKZ" as an example, the initial positions and orientations of the robots were the same in each test round, and the visualization results are shown in Fig. 10. The meanings of the colored elements in Fig. 10 are the same as those in Fig. 9. During the visualization process, we also recorded the time taken by each method to complete the exploration. It can be seen that the AIM-Mapping method achieved better exploration efficiency, completing the overall exploration of the environment in a relatively short time. Additionally, except for Ans-Merge, most of the methods successfully completed the full exploration of the unknown environment and reconstructed a top-down map. This is because Ans-Merge selects long-term goal points using a regression-based approach, choosing a point anywhere on the map as the long-term goal point, without ensuring that the selected point is a boundary point.

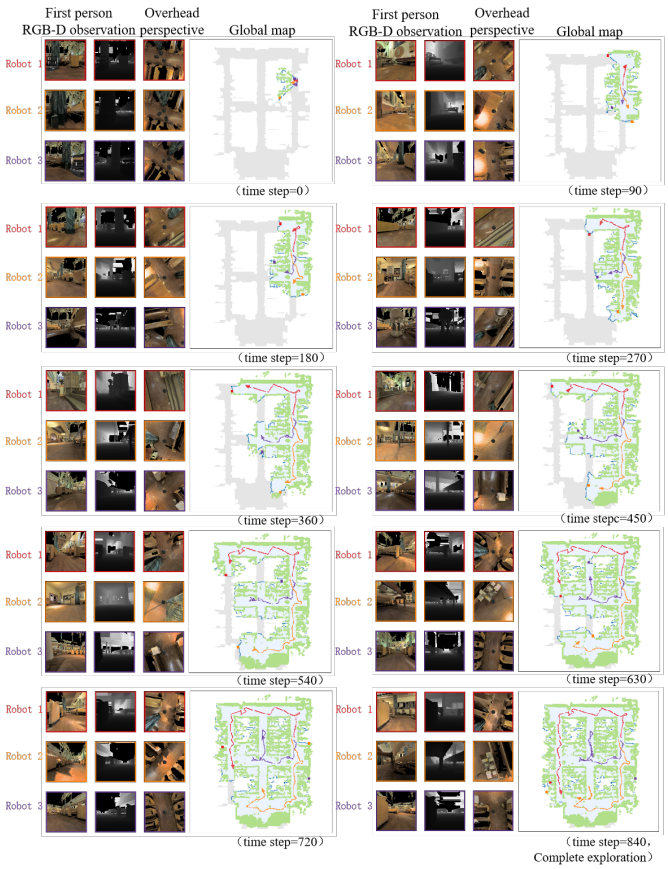

Fig. 9. Visualization schematic diagram of robot mapping process.

