

MARVEL: Multi-Agent Reinforcement Learning for constrained field-of-View multi-robot Exploration in Large-scale environments

Jimmy Chiun¹, Shizhe Zhang¹, Yizhuo Wang¹, Yuhong Cao¹, Guillaume Sartoretti^{1†}

Abstract—In multi-robot exploration, a team of mobile robot is tasked with efficiently mapping an unknown environments. While most exploration planners assume omnidirectional sensors like LiDAR, this is impractical for small robots such as drones, where lightweight, directional sensors like cameras may be the only option due to payload constraints. These sensors have a constrained field-of-view (FoV), which adds complexity to the exploration problem, requiring not only optimal robot positioning but also sensor orientation during movement. In this work, we propose MARVEL, a neural framework that leverages graph attention networks, together with novel frontiers and orientation features fusion technique, to develop a collaborative, decentralized policy using multi-agent reinforcement learning (MARL) for robots with constrained FoV. To handle the large action space of viewpoints planning, we further introduce a novel information-driven action pruning strategy. MARVEL improves multi-robot coordination and decision-making in challenging large-scale indoor environments, while adapting to various team sizes and sensor configurations (i.e., FoV and sensor range) without additional training. Our extensive evaluation shows that MARVEL’s learned policies exhibit effective coordinated behaviors, outperforming state-of-the-art exploration planners across multiple metrics. We experimentally demonstrate MARVEL’s generalizability in large-scale environments, of up to 90m by 90m, and validate its practical applicability through successful deployment on a team of real drone hardware.

I. INTRODUCTION

Robotic exploration is crucial for autonomous systems to navigate and map unknown environments, with applications ranging from search and rescue [1], [2] to scene reconstruction [3]. Robots rely on on-board sensors for data collection, obstacle avoidance, and real-time decision-making to achieve efficient coverage. Multi-robot exploration, or multi-agent active SLAM [4], leverages multiple robots to enhance efficiency, scalability, and robustness compared to single-agent systems [5]. While most existing approaches use heavy omnidirectional sensors like LiDAR, smaller robots, such as drones, may only be able to carry lightweight, directional sensors (constrained field-of-view (FoV)) like cameras. This paper introduces a framework for multi-robot exploration with constrained FoV sensors, which integrates deep multi-agent reinforcement learning with graph-based attention mechanisms. This framework enhances coordinated decision-making and exploration in complex indoor environments while accommodating diverse sensor configurations.

[†] Corresponding author, to whom correspondence should be addressed.

¹ Authors are with the Department of Mechanical Engineering, College of Design and Engineering, National University of Singapore. {jimmy.chiun, ShizheZhang, wy98,yxian11}@u.nus.edu, {caoyuhong,mpegas}@nus.edu.sg

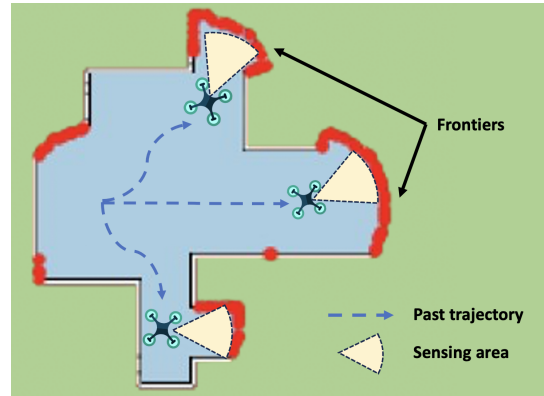


Fig. 1: **Illustration of multi-robot exploration.** 3 drones are collaboratively exploring an indoor environment. The start region is indicated by the yellow star.

