IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Active Neural Topological Mapping for Multi-Agent Exploration

Xinyi Yang[1], Yuxiang Yang[1], Chao Yu[1], Jiayu Chen[1], Jincheng Yu[1], Haibing Ren[2], Huazhong Yang[1], and Yu Wang[1]

Abstract—**This paper investigates the multi-agent cooperative exploration problem, which requires multiple agents to explore an unseen environment via sensory signals in a limited time. A popular approach to exploration tasks is to combine active mapping with planning. Metric maps capture the details of the spatial representation, but are with high communication traffic and may vary significantly between scenarios, resulting in inferior generalization. Topological maps are a promising alternative as they consist only of nodes and edges with abstract but essential information and are less influenced by the scene structures. However, most existing topology-based exploration tasks utilize classical methods for planning, which are time-consuming and sub-optimal due to their handcrafted design. Deep reinforcement learning (DRL) has shown great potential for learning (near) optimal policies through fast end-to-end inference. In this paper, we propose Multi-Agent Neural Topological Mapping (MANTM) to improve exploration efficiency and generalization for multi-agent exploration tasks. MANTM mainly comprises a Topological Mapper and a novel RL-based Hierarchical Topological Planner (HTP). The Topological Mapper employs a visual encoder and distance-based heuristics to construct a graph containing main nodes and their corresponding ghost nodes. The HTP leverages graph neural networks to capture correlations between agents and graph nodes in a coarse-to-fine manner for effective global goal selection. Extensive experiments conducted in a physically-realistic simulator, Habitat, demonstrate that MANTM reduces the steps by at least 26.40% over planning-based baselines and by at least 7.63% over RL-based competitors in unseen scenarios.**

Index Terms—**Reinforcement Learning, Path Planning for Multiple Mobile Robots or Agents**

I. Introduction

EXploration [1] is one of the fundamental building blocks for developing intelligent autonomous agents, and is widely applied in autonomous driving [2], disaster rescue [3], and planetary exploration [4]. In this paper, we focus on multi-agent exploration, where agents simultaneously explore an unknown scene via sensory signals. To achieve efficient exploration, we typically employ a workflow of autonomous map construction and collaborative planning.

The spatial arrangement of metric maps [5], [6] can vary significantly between scenes, which hinders the generalization ability for exploration. Metric maps also perform poorly in efficiency, as they are with high communication traffic and have difficulty scaling to larger environments. In contrast, topological maps, which contain abstract but essential information, require less communication traffic and are less sensitive to changes in scene structure. Therefore, applying topological maps [7], [8] offers significant generalization potential and high efficiency.

Topology-based exploration tasks commonly utilize classical methods [9], [10] for planning due to their minimal training time and direct deployment for evaluation. However, they often suffer from numerous handcrafted parameters and rule-based coordination strategies, which limits their effectiveness. In contrast, DRL has shown potential for topological exploration [11], [12] due to its ability to model arbitrarily complex strategies and execute real-time actions. However, these methods are based on pre-built graphs [11], [13] or tested on simple grid maps [12]. Applying active topological mapping in RL-based multi-agent exploration is confronted with the following limitations: (a) the number of nodes in the merged graph is large and constantly changing during exploration, leading to unstable RL training and suboptimal results in such a large and varying search space; (b) capturing complex relationships between agents and topological maps is difficult, resulting in an unbalanced workload distribution among agents.

To address the above challenges, we propose *Multi-Agent Neural Topological Mapping* (MANTM), an RL-based topological solution for multi-agent exploration. We adopt a modular exploration strategy that divides the planning process into two phases. In the first phase, the global planner infers global goals in each global decision-making step. Subsequently, the local planner predicts the environmental actions to encourage the agents to reach the global goals. MANTM comprises a Topological Mapper to build topological maps, a Hierarchical Topological Planner (HTP) to infer the global goals, and a Local Planner and a Local Policy to generate environmental actions. The Topological Mapper employs a visual encoder and distance-based heuristics to construct graphs with main nodes (i.e., explored areas) and ghost nodes (i.e., unexplored areas). To build more accurate graphs, each agent also maintains a predicted metric map to identify explored areas and prune graphs. We remark that the metric maps are not shared among agents and are not used for global planning, thus promising

Manuscript received: July, 25, 2023; Revised September, 19, 2023; Accepted October, 21, 2023.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by National Natural Science Foundation of China (No.62203257, 62325405), Tsinghua University Initiative Scientific Research Program, Tsinghua-Meituan Joint Institute for Digital Life, Beijing National Research Center for Information Science, Technology (BNRist) and Beijing Innovation Center for Future Chips. *(Xinyi Yang and Yuxiang Yang contributed equally to this work.) (Corresponding authors: Chao Yu and Yu Wang.)*

[1]Xinyi Yang, Yuxiang Yang, Chao Yu, Jiayu Chen, Jingchen Yu, Huazhong Yang, and Yu Wang are with Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China (e-mail: yang-xy20@mails.tsinghua.edu.cn)

[2]Haibing Ren is with Meituan, Beijing, 100012, China

More information can be found at https://sites.google.com/view/mantm

Digital Object Identifier (DOI): see top of this page.

reduced communication traffic and enhanced generalization capabilities. The RL-based global planner, the Hierarchical Topological Planner (HTP), is the most crucial component that selects a main node and then chooses a corresponding ghost node as a global goal for each agent at each global step. This hierarchical goal selection significantly reduces the search space with much fewer candidate nodes, thus addressing challenge (a). Furthermore, the HTP leverages graph neural networks to capture the relationship between agents and topological maps, solving challenge (b).

We conduct extensive experiments in the physically realistic simulator, Habitat [14]. The experimental results demonstrate that MANTM has at least 7.63% fewer steps than RL-based baselines, and at least 26.40% fewer steps than planning-based competitors in unseen scenarios.