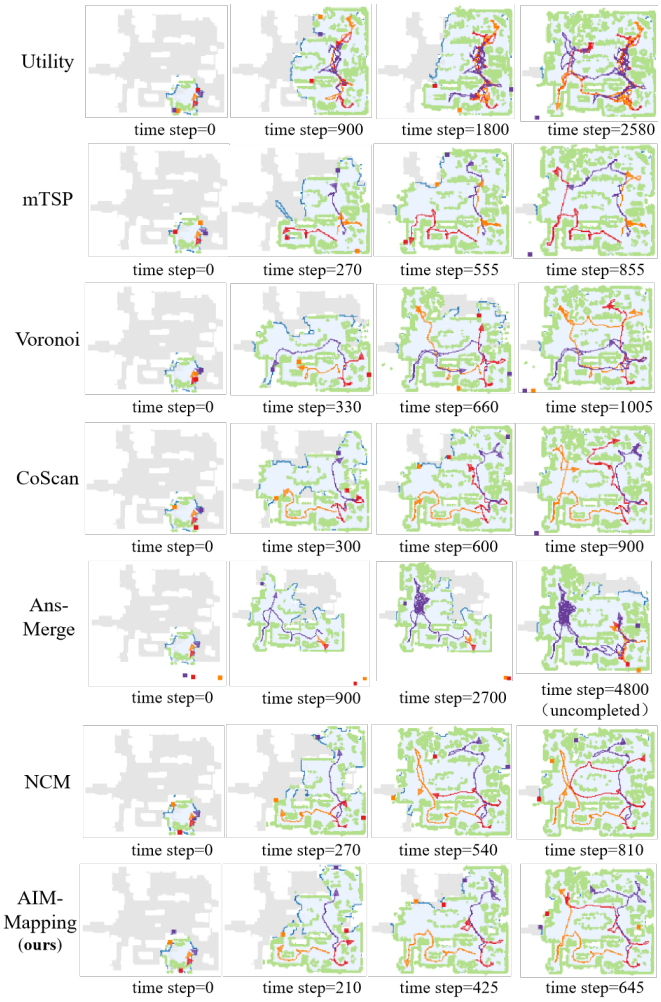

Fig. 10. Visualization comparative schematic diagram of mapping process.

As a result, Ans-Merge cannot guarantee complete exploration of the environment. Furthermore, under the Utility method, the trajectories of multiple robots show significant overlap, and individual robots exhibit repeated movements, which is consistent with the analysis in Table II. Finally, among the various methods, the AIM-Mapping method shows less overlap in robot trajectories, and the movements of the robots are relatively smoother. This suggests that the long-term goal point selection of the AIM-Mapping method is more reasonable and efficient to some extent.

D. Ablation and Generalization Experiment

1) Generalization Experiment: To verify the generalization ability of the proposed method with different numbers of robots, the models trained with 3 robots were extended to settings with 4 and 5 robots for testing. The average test results on the entire test set are shown in Table III. From the data in the table, it can be observed that despite the change in the number of robots during testing, the AIM-Mapping method proposed in this paper still achieves relatively superior time efficiency compared to the baseline methods. The strong generalization ability of AIM-Mapping is partly attributed to its construction of the topological representation graph, where topological relationships and distance information are minimally impacted by changes in the number of robots. This indicates that the AIM-Mapping model trained in only 9 scenes also has certain performance limitations in terms of generalization to the number of robots.

2) Ablation Experiments: To further validate the effectiveness of each module proposed in the AIM-Mapping method, we present the results of ablation experiments on different modules. Ablated privileged representation: The privileged information introduced in the asymmetric feature representation module is removed, and only current and historical observations are used to evaluate the state value. Ablated mutual information evaluation: The mutual information evaluation module, which assesses environmental uncertainty using privileged information, is removed, and the robots are guided using the explored area for exploration. The comparison of training results after removing the respective modules is shown in Fig.11, while the comparison test results of the trained models on the test set are presented in Table IV.

Experimental results show that removing privileged observational information reduces time efficiency of the algorithm, as reflected in the training curves and the tests conducted on three scenarios of different sizes. This, to some extent, indicates that utilizing privileged global observations in the network evaluation module during training not only helps the feature encoding module capture accurate and valuable feature

TABLE III
COMPARATIVE EXPERIMENTAL PERFORMANCE ON GENERALIZATION OF ROBOT QUANTITIES.

Methods	Number of robots=3		Number of robots=4		Number of robots=5	
	Time (step)	Explo (%)	Time (step)	Explo (%)	Time (step)	Explo (%)
Utility[7]	1881.27	92.93	1681.73	93.69	1589.45	94.90
mTSP[22]	1212.49	96.84	1038.67	97.13	873.65	96.95
Voronoi[14]	1187.22	96.83	999.14	97.17	869.63	96.95
CoScan[5]	1084.27	**97.05**	988.78	**97.18**	835.63	97.00
Ans-Merge[40]	2476.57	88.43	1681.73	93.69	1536.73	95.00
NCM[42]	1015.22	96.87	868.67	97.01	761.78	97.00
AIM-Mapping	**763.14**	97.02	**747.74**	97.03	**734.66**	**97.00**