The goal of multi-robot exploration is to plan the shortest trajectory to fully map/cover an unknown environment through efficient robot coordination. Challenges include non-myopic decision-making, i.e., the need to optimize long-term information gain rather than immediate rewards, as well as adapting to dynamic environments that require ongoing coordination to effectively cover new areas. Frontier-based approaches are widely used for both single- and multi-agent robotic exploration [6], [7], utilizing heuristics to guide robots toward frontiers—the boundaries between known and unknown areas. These methods aim to balance utility (observable frontiers) and cost (path length) [8], but fall short in optimizing long-term exploration, often resulting in sub-optimal trajectories and inefficiencies due to the revisiting of previously explored regions. This occurs because robots are always oriented towards their goal, which can fragment frontiers into smaller parts, increasing the likelihood of revisitation. Despite their adaptability and lack of training requirements, frontier-based methods frequently demand extensive, complex hyperparameter tuning, which limits their generalization to different environments.

An alternative to frontier-based methods is using reinforcement learning (RL), where agents learn an exploration policy through deep neural networks that process raw sensory data. Once trained, these policies enable robots to perform complex, real-time coordinated actions. However, RL-based exploration faces several challenges: (1) Constrained FoV sensors further complicate viewpoint planning, as restricted visibility amplifies the already vast action space, making decision-making more challenging. (2) Sparse rewards, typically granted when new areas are uncovered or specific goals are achieved, heighten the learning difficulties, as sensor

constraints limit how often frontiers are detected. Without reward shaping or intrinsic motivation, robots face inefficient learning. (3) Reward assignment is also challenging, requiring careful attribution of reward to the viewpoints of individual robots within the team, complicating coordination learning and effective exploration. (4) Addressing both spatial and temporal short-sightedness: robots with constrained FoV must balance exploiting their current viewpoint with exploring unknown regions, while factoring in the long-term consequences of their actions on the overall exploration process.

In this work, we present MARVEL to address multi-robot exploration challenges with constrained FoV. MARVEL uses an attention-based neural network, with intelligent fusion of frontiers and orientation features, for enhanced environmental understanding, as well as ability to handle different spatial scale. Our approach includes an attentive privileged critic network to enhance action estimation and credit assignment. We also introduce a novel information-driven action pruning strategy to manage the large action space by focusing on informative waypoint-heading pairs. Our environment model is a collision-free graph, where nodes represent accessible locations and potential actions guide viewpoint selection. MARVEL, tested in large 2D indoor environments, outperforms existing exploration planners and is validated with drones, showing adaptability to real robot hardware.

II. RELATED WORKS

Conventional Approaches The field of robotic exploration has progressed significantly since Yamauchi’s early work on frontier-based methods, which directed robots toward the nearest unexplored areas by orienting them toward the closest frontiers [6]. This approach was later extended to multi-robot systems using shared global maps [9]. More advanced frontier-based techniques now incorporate gain functions to balance utility and cost when selecting viewpoints for exploration [10], [11]. However, with constrained field-of-view (FoV) sensors, these methods struggle to efficiently evaluate numerous frontiers due to limited visibility.

In response, sampling-based methods have been proposed, leveraging algorithms such as Rapidly-exploring Random Trees (RRT) [8], [12], Rapidly-exploring Random Graphs [13], and Probabilistic Random Maps (PRM) [14]. These techniques reduce computational overhead by evaluating only sampled paths through stochastic processes rather than exhaustively considering all possible viewpoints. However, they perform poorly when informative paths are difficult to sample, especially with constrained FoV. Additionally, methods like Artificial Potential Fields (APF) have been applied to multi-robot exploration by guiding robots toward frontiers based on the occupancy grid and a resistance force for each agent [15]. APF tends to make robots face nearby frontiers, which can lead to inefficient local exploration in constrained FoV scenarios. oronoi-based methods [16] assign exploration partitions to each robot to minimize redundancy but focus on short-term planning, limiting their ability to handle complex multi-agent interactions.