II. Related Work

A. Multi-Agent Exploration

In classical visual exploration, agents first perform simultaneous localization and mapping (SLAM) [15] to locate their position and reconstruct 2D maps from sensory signals. They then utilize search-based planning algorithms to generate valid exploration trajectories. Representative works mainly include frontier-based methods [16], [17], sampling-based methods [18], [19], and graph-based methods [9], [20]. However, these solutions suffer from expensive computational costs and limited representation capabilities. Recently, deep reinforcement learning [21], [22], [23] has attracted significant attention due to its powerful expressiveness. NeuralCoMapping [21] utilizes a multiplex graph neural network to choose effective frontiers as global goals. MAANS [22] uses a transformer-based architecture to infer spatial relationships and intra-agent interactions. However, all these approaches are based on metric maps, which are sensitive to different scene structures and result in subpar generalization. In this paper, we introduce an RL-based topological approach for efficient exploration and superior generalization in unseen scenarios.

B. Spatial Representation

Spatial representation usually includes two main types of maps: metric and topological maps. Metric maps are grid maps where each grid predicts its traversability [5], [24]. However, metric maps struggle with generalization due to significant structural variations across different scenes. In contrast, topological maps [25] abstractly preserve essential environmental features with nodes and edges, offering a potential solution for improved generalization. Several works [11], [13] are based on pre-built graphs, focusing on graph refinement or finding optimal paths for coverage. Recent literature [7], [8], [26] utilizes active topological mapping for navigation tasks. For instance, [7] employs the cosine similarity of visual embeddings to construct graphs. However, it requires expert trajectories which are difficult to acquire in the NP-hard multi-agent exploration problem. Furthermore, [26] leverages depth images to predict explored/unexplored nodes, while [8] utilizes semantics for approximate geometric reasoning in topological representations. However, [26], [8] requires a predefined goal to predict the geodesic distance from unexplored nodes and select a node with the shortest predicted distance. This is unsuitable for multi-agent exploration where there are no pre-defined goals. In this work, we introduce an RL-based Hierarchical Topological Planner to effectively apply active topological mapping in multi-agent exploration.

C. Graph Neural Networks

Graph neural networks (GNNs) [27] are widely utilized in multi-agent systems to model interactions between agents. [28] proposes a hierarchical graph attention network that captures the underlying relationships at the agent and the group level, thereby improving generalization. [7] considers GNN as an encoder for mapping tasks to extract node features from images. Additionally, [21] formulates exploration tasks as bipartite graph matching. In this paper, we propose a hierarchical goal selector based on GNN to capture the correlations between agents and topological maps in a coarse-to-fine manner.

III. Task Setup

Multi-agent cooperative exploration requires agents to explore an unknown scene based on sensory signals. At each time step, each agent receives a first-person RGB-D image and the estimated pose from sensors. Agents then perform environmental actions in the physically realistic simulator, Habitat [14]. The horizon of the global decision-making step is 15 steps, and the available environmental actions include *Turn Left*, *Turn Right*, and *Forward*. Following the settings in ANS [5] and NRNS [26], we introduce Gaussian noise in the sensor readings and simulate real-world action noise. In the multi-agent scenario, we further consider the following settings. Firstly, we assume perfect communication, where relative spawn locations are shared between agents. This allows us to estimate the relative position of each agent at each timestep by using sensory pose readings and shared spawn locations. Besides, agents are randomly initialized within a 2-meter geodesic distance constraint. This spatially close initialization requires agents to expend more scanning effort for exploration, further increasing the difficulty of cooperation.

IV. Methodology

We follow the paradigm of centralized training and decentralized execution (CTDE) with partial observations of N agents, where agent k receives local observation, o_t^k, at timestep t. The overview of MANTM is depicted in Fig. 1. Each agent shares the same Topological Mapper, Hierarchical Topological Planner, Local Planner, and Local Policy while making decisions independently. The Topological Mapper of each agent utilizes a visual encoder and distance-based heuristics to construct a topological map based on RGB-D images and estimated poses from sensory signals. For better cooperation, the Topological Mapper transforms all individual maps into the same coordination and merges them. The Hierarchical Topological Planner (HTP) receives graphs that respectively contain current agent information, main node, and ghost nodes, along with corresponding historical graphs containing agent trajectories, selected main nodes, and selected

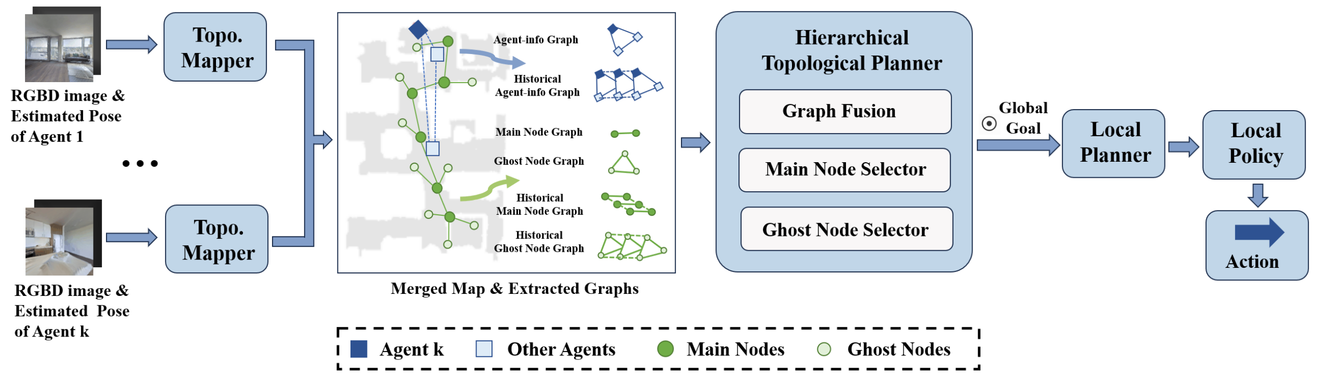

RGBD image & Estimated Pose of Agent 1
Topo. Mapper
RGBD image & Estimated Pose of Agent k
Topo. Mapper
Agent-info Graph
Historical Agent-info Graph
Main Node Graph
Ghost Node Graph
Historical Main Node Graph
Historical Ghost Node Graph
Merged Map & Extracted Graphs
Hierarchical Topological Planner
Graph Fusion
Main Node Selector
Ghost Node Selector
Global Goal
Local Planner
Local Policy
Action
Agent k
Other Agents
Main Nodes
Ghost Nodes