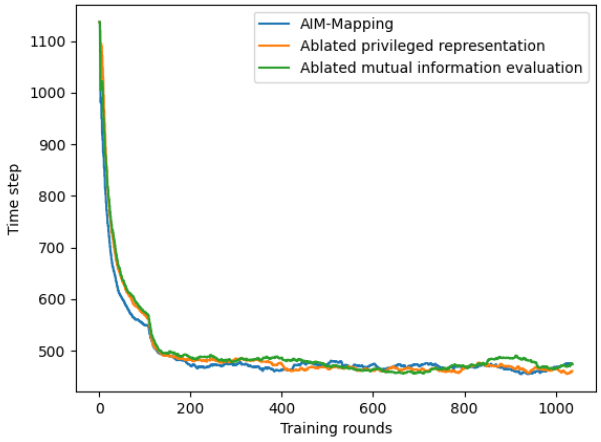

Fig. 11. A comparison of the reduction in exploration timesteps between the complete method and the ablated method as the number of training epochs increases.

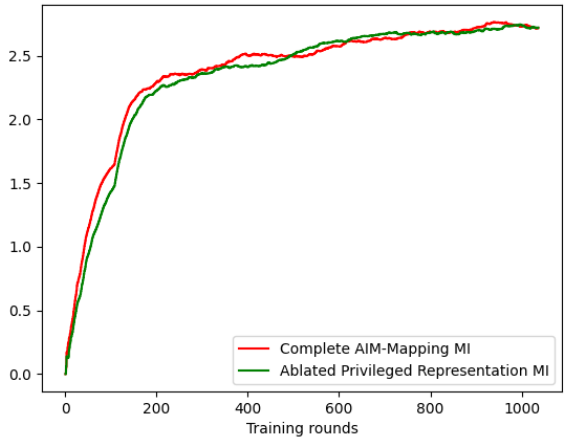

Fig. 12. A comparison of the increase in mutual information between the complete method and the method with ablated privileged representation as the number of training epochs increases.

information but also allows for a more accurate assessment of the robots state and action values, thereby improving decision-making efficiency. From the training curves, it can be seen that in the ablation experiments for both types of privileged information representations, removing privileged information slows down the early convergence of the training curves, and the final convergence position is not as optimal as the complete AIM-Mapping method. Experimental results on the test set also indicate that not using privileged information reduces the robots decision-making efficiency.Thus, using global information as privileged input, extracting feature mappings through asymmetric feature representation, and evaluating map uncertainty during exploration with the mutual information evaluation method can improve the overall decision-making performance of the system.

In addition, this paper also recorded the change in the average value of mutual information evaluation over the training time steps, as shown in Fig.12. It can be observed that as the training time steps increase, the mutual information value gradually rises. After the ablation of privileged information, the growth of mutual information is slower compared to the complete method, which indirectly confirms the effectiveness of the privileged information approach. In the early stages of training, the mutual information value increases as the average step length decreases, and in the mid-to-late stages of training, it continues to rise as the exploration actions are optimized. This indicates that under the guidance of the mutual information evaluation network, the robots are progressively finding the optimal exploration paths. The mutual information evaluation network utilizes privileged information to calculate the reduction in environmental uncertainty during exploration, moving beyond simple area-based assessments. This allows the robots to evaluate the information gained during exploration more accurately. By combining information theory with spatial prediction, the network guides the robots to prioritize exploring areas with high information value.

VI. CONCLUSION

This paper studies the multirobot active mapping problem and proposes AIM-Mapping, which is an effective mapping framework based on deep reinforcement learning. The framework uses an asymmetric feature representation module to encode the disparity of observation and privilege information,

TABLE IV
ABLATION EXPERIMENT RESULTS

Methods	Small Scene ($< 60m^2$)		Medium Scene ($60 - 100m^2$)		Large Scene ($> 100m^2$)	
	Time (step)	Explo (%)	Time (step)	Explo (%)	Time (step)	Explo (%)
Ablated privileged	605.08	97.32	950.00	96.62	1495.91	96.16
Ablated MI evaluation	686.70	97.55	977.90	**96.82**	1481.36	**96.20**
AIM-Mapping	**542.35**	**97.67**	**803.37**	96.76	**1341.59**	96.03

and use the disparity feature as the state value of the actor-critic training framework. The mutual information between observation and privilege information will be used as the supervised information of the above framework. For decision-making, a topological representation is first constructed incorporating both structure information and geometric distance information. A graph matching mechanism is then applied to assign the goal point to each robot. Qualitative and quantitative experiments are conducted on the public iGibson environment and the results validate the effectiveness of the proposed method. In our future work, we plan to further explore the potential of the asymmetric information.