Learning-based approaches Learning-based approaches mainly involve reinforcement learning (RL) as they offer training flexibility and strong expressivity of the environment. Niroui et al.[17] proposed combining frontier-based methods with deep reinforcement learning, and adaptively tuning gain function parameters for frontier selection to enhance exploration performance. Studies by [18], [19] utilized convolutional neural networks (CNNs) in their deep reinforcement learning frameworks. There are also studies that explore incorporating spatial map memory into the network by utilizing a differentiable spatial memory [20], [21]. A notable work for single-agent visual exploration is Active Neural Slam (ANS), where it combines a RL-based global planner with a planning-based local planner [22]. ANS has also been extended with a multi-agent planning module, leveraging a transformer-based architecture. This approach employs hierarchical self-attention mechanisms to capture spatial relationships and interactions between agents [23]. However, all these studies have typically been limited in scope, i.e., they are usually confined to small-scale environments, which lacks complex topologies.

Multi-agent reinforcement learning Multi-agent reinforcement learning (MARL) has shown significant promise in complex cooperative tasks [24], with advancements such as value decomposition [25] aiding in credit assignment and intrinsic rewards [26], [27] addressing sparse rewards. Curriculum learning has also been utilized to progressively increase task complexity [28], [29]. However, optimizing multiple policies in MARL remains challenging compared to single-agent approaches, often requiring domain simplifications like grid worlds or particle simulations [30]. To address these challenges, our work employs the centralized training with decentralized execution (CTDE) paradigm and introduces a multi-agent attentive critic algorithm, inspired by [31]. This approach enhances credit assignment by evaluating each agent’s contribution, thereby improving collaborative policy development for multi-robot visual exploration.

III. PROBLEM FORMULATION

We consider the multi-agent indoor active SLAM problem, where a team of n agents must collaboratively explore an unknown indoor environment in the shortest time. The environment is modeled as a bounded occupancy grid map \mathcal{E} of size $x \times y$, divided into free areas \mathcal{E}_f and occupied areas \mathcal{E}_o , such that $\mathcal{E}_f \cup \mathcal{E}_o = \mathcal{E}$ and $\mathcal{E}_f \cap \mathcal{E}_o = \emptyset$. Each agent i maintains its own belief of the environment \mathcal{B}^i , while their combined belief is denoted as \mathcal{B} , consisting of free areas \mathcal{B}_f , occupied areas \mathcal{B}_o , and unknown areas \mathcal{B}_u , where $\mathcal{B}_f \cup \mathcal{B}_o \cup \mathcal{B}_u = \mathcal{B}$. The known areas are represented by $\mathcal{B}_k = \mathcal{B}_f \cup \mathcal{B}_o$.

Every agent is equipped with a sensor that has a constrained FoV, capturing observations within its line-of-sight range. These sensors update the belief state \mathcal{B}^i of agent i with information from its visible environment, assuming they are effective within a fixed sensing range and FoV. At time t , the sensor observation $\zeta^i(t)$ updates the belief states \mathcal{B}_o , \mathcal{B}_f , and \mathcal{B}_u . For deriving these belief states, we use a 2D sector footprint to approximate the sensor’s constrained