Fig. 1: Overview of *Multi-Agent Neural Topological Mapping* (MANTM). Here, we take Agent k as an example.

ghost nodes from the merged topological map. HTP utilizes GNN on these graphs to hierarchically capture spatio-temporal and intra-agent information. It then selects a ghost node as a global goal at each global step. Finally, the Local Planner plans a reachable path to the predicted global goal via the Fast Marching Method (FMM) [29], and the Local Policy generates environmental actions based on the reachable path. We remark that the Local Planner and the Local Policy do not involve multi-agent interactions, so they are directly adopted from ANS [5].

A. Topological Mapper

We introduce a Topological Mapper to provide a merged topological map, as shown in Fig. 1. Inspired by [8], [26], we consider two types of nodes in the topological map: main nodes and ghost nodes. Main nodes are located where agents have already explored. Ghost nodes are located in the unexplored area and adjacent to the corresponding main node.

In map construction, the Topological Mapper relies on a visual encoder and distance-based heuristics. Firstly, we adopt an encoder [7] to predict visual embeddings of panoramic RGB-D images. The topological map constructs main nodes based on the cosine similarity between visual embeddings. Specifically, when the cosine similarity between the visual embedding at the current location and that of existing main nodes is below a threshold (i.e., 0.75 in our work), a new main node is constructed at that location and connected to the main node that the agent most recently localized. However, images of some spatially irrelevant locations may be similar (e.g., images of corners and corridors both contain a large portion of the wall), resulting in incorrect connections between main nodes. To deal with this problem, we delete the connection of main nodes with far-distant FMM distance at each global step. Once a main node is constructed, m new ghost nodes are uniformly adjacent to this main node at a distance λ. Ghost nodes can be removed if they are located in the explored areas identified by the current agent's predicted metric map via SLAM. We remark that the metric map is just for identifying explored regions and not for global planning. Besides, a ghost node can be converted to a main node if the agent passes through it. Therefore, the potential number of remaining ghost nodes ranges from 0 to $m \times N_{main}$, where N_{main} is the number of main nodes. In our work, we choose $\lambda = 3$ meters and $m = 12$. Moreover, we delete spurious ghost nodes and their connected edges based on the FMM distance between two ghost nodes belonging to two different main nodes and between a main node and a ghost node belonging to another main node.

For better cooperation, we transform all individual topological maps into the same coordinate system and merge them based on the estimated poses of the agents at each global step. During merging, if the distance between two main nodes from different maps is below a threshold (i.e., 3 meters in our work), we randomly remove one of the nodes and redirect its connected edges to the remaining node.

B. Hierarchical Topological Planner

The Hierarchical Topological Planner (HTP) selects a ghost node as a global goal at each global step. However, it is difficult for MARL to directly explore the (near) optimal strategy due to the large and varying search space associated with the number of ghost nodes. To address this issue, we propose a hierarchical network that matches agents to main nodes and then to the corresponding ghost nodes. Therefore, the HTP can be easily optimized in the reduced search space with much fewer candidate nodes and potentially provide a more appropriate probability distribution of ghost nodes in a coarse-to-fine manner. The workflow of the HTP is illustrated in Fig. 2. Firstly, at each global step, we extract an agent-info graph, G_a, a main node graph, G_m, a ghost node graph, G_g, and three corresponding historical graphs, $\hat{G}$, from a merged topological map. Subsequently, we update each G with its historical graph in the Memory Fusion to aggregate current and historical information. The Main Node Selector computes matching scores between the agent and the main nodes via the attention mechanism [30]. Afterward, the Ghost Node Selector calculates the probability distribution of ghost nodes based on ghost node features and the matching results from the Main Node Selector. Finally, the Action Generator takes in the probability distribution of ghost nodes and determines the most appropriate ghost node as the global goal at each global decision-making step.

We remark that G_a contains the current agent information, while G_m contains current main node features, and G_g includes current ghost node features. Besides, in historical graphs, $\hat{G_a}$ contains agent trajectories, $\hat{G_m}$ contains selected main node features, and $\hat{G_g}$ includes selected ghost node features from past global steps. Each node consists of its grid position, $(x, y) \in R^2, x, y \in [0, map\ size]$, and a semantic label, $(s_1, s_2) \in$