REFERENCES

[1] Alberto Vergnano et al. "Modeling and optimization of energy consumption in cooperative multi-robot systems". In: *IEEE Transactions on Automation Science and Engineering* 9.2 (2012), pp. 423–428.

[2] Wolfram Burgard et al. "Collaborative multi-robot exploration". In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. Vol. 1. IEEE. 2000, pp. 476–481.

[3] Kai M Wurm, Cyrill Stachniss, and Wolfram Burgard. "Coordinated multi-robot exploration using a segmentation of the environment". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2008, pp. 1160–1165.

[4] Lukas Klodt and Volker Willert. "Equitable workload partitioning for multi-robot exploration through pairwise optimization". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 2809–2816.

[5] Siyan Dong et al. "Multi-robot collaborative dense scene reconstruction". In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–16.

[6] Brian Yamauchi. "A frontier-based approach for autonomous exploration". In: *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE. 1997, pp. 146–151.

[7] Miguel Juliá, Arturo Gil, and Oscar Reinoso. "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments". In: *Autonomous Robots* 33 (2012), pp. 427–444.

[8] Haye Lau. "Behavioural approach for multi-robot exploration". In: *Australasian Conference on Robotics and Automation*. Australian Robotics and Automation Association Inc. 2003.

[9] Kaiwen Li et al. "Deep reinforcement learning for combinatorial optimization: Covering salesman problems". In: *IEEE transactions on cybernetics* 52.12 (2021), pp. 13142–13155.

[10] Szilárd Aradi. "Survey of deep reinforcement learning for motion planning of autonomous vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 23.2 (2020), pp. 740–759.

[11] Chao Yan et al. "Deep reinforcement learning of collision-free flocking policies for multiple fixed-wing uavs using local situation maps". In: *IEEE Transactions on Industrial Informatics* 18.2 (2021), pp. 1260–1270.

[12] Lei Xue et al. "Extended kalman filter based resilient formation tracking control of multiple unmanned vehicles via game-theoretical reinforcement learning". In: *IEEE Transactions on Intelligent Vehicles* (2023).

[13] Ho-Bin Choi et al. "MARL-based cooperative multi-AGV control in warehouse systems". In: *IEEE Access* 10 (2022), pp. 100478–100488.

[14] Rafael Gonçalves Colares and Luiz Chaimowicz. "The next frontier: Combining information gain and distance cost for decentralized multi-robot exploration". In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. 2016, pp. 268–274.

[15] Subhrajit Bhattacharya, Robert Ghrist, and Vijay Kumar. "Multi-robot coverage and exploration on Riemannian manifolds with boundaries". In: *The International Journal of Robotics Research* 33.1 (2014), pp. 113–137.

[16] Charles W Warren. "Global path planning using artificial potential fields". In: *1989 IEEE International Conference on Robotics and Automation*. IEEE Computer Society. 1989, pp. 316–317.

[17] Alessandro Renzaglia and Agostino Martinelli. "Potential field based approach for coordinate exploration with a multi-robot team". In: *2010 IEEE Safety Security and Rescue Robotics*. IEEE. 2010, pp. 1–6.

[18] Tsung-Ming Liu and Damian M Lyons. "Leveraging area bounds information for autonomous decentralized multi-robot exploration". In: *Robotics and Autonomous Systems* 74 (2015), pp. 66–78.

[19] Jincheng Yu et al. "Smmr-explore: Submap-based multi-robot exploration system with multi-robot multi-target potential field exploration method". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 8779–8785.

[20] Shahriar Tanvir Alam et al. "A Comparative Analysis of Assignment Problem". In: *International Conference on Big Data Innovation for Sustainable Cognitive Computing*. Springer. 2022, pp. 125–142.

[21] Patrick Bernard and Boris Buffoni. "Optimal mass transportation and Mather theory". In: *Journal of the European Mathematical Society* 9.1 (2007), pp. 85–121.