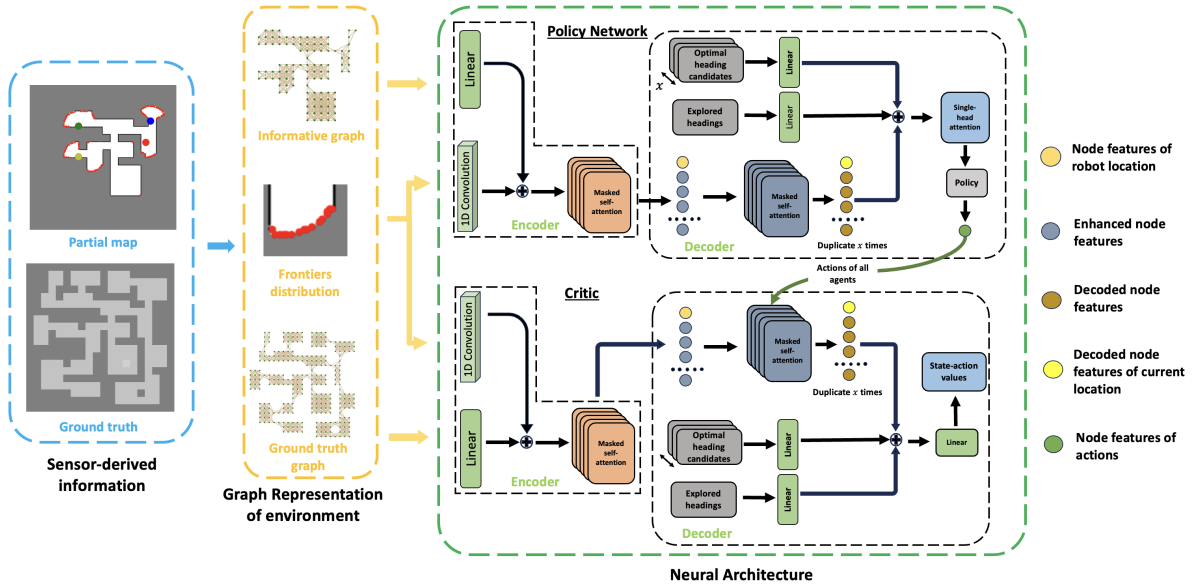


Fig. 2: **MARVEL’s policy and critic network architecture.** We proposed a policy and critic network that leverage on graph-based attention. In the graphs, blue circles indicates the nodes that are connected by edges, indicates as tan lines. We also extract the frontiers (red dots) distribution of each nodes to provide more context to our neural networks.

FoV, similar to [5]. We also assume perfect communication between agents, allowing them to exchange information and maintain a shared map throughout the task.

A. Problem Statement

We task n agents to coordinate their trajectories $\Psi = \{\psi_1, \dots, \psi_n\}$ to complete exploration of the unknown environment in the shortest trajectory. ψ^i represent the trajectory of agent i , where $\psi^i(t)$ represents the agent trajectory at timestep t . The rate of exploration ρ in an environment is defined as the proportion of \mathcal{B}_k to \mathcal{B}_f . In practice, we considered the environment to be fully explored, when ρ reaches a certain threshold, i.e. 0.99. While executing ψ , each individual robot updates its belief \mathcal{B}^i concerning the environment using measurements $\zeta^i(t)$ collected by its onboard constrained FoV sensor. The problem at hand is to determine the optimal set of trajectories, denoted as Ψ^* , where the objective is to minimize the *cost* $C(\psi)$ (e.g., the trajectory length $L(\psi)$ or makespan $T(\psi)$) of the exploration path, i.e., find

$$\Psi^* = \arg \min_{\Psi} \sum_t \max_i C(\psi) \quad (1)$$

Note that we optimize the total maximum distance traveled per step, assuming synchronized decision-making and execution, where all agents are required to reach their viewpoints simultaneously during training.

IV. METHOD

In this section, we formulate multi-robot exploration with constrained FoV as a RL problem and present our proposed attention-based policy and critic neural networks, along with the specifics of our training approach.

A. Multi-robot exploration as a RL problem

Sequential Decision-Making: Multi-robot visual exploration involves a sequential decision-making process where agents make decisions to maximize collective information gain and minimize redundant exploration. At each time step, agents choose actions based on partial observations to plan paths that cover unknown areas efficiently. Each robot’s path is a sequence of viewpoints, denoted as $\psi^i = v_1^i, \dots, v_t^i$, where $v_j^i \in \mathcal{E}_f$ represents the robot’s viewpoint at step j .

We start by uniformly sampling potential viewpoints $V_t = (v_0, v_1, \dots)$, with $v_t = (x_t, y_t) \in \mathcal{B}_f$, and connect them with collision-free edges $E_t = ((v_1, v_2), \dots, (v_{t-1}, v_t))$, forming a graph $G_t = (V_t, E_t)$. At each timestep, agents select actions, appending new nodes to G_t when new areas are discovered. This graph models the environment’s connectivity and navigability. During training, agents build their trajectories $\psi^i(t) = v_1^i, \dots, v_t^i$ as they reach selected viewpoints, and during execution, new viewpoints are computed at regular intervals based on the planning frequency.