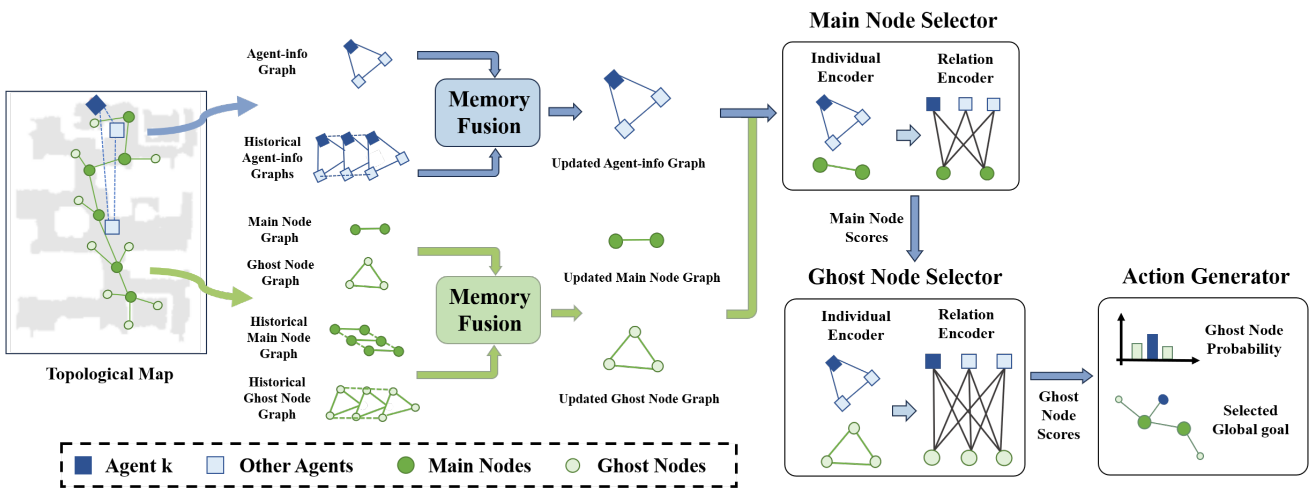


Fig. 2: Workflow of *Hierarchical Topological Planner* (HTP), including a Memory Fusion, a Main Node Selector, a Ghost Node Selector and an Action Generator. HTP is under decentralized decision-making setting. We take Agent k as an example.

$N^2, N = \{0, 1\}$. This semantic label represents the type of node, including agent nodes, historical agent nodes, map nodes, and historical map nodes.

1) Memory Fusion: Exploration is a memory-based task [31] which heavily relies on historical information. Therefore, this module incorporates each graph with its historical counterpart to mitigate the risk of memory loss. Concretely, we first use a weight-shared multi-layer perception (MLP) layer to encode node features due to their consistent information types across different graphs. Then, the Memory Fusion merges each graph with its corresponding historical graph and yields an updated graph, $\tilde{G}$, via the self-attention and the cross-attention mechanisms [30]. The output dimensions of the last cross-attention layer match those of the original node features, ensuring that the shape of $\tilde{G}$ remains consistent with that of G.

2) Main Node Selector: The Main Node Selector is introduced for the high-level goal selection and yields the matching scores between the agent and the main nodes. It predicts the next preferred region to explore since a main node with a higher matching score implies a higher probability of selecting its corresponding ghost nodes as the global goal. More concretely, this module receives $\tilde{G}_a$ and $\tilde{G}_m$ and then leverages an Individual Encoder and a Relation Encoder to infer the matching scores, $S_{m,re}$. The Individual Encoder perceives the states of the agents and the spatial information of the main nodes, while the Relation Encoder captures the correlation between the agents and the main nodes.

Individual Encoder: The Individual Encoder captures relationships between any two nodes in the same graph and updates these nodes. As shown in Fig. 3(a), this module first calculates the normalized matching scores of any two nodes:

$$S_{m,in} = Softmax(W_Q X \times W_K X). \tag{1}$$

Here X denotes the node features, while W_Q and W_K represent the linear projections of X. The $S_{m,in}$ is a matrix where each element, $S_{m,in}^{(i,j)}$, refers to the matching score between node i and node j.

After that, each node is updated by an MLP layer, f_{in}, with a residual connection. The input to f_{in} is the concatenation of each node feature and the weighted sum of its neighboring features. The update of each node is formulated as:

$$X^{(t+1)} = X^t + f_{in}(X^t, W_V X^t \times S_{m,in}^T), \tag{2}$$

where X^t represents the node features at time step t, and W_V is the linear projection of X^t. The matching scores between nodes in a graph, $S_{m,in}$, serve as the weights for neighboring nodes.

Relation Encoder: The Relation Encoder captures correlations between any two nodes from different graphs. The architecture of the Relation Encoder is shown in Fig. 3(b). Considering the node features in two different graphs as Y and Z, we calculate the normalized matching scores for each node pair (y, z) by taking a softmax operator over the outputs of an MLP layer, f_{dis}. The input to f_{dis} is the concatenation of $W_Q Y$, $W_K Z$, and d_{fmm}, where d_{fmm} is the FMM distance between the given node pair. The calculation of the scores is formulated as:

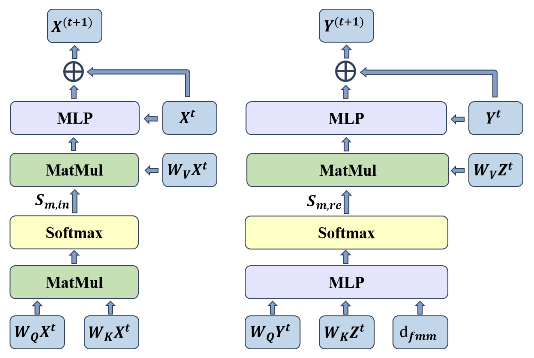
$$S_{m,re} = Softmax(f_{dis}(W_Q Y, W_K Z, d_{fmm})). \tag{3}$$

Afterward, we update all nodes via an MLP layer, f_{re}, with a residual connection:

$$Y^{(t+1)} = Y^t + f_{re}(Y^t, W_V Z^t \times S_{m,re}^T). \tag{4}$$

Here W represents linear projections, while Y^t and Z^t denote different node features at time step t. Note that $S_{m,re}^{(k)}$ denotes the matching score between agent k and the main nodes and is then sent to the Ghost Node Selector for agent k.

3) Ghost Node Selector: For the low-level goal selection, the Ghost Node Selector provides a probability distribution of ghost nodes and predicts the next location to explore based on the preferred region. Similar to the Main Node Selector, the Ghost Node Selector first receives $\tilde{G}_a$ and $\tilde{G}_g$ and provides ghost node scores, $S_{g,re}$, via an Individual Encoder and a Relation Encoder. The Individual Encoder captures intra-agent interactions and spatial information of ghost nodes, while the Relation Encoder infers the correlations between agents and ghost nodes. Additionally, we update $S_{g,re}^{(k,j)}$ by multiplying



(a) Individual Encoder (b) Relation Encoder

Fig. 3: Network architecture of the Individual Encoder and the Relation Encoder.

it with $S_{m,re}^{(k,h)}$ to aggregate information from main and ghost nodes for more appropriate goal selection:

$$\hat{S}_{g,re}^{(k,j)} = S_{g,re}^{(k,j)} \times S_{m,re}^{(k,h)}. \tag{5}$$

Here, k is a specific agent, j is the main node, and h denotes the corresponding ghost node of the main node j. $\hat{S}_{g,re}^{(k)}$ is the final matching score of agent k and ghost nodes.