[22] Jan Faigl, Miroslav Kulich, and Libor Přeučil. "Goal assignment using distance cost in multi-robot exploration". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 3741–3746.

[23] Tolga Bektas. "The multiple traveling salesman problem: an overview of formulations and solution procedures". In: *omega* 34.3 (2006), pp. 209–219.

[24] Lillian Clark et al. "A queue-stabilizing framework for networked multi-robot exploration". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 2091–2098.

[25] Leandros Tassiulas and Anthony Ephremides. "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks". In: *29th IEEE Conference on Decision and Control*. IEEE. 1990, pp. 2130–2132.

[26] Peter Whaite and Frank P Ferrie. "Autonomous exploration: Driven by uncertainty". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.3 (1997), pp. 193–205.

[27] Alberto Elfes. "Robot navigation: Integrating perception, environmental constraints and task execution within a probabilistic framework". In: *International Workshop on Reasoning with uncertainty in Robotics*. Springer. 1995, pp. 91–130.

[28] Stewart J Moorehead, Reid Simmons, and William L Whittaker. "Autonomous exploration using multiple sources of information". In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*. Vol. 3. IEEE. 2001, pp. 3098–3103.

[29] Frederic Bourgault et al. "Information based adaptive robotic exploration". In: *IEEE/RSJ international conference on intelligent robots and systems*. Vol. 1. IEEE. 2002, pp. 540–545.

[30] Arnoud Visser and Bayu A Slamet. "Balancing the information gain against the movement cost for multi-robot frontier exploration". In: *European Robotics Symposium 2008*. Springer. 2008, pp. 43–52.

[31] Wolfram Burgard et al. "Coordinated multi-robot exploration". In: *IEEE Transactions on robotics* 21.3 (2005), pp. 376–386.

[32] Anna Dai et al. “Fast frontier-based information-driven autonomous exploration with an mav”. In: *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2020, pp. 9570–9576.

[33] Theia Henderson, Vivienne Sze, and Sertac Karaman. “An efficient and continuous approach to information-theoretic exploration”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 8566–8572.

[34] Arash Asgharivaskasi and Nikolay Atanasov. “Semantic OcTree mapping and Shannon mutual information computation for robot exploration”. In: *IEEE Transactions on Robotics* 39.3 (2023), pp. 1910–1928.

[35] Mingyang Geng et al. “Learning to cooperate in decentralized multi-robot exploration of dynamic environments”. In: *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part VII 25*. Springer. 2018, pp. 40–51.

[36] Mingyang Geng et al. “Learning to cooperate via an attention-based communication neural network in decentralized multi-robot exploration”. In: *Entropy* 21.3 (2019), p. 294.

[37] Aaron Hao Tan et al. “Deep reinforcement learning for decentralized multi-robot exploration with macro actions”. In: *IEEE Robotics and Automation Letters* 8.1 (2022), pp. 272–279.

[38] Lina Zhu et al. “Multi-Robot Environmental Coverage With a Two-Stage Coordination Strategy via Deep Reinforcement Learning”. In: *IEEE Transactions on Intelligent Transportation Systems* (2024).

[39] Junyan Hu et al. “Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning”. In: *IEEE Transactions on Vehicular Technology* 69.12 (2020), pp. 14413–14423.

[40] Devendra Singh Chaplot et al. “Learning to explore using active neural slam”. In: *arXiv preprint arXiv:2004.05155* (2020).

[41] Chao Yu et al. “Learning efficient multi-agent cooperative visual exploration”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 497–515.

[42] Kai Ye et al. “Multi-robot active mapping via neural bipartite graph matching”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14839–14848.

[43] Max Lodel et al. “Where to look next: Learning viewpoint recommendations for informative trajectory planning”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 4466–4472.

[44] Eun Sun Lee and Young Min Kim. “Multi-Agent Exploration With Similarity Score Map and Topological Memory”. In: *IEEE Robotics and Automation Letters* (2024).

[45] James A Sethian. “Fast marching methods”. In: *SIAM review* 41.2 (1999), pp. 199–235.

[46] Kevin Chen et al. “Topological planning with transformers for vision-and-language navigation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11276–11286.