Observation: The observation $o_t = (V_t', \Psi_t, F_t)$ includes the informative graph G_t' and the current positions of all agents Ψ_t , which together represent environmental structure and agent distribution. We introduce F_t , a set of frontier distributions around each node in V_t . This distribution is captured by sampling 36 uniformly spaced FoVs around each node and recording the number of observable frontiers within each FoV. We derive $G_t' = (V_t', E_t')$ from G_t to enrich the neural network’s input. Each node $v_k' \in V'$ has the following properties $(\Delta x_{ik}, \Delta y_{ik}, u_k, o_k, g_k, h_k)$: (1) Relative position $(\Delta x_{ik}, \Delta y_{ik})$, the node’s position relative to agent k . (2) Utility u_k , the number of visible frontiers at node v_k , defined by observable frontiers within sensor range d_s and an unobstructed line of sight. (3) Agent occupancy o_k , a ternary signal indicating if v_k is occupied by the current

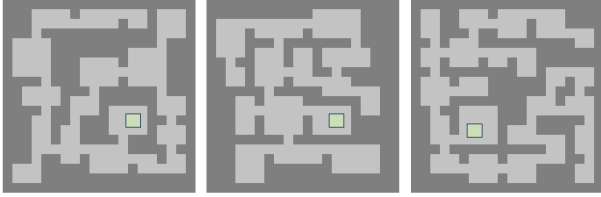


Fig. 3: **Train/Test environments.** Green box: the starting region, which is kept constant for all evaluations conducted.

agent, another robot, or no robot. (4) Guidepost signal g_k , a binary signal indicating if v_k is on the A* path to the nearest frontier. This signal has been demonstrated to significantly improve the agent’s navigation, particularly when the nearest frontier is distant [32]. (5) Informative heading h_k , the orientation maximizing observable frontiers within the FoV. We normalize node coordinates, utility, informative headings, and frontier distributions before inputting them into the policy network, to improve learning dynamics.

Action space: The agent’s action space includes neighboring nodes $(v_j, \psi_i(t)) \in E_t$, with the next state $\psi_i(t+1) = v_j$. To manage the large action space, we use an *action pruning* strategy based on information gain. For each neighboring node, we identify x optimal headings a from observable frontiers, forming joint action pairs $(v_j, a_q) \mid a_q \in a_1, a_2, \dots, a_x$. We set x to 3. When no frontiers are observable, we use the A* path from guidepost g_k to sample headings, aligning with the path if v_j is on it, or pointing to the nearest A* node otherwise. Once agents reach their viewpoints Ψ_t , our graph attention networks, parameterized by θ , generate stochastic policies $\pi_\theta(a_{i,t} \mid o_{i,t})$. Agents then move to their viewpoints, update their maps, and if multiple agents select the same viewpoint, a coordination mechanism reroutes them to the nearest nodes. During training, a motion model ensures smooth rotation and adjusts heading goals as needed. Agents move at 1m/s with a maximum yaw rate of $35^\circ/\text{s}$.

Reward structure: We designed a reward function to optimize the objective in (1), defined as $r_i = r_o + a \cdot r_h + r_t + r_f$. This function incorporates: r_o , the reward based on the number of observable frontiers from the new viewpoint; r_h , the heading reward, calculated as the cosine of the angle deviation from the A* path to the nearest frontier, with a as a scaling factor (set to 0.3); r_t , a team-shared reward for the total number of frontiers observed, where r_o and r_t are normalized by the maximum observable frontiers within the field of view; and r_f , a reward of +10 for task completion when the cumulative utility of nodes reaches 0. This reward structure encourages agents to explore new areas effectively and prioritize task completion while ensuring smooth trajectory planning.

B. Network Architecture

Our policy network (see Fig. 2) utilizes an encoder-decoder architecture with stacked masked attention layers. The encoder captures interdependencies among nodes in G'_t , enhancing node features. These, along with the robot’s current location features, are inputted to the decoder, which

generates policies for joint action of neighboring nodes and their headings. Each attention layer grants nodes access solely to their neighbors’ features.