4) Action Generator: The Action Generator selects a global goal from all ghost nodes at each global step. We regard $\hat{S}_{g,re}$ as the probability distribution over the ghost nodes. The action space for the HTP is $A = \{a|a \in ghost\ nodes\}$, where a is a discrete variable sampled from a categorical distribution, indicating the index of the selected ghost node.

C. Reinforcement Learning Design

We train the HTP by using multi-agent proximal policy optimization (MAPPO) [32], which is a multi-agent variant of proximal policy optimization (PPO) [33].

Reward Function: To promote efficient and cooperative exploration, we introduce four types of rewards. The coverage reward, R_{cov}, represents the increase of the explored area to encourage thorough exploration. Besides, the success reward, R_{suc}, offers a bonus for reaching the target coverage ratio. The overlap penalty, R_o, denotes the overlapped area passed by agents to encourage cooperation. Finally, for efficient exploration, agents receive a constant time penalty, R_t, at each timestep until they attain the target coverage ratio. As a result, the reward function, R_{total}, is a linear combination of these kinds of rewards.

V. Experiments

A. Experimental Details

We conduct all experiments in the Habitat simulator [14], using the Gibson Challenge dataset [34] and the Habitat-Matterport 3D dataset (HM3D) [35]. The scenes in these datasets are collected from real building-scale residential, commercial, and civic spaces using 3D scanning and reconstruction. We filter out some scenes that are inappropriate for our task, following [22], which is one of the best RL-based approaches for cooperative exploration. This filtering process involved removing scenes with large disconnected regions or multiple floors where agents couldn't attain 90% coverage of the entire house. Furthermore, we exclude scenes smaller than 70 m^2, as their topological maps would contain too few nodes to fully show the advantages of graphs. To better demonstrate the robustness of MANTM on training scenes and its effective generalization to novel scenes, we follow [22], [21] by dividing the remaining scenes into 10 training scenes from the Gibson Challenge dataset and 28 testing scenes from both datasets. We perform RL training with 10^6 timesteps over 3 random seeds. Each evaluation score has the format of "mean (standard deviation)", and is averaged over 300 testing episodes.

B. Evaluation Metrics

We consider 3 statistical metrics to capture different characteristics of a particular exploration strategy. These metrics are only for analysis, and we primarily focus on *Steps* as our performance criterion.

- **Steps:** This metric considers the timesteps required to achieve 90% coverage within an episode. Fewer *Steps* imply faster exploration.
- **Coverage:** This metric denotes the final ratio of the explored area to total explorable area at the end of the episode. A higher *Coverage* ratio reflects a more effective exploration.
- **Mutual Overlap:** This metric shows the ratio of the overlapped area to the currently explored area when the *Coverage* ratio achieves 90%. Lower *Mutual Overlap* ratio indicates better collaboration.

C. Baselines

We challenge MANTM against three representative planning-based approaches (CoScan, Topological Frontier, Voronoi) and three prominent RL-based solutions (ANS-Merge, NeuralCoMapping, MAANS). Note that Topological Frontier and Voronoi are also graph-based approaches.

- **CoScan** [17]: This frontier-based method applies k-means clustering to all frontiers and assigns a frontier cluster to each agent. Afterward, each agent plans an optimal traverse path over the assigned frontiers.
- **Topological Frontier (TF)** [9]: This graph-based approach calculates a normalized traveling cost for each ghost node built from the Topological Mapper and considers the node with the lowest cost as the global goal.
- **Voronoi** [20]: This graph-based solution divides the map into several parts and transforms it into a Voronoi graph. Each agent then only searches the unexplored region in its partition, reducing the overlapped area.
- **ANS-Merge** [5]: ANS is exemplary in RL-based single-agent exploration. It takes in egocentric local and global metric maps and infers global goals for the agents. We extend ANS to multi-agent exploration by sending merged maps to the global planner and use the same reward function as ours.
- **NeuralCoMapping (NCM)** [21]: NeuralCoMapping introduces a multiplex graph neural network to predict the neural distance between frontier nodes and agents.

Agents	Sce.	Metrics	CoScan	NF	Voronoi	ANS-Merge	NCM	MAANS	MANTM
N =3	Middle (>70m^2)	*Mut. Over.* ↓	0.39(0.01)	0.56(0.01)	**0.29(0.01)**	0.44(0.01)	0.38(0.01)	0.39(0.01)	0.36(0.02)
		Steps ↓	244.86(7.25)	231.72(5.43)	226.35(3.51)	211.88(4.23)	202.20(5.42)	185.31(5.81)	**166.59(4.19)**
		Coverage ↑	0.89(0.02)	0.92(0.01)	0.92(0.01)	0.94(0.01)	**0.97(0.01)**	0.96(0.01)	**0.97(0.01)**
	Large (>100m^2)	*Mut. Over.*↓	0.42(0.01)	0.65(0.03)	**0.33(0.01)**	0.52(0.02)	0.45(0.01)	0.43(0.01)	0.38(0.01)
		Steps ↓	485.74(6.89)	454.34(6.89)	439.81(8.47)	386.54(11.96)	381.84(8.63)	379.45(5.69)	**323.72(10.64)**
		Coverage ↑	0.90(0.03)	0.92(0.01)	0.88(0.01)	0.94(0.01)	0.93(0.01)	0.95(0.01)	**0.96(0.01)**
N = 4	Middle (>70m^2)	*Mut. Over.* ↓	0.36(0.01)	0.53(0.03)	**0.21(0.01)**	0.42(0.03)	0.35(0.01)	0.25(0.01)	0.22(0.01)
		Steps ↓	236.81(4.39)	219.35(3.68)	218.80(2.46)	173.30(4.14)	174.63(4.57)	162.09(3.94)	**149.72(1.94)**
		Coverage ↑	0.97(0.03)	0.97(0.01)	0.95(0.01)	**0.98(0.01)**	0.97(0.01)	**0.98(0.01)**	**0.98(0.01)**
	Large (>100m^2)	*Mut. Over.*↓	0.36(0.01)	0.60(0.01)	**0.25(0.01)**	0.45(0.03)	0.38(0.01)	0.38(0.01)	0.28(0.01)
		Steps ↓	479.47(7.35)	425.88(6.15)	418.49(5.23)	325.29(5.48)	322.27(8.67)	315.80(3.55)	**284.50(3.66)**
		Coverage ↑	0.91(0.03)	**0.96(0.01)**	**0.96(0.01)**	0.95(0.01)	0.95(0.01)	0.95(0.01)	**0.96(0.01)**

TABLE I: Performance of MANTM, planning-based baselines, and RL-based baselines with $N = 3, 4$ agents on the Gibson dataset. Note that the horizon of middle and large maps is 300 steps and 600 steps, respectively.

Agents	Sce.	Metrics	CoScan	NF	Voronoi	ANS-Merge	NCM	MAANS	MANTM
N =3	Middle (>70m^2)	*Mut. Over.* ↓	0.40(0.01)	0.63(0.04)	**0.32(0.03)**	0.48(0.03)	0.41(0.03)	0.46(0.07)	0.36(0.01)
		Steps ↓	354.02(5.40)	333.09(4.26)	256.78(2.58)	279.72(5.38)	268.03(3.43)	267.23(4.82)	**238.10(3.39)**
		Coverage ↑	0.91(0.01)	0.95(0.01)	0.96(0.01)	0.96(0.01)	0.96(0.01)	**0.97(0.01)**	**0.97(0.01)**
	Large (>100m^2)	*Mut. Over.*↓	0.40(0.01)	0.63(0.01)	**0.28(0.03)**	0.52(0.05)	0.42(0.03)	0.55(0.07)	0.39(0.04)
		Steps ↓	698.22(8.16)	649.60(11.23)	509.24(5.54)	497.41(11.60)	463.75(12.10)	458.69(14.04)	**419.88(7.91)**
		Coverage ↑	0.82(0.01)	0.92(0.01)	0.94(0.01)	0.93(0.01)	0.95(0.01)	0.95(0.01)	**0.96(0.01)**
	Super Large (>200m^2)	*Mut. Over.*↓	0.33(0.01)	0.63(0.01)	**0.28(0.01)**	0.45(0.04)	0.41(0.01)	0.40(0.04)	0.35(0.01)
		Steps ↓	1710.88(41.11)	1456.80(48.94)	1321.62(44.73)	1343.17(50.68)	1147.25(57.26)	1135.24(53.06)	**982.54(43.51)**
		Coverage ↑	0.82(0.02)	0.87(0.01)	0.87(0.01)	0.92(0.01)	0.92(0.01)	0.94(0.02)	**0.97(0.01)**
N = 4	Middle (>70m^2)	*Mut. Over.* ↓	0.35(0.01)	0.57(0.02)	**0.30(0.02)**	0.34(0.01)	0.35(0.03)	0.34(0.01)	0.35(0.01)
		Steps ↓	280.38(4.86)	314.12(4.30)	233.51(2.79)	227.42(5.81)	227.38(2.10)	219.68(5.64)	**191.31(1.66)**
		Coverage ↑	0.95(0.01)	0.96(0.01)	**0.97(0.01)**	**0.97(0.01)**	0.96(0.02)	**0.97(0.01)**	**0.97(0.01)**
	Large (>100m^2)	*Mut. Over.*↓	0.37(0.01)	0.58(0.02)	**0.29(0.01)**	0.44(0.02)	0.42(0.02)	0.41(0.01)	0.38(0.01)
		Steps ↓	608.14(7.24)	601.15(8.60)	407.31(8.34)	384.27(9.63)	389.12(8.53)	363.75(7.87)	**345.52(7.85)**
		Coverage ↑	0.92(0.01)	0.90(0.02)	0.96(0.01)	0.95(0.01)	**0.97(0.01)**	**0.97(0.02)**	**0.97(0.02)**
	Super Large (>200m^2)	*Mut. Over.*↓	0.29(0.01)	0.59(0.01)	**0.22(0.01)**	0.35(0.01)	0.34(0.01)	0.31(0.02)	0.33(0.01)
		Steps ↓	1538.13(22.13)	1338.95(47.36)	1264.12(39.34)	1156.49(43.60)	1064.16(42.85)	1012.52(38.62)	**900.71(29.24)**
		Coverage ↑	0.89(0.01)	0.90(0.01)	0.91(0.02)	0.91(0.01)	0.94(0.02)	0.93(0.01)	**0.97(0.01)**

TABLE II: Performance of MANTM, planning-based baselines and RL-based baselines with $N = 3, 4$ agents on the HM3D dataset. Note that the horizon of middle, large, and super large maps is 450 steps, 720 steps, and 1800 steps, respectively.

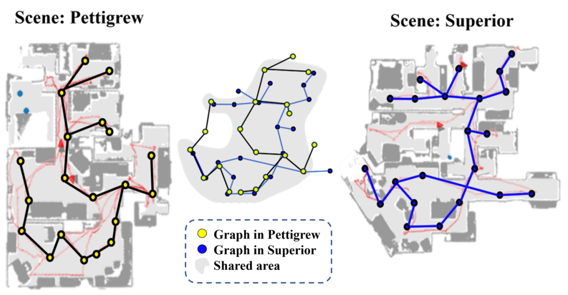
It then assigns each agent a frontier node based on the neural distance in each global step.

- **MAANS** [22]: MAANS is a variant of ANS for multi-agent exploration. This method leverages a transformer-based Spatial-TeamFormer to enhance cooperation. For a fair comparison, we conduct training on 10 maps without the policy distillation mentioned in [22].

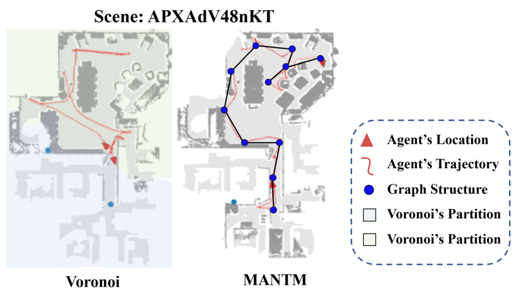
We remark that MANTM and the baselines are under the same assumptions in our task. All the baselines only replace the Topological Mapper and the HTP with alternatives and keep the rest the same as MANTM, except for TF, which only substitutes the HTP.

D. Main Results

1) Evaluation Results: The results in Tab. I show that MANTM outperforms all baselines with $N = 3, 4$ agents on unseen scenes on Gibson. In both middle and large maps, MANTM attains the fewest *Steps* and the highest *Coverage* ratio. More concretely, MAANS is the best RL baseline since its transformer-based Spatial-TeamFormer captures the spatial relationship and team representation, while MANTM has 10.10% and 7.63% fewer *Steps* than MAANS with $N = 3, 4$ agents, respectively. This indicates that ghost nodes, which are always located in unexplored areas, motivate agents to explore unseen regions. MANTM also excels in the *Mutual Overlap* ratio among the RL solutions, suggesting that it better assigns global goals to agents in different unexplored directions.



(a) Comparison of Map Construction



(b) Comparison of Planning Strategy

Fig. 4: Case studies of Map Construction and Planning Strategy. (a) shows that the graph structures of different scenes seem congruous in general, marked in grey. (b) displays agents' trajectories in Voronoi and MANTM, respectively.

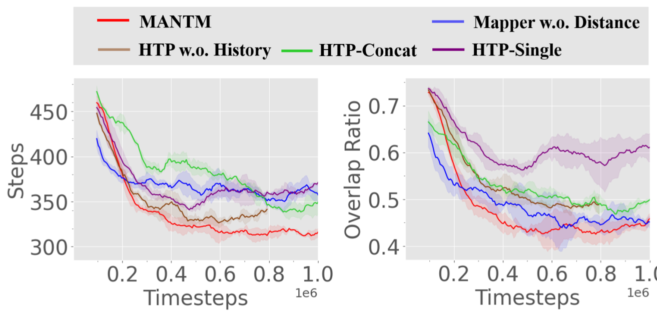


Fig. 5: Comparison between MANTM (red) and its variants. MANTM has the lowest *Steps* and *Mutual Overlap*

Among the planning-based baselines, the best competitor, Voronoi, achieves the lowest *Mutual Overlap* ratio in the 3-agent setting, demonstrating that the Voronoi separates agents to reduce the overlapped area. However, MANTM is superior to Voronoi in the *Steps* and the *Coverage* ratio, reducing *Steps* by 26.40% and 31.57% for $N = 3, 4$ agents, respectively. This implies that although the Voronoi partition separates agents, it may hinder exploration efficiency by confining agents to stay in their respective areas.

2) Domain Generalization: We also report the domain generalization performance in Tab. II, where all models trained on the Gibson dataset are evaluated in the HM3D domain. The results indicate that MANTM performs best. Compared to MAANS, the best competitor, MANTM achieves 13.45% and 11.04% fewer *Steps* with $N = 3, 4$ in super large scenes. This implies that the trained MAANS may overfit to the spatial arrangements in the training maps, resulting in suboptimal performance on unseen scenes in other domains due to variations in different metric maps. In contrast, MANTM exploits the graph structure with abstract but essential information to better adapt to scenarios in an unseen domain.

The planning-based methods fail to achieve an average *Coverage* ratio of 90% on super large maps. This indicates that the factors affecting exploration success are more complex on unseen super large maps. It is difficult for planning-based agents to take all factors into account in manual parameter tuning. As a result, agents may get stuck in the corner and fail to reach 90% *Coverage*.

3) Case Study: We present case studies of map construction in two different scenes to showcase the generalization of MANTM. Besides, to further demonstrate the cooperative exploration strategy of MANTM, we visualize the planning strategies of MANTM and the most competitive planning-based method, Voronoi.

In Fig. 4(a), the graph structures in two different scenes appear to be generally congruent (i.e., the area in grey). This suggests that topological maps, which contain abstract but essential information, are less influenced by scene structures, endowing them with significant generalization capabilities. Conversely, the shape of metric maps depends on the layout of the scene. As a result, finding a (near) optimal exploration strategy for various metric maps is challenging. In Fig. 4(b), the agent trajectories show that MANTM successfully allocates agents to different unexplored areas via the selected ghost nodes. Moreover, MANTM agents can temporarily revisit previously explored areas to reach unexplored areas in different directions, thereby increasing the final coverage. On the contrary, when the only path to the unexplored area belongs to the partition of a particular Voronoi agent (i.e., the area in blue), the other agent can only be constrained to its own partition (i.e., the area in green), resulting in inefficient exploration.

E. Ablation Study

We report the training performance of several RL variants to investigate the importance of the Mapper and the HTP.

- **Mapper w.o. Distance:** Without the help of the FMM distance, the Topological Mapper constructs the graph based solely on the similarity of the visual embeddings.
- **HTP w.o. History:** We abandon the historical graphs, $\hat{G}$, and the Graph Fusion. The Main Node Selector only takes G as input.
- **HTP-Single:** We abandon the Main Node Selector and only adopt the Ghost Node Selector to infer global goals.
- **HTP-Concat:** Before calculating $S_{g,re}$ in the Ghost Node Selector, we update ghost node features by concatenating them and the corresponding main node features. We then discard the multiplication in Equation 5 and directly consider $S_{g,re}$ as the final matching score.

As shown in Fig. 5, MANTM has the lowest *Steps* and *Mutual Overlap* ratio in N= 2 agents on Gibson. MANTM is superior to *Mapper w.o. Distance* with over 10% lower *Steps*, suggesting that distance-based heuristics reduce incorrect connections between nodes and provide a more accurate graph for effective global planning. Among all HTP variants, *HTP-Single* has the highest *Mutual Overlap* ratio and *Steps*. This indicates that directly selecting ghost nodes as global goals may lead to a sub-optimal solution due to the large number of node candidates. The performance of *HTP w.o. History* is worse in *Mutual Overlap* ratio, suggesting that the lack of historical memory affects cooperation. *HTP-Concat* shows the lowest training convergence. The result expresses that the concatenation of the main and ghost node features prevents the HTP from better perceiving the relationship between these two types of nodes.

VI. Conclusion and Limitations

We propose *Multi-Agent Topological Neural Mapping* (MANTM), a multi-agent topological exploration framework, to improve exploration efficiency and generalization. In MANTM, the Topological Mapper constructs graphs via a visual encoder and distance-based heuristics. The RL-based Hierarchical Topological Planner (HTP) captures the relationships between agents and graph nodes to infer global goals. Experiments in Habitat demonstrate that MANTM outperforms planning-based baselines and RL variants in unseen scenes. However, there is a huge room for improvement in MANTM. For example, our method leverages metric maps to prune graphs, thus the precision of the metric map may influence the quality of the topological maps. Besides, we assume fully synchronized decision-making, which may fail in environments with communication latency.

References

[1] S. K. Ramakrishnan, D. Jayaraman, and K. Grauman, "An exploration of embodied visual exploration," *International Journal of Computer Vision*, vol. 129, pp. 1616–1649, 2021.

[2] G. Bresson, Z. Alsayed, *et al.*, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[3] A. Kleiner, J. Prediger, *et al.*, "Rfid technology-based exploration and slam for search and rescue," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 4054–4059.

[4] A. Tagliabue, S. Schneider, M. Pavone, and A.-a. Agha-mohammadi, "Shapeshifter: A multi-agent, multi-modal robotic platform for exploration of titan," in *2020 IEEE aerospace conference*. IEEE, 2020, pp. 1–13.

[5] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," *arXiv preprint arXiv:2004.05155*, 2020.

[6] T. Chen, S. Gupta, and A. Gupta, "Learning exploration policies for navigation," *arXiv preprint arXiv:1903.01959*, 2019.

[7] O. Kwon, N. Kim, Y. Choi, H. Yoo, J. Park, and S. Oh, "Visual graph memory with unsupervised representation for visual navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 890–15 899.

[8] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, "Neural topological slam for visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 875–12 884.

[9] J. L. S. Rincon and S. Carpin, "Time-constrained exploration using toposemantic spatial models: A reproducible approach to measurable robotics," *IEEE Robotics & Automation Magazine*, vol. 26, no. 3, pp. 78–87, 2019.

[10] Z. Zhang, J. Yu, J. Tang, *et al.*, "Mr-topomap: Multi-robot exploration based on topological map in communication restricted environment," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 794–10 801, 2022.

[11] E. Tolstaya, J. Paulos, V. Kumar, and A. Ribeiro, "Multi-robot coverage and exploration using spatial graph neural networks," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8944–8950.

[12] H. Zhang, J. Cheng, L. Zhang, Y. Li, and W. Zhang, "H2gnn: hierarchical-hops graph neural networks for multi-robot exploration in unknown environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3435–3442, 2022.

[13] E. Beeching, J. Dibangoye, O. Simonin, and C. Wolf, "Learning to plan with uncertain topological maps," in *European Conference on Computer Vision*. Springer, 2020, pp. 473–490.

[14] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9339–9347.

[15] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial intelligence review*, vol. 43, pp. 55–81, 2015.

[16] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1396–1402.

[17] S. Dong, K. Xu, Q. Zhou, A. Tagliasacchi, S. Xin, M. Nießner, and B. Chen, "Multi-robot collaborative dense scene reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–16, 2019.

[18] A. Quattrini Li, "Exploration and mapping with groups of robots: Recent trends," *Current Robotics Reports*, vol. 1, pp. 227–237, 2020.

[19] A. Bhatia, L. E. Kavraki, and M. Y. Vardi, "Sampling-based motion planning with temporal goals," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2689–2696.

[20] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 413–14 423, 2020.

[21] K. Ye, S. Dong, Q. Fan, H. Wang, L. Yi, F. Xia, J. Wang, and B. Chen, "Multi-robot active mapping via neural bipartite graph matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 839–14 848.

[22] C. Yu, X. Yang, J. Gao, H. Yang, Y. Wang, and Y. Wu, "Learning efficient multi-agent cooperative visual exploration," in *European Conference on Computer Vision*. Springer, 2022, pp. 497–515.

[23] X. Yang, S. Huang, Y. Sun, Y. Yang, C. Yu, W.-W. Tu, H. Yang, and Y. Wang, "Learning graph-enhanced commander-executor for multi-agent navigation," *arXiv preprint arXiv:2302.04094*, 2023.

[24] X. Yang, C. Yu, J. Gao, Y. Wang, and H. Yang, "Save: Spatial-attention visual exploration," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 1356–1360.

[25] W. A. Mackaness and K. M. Beard, "Use of graph theory to support map generalization," *Cartography and Geographic Information Systems*, vol. 20, no. 4, pp. 210–221, 1993.

[26] M. Hahn, D. S. Chaplot, S. Tulsiani, M. Mukadam, J. M. Rehg, and A. Gupta, "No rl, no simulation: Learning to navigate without navigating," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 661–26 673, 2021.

[27] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[28] H. Ryu, H. Shin, and J. Park, "Multi-agent actor-critic with hierarchical graph attention network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 7236–7243.

[29] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts." *proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.

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

[31] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese, "Scene memory transformer for embodied agents in long-horizon tasks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 538–547.

[32] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 611–24 624, 2022.

[33] J. Schulman, F. Wolski, P. Dhariwal, *et al.*, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[34] F. Xia, A. R. Zamir, Z. He, A. Sax, *et al.*, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9068–9079.

[35] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, *et al.*, "Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai," *arXiv preprint arXiv:2109.08238*, 2021.