Encoder: Our encoder leverages a sequence of masked self-attention layers to enhance the feature representations of each node by integrating information from other nodes within the graph. We begin by applying a feed-forward layer to transform the node properties in V'_t into d-dimensional node features h_1^n . We apply 1D convolutions to F_t that encodes the frontier distribution for each node individually to get frontier features h_2^n . This vector captures the node-specific frontier data, and extract local patterns and relationships within the node’s frontier distribution. The resulting node features h_1^n and frontier features h_2^n are concatenated, then projected back to d-dimensional space via a linear layer. These features are then processed through N stacked self-attention layers (with N=6 in our implementation), where each layer’s input is the output of the previous one (i.e., $h^q = h^{k,v} = h^n$). We apply a graph’s mask M_t , computed from the adjacency matrix of E_t , to ensure that each node only attends to features of its neighboring nodes (i.e., attention weight $w_{ij} = 0, \forall (v_i, v_j) \notin E_t$). While attention is restricted to neighboring nodes at each layer, nodes still gather non-neighboring node information by progressively aggregating features across the stacked self-attention layers. In comparison to unmasked self-attention, our approach—where node features propagate layer by layer through neighboring nodes—demonstrates enhanced learning in path-finding tasks. This structured feature propagation allows the encoder to enhance node features h^n into \tilde{h}^n , where each updated set of node features reflects dependencies on other nodes in the graph, influenced by both M_t and the depth N of the attention layers.

Policy decoder: The decoder determines the final policy using the enhanced node features \tilde{h}^n . From these features, we select the node at the robot’s current position as the query: $h^q = h^c = \tilde{h}_{\psi_i(t)}^n$. The current node features h^c , along with neighboring node features from \tilde{h}_e^n , are input into an attention layer. This process refines the output \tilde{h}^c by incorporating information from neighboring nodes. To improve the decoder’s awareness of the local environment, we compute two orientation-specific features for each v_j : We first discretize the heading space into 36 uniform bins, where each bin represents a distinct heading. Then we compute a binary feature vector based on the heading space for each a_q of v_j , indicating whether the heading falls within the FoV centered on a_q . The second is another 36-bin vector that tracks the previously explored headings of v_j . This data is then processed through feed-forward layers to produce the corresponding embedding, h_a^h and h_e^h . Since \tilde{h}^n pertains solely to waypoints and does not include heading information, we replicate each neighboring node feature x times, to generate joint action pairs. Then, we concatenate the expanded \tilde{h}^n with h_a^h and h_e^h and then projected back to d-dimensional space via a feed-forward layer to get heading enhanced neighboring features \tilde{h}_e^n . We then apply

TABLE I: **Comparison with baseline multi-robot planners.** We report the average and standard deviation of the trajectory length to complete exploration, to complete 90% exploration of the environment as well as the overlap ratio. All tests are conducted with 120° FoV and sensor range $d_s = 10m$. The downward arrow indicates that lower values are better.

Agents	Metrics	Nearest	APF	NBVP	Learnt-Greedy	MARVEL
2	Trajectory Length. ↓	637.84(±103.85)	597.91(±123.12)	533.94(±91.63)	665.36(±134.11)	505.25 (±89.03)
	0.90 Coverage ↓	529.49(±96.97)	536.73(±115.03)	440.95(±88.20)	447.32(±106.44)	416.44 (±84.62)
	Overlap Ratio ↓	0.684(±0.097)	0.023 (±0.101)	0.113(±0.212)	0.081(±0.183)	0.048(±0.137)
4	Trajectory Length. ↓	417.92(±89.76)	427.30(±86.46)	416.50(±80.35)	433.03(±96.89)	357.5 (±67.07)
	0.90 Coverage ↓	346.50(±80.11)	358.56(±83.64)	329.75(±81.73)	318.46(±95.15)	294.21 (±62.45)
	Overlap Ratio ↓	0.693(±0.096)	0.075 (±0.143)	0.246(±0.214)	0.191(±0.187)	0.170(±0.169)
8	Trajectory Length. ↓	319.57(±73.66)	311.21(±74.89)	318.99(±59.40)	333.73(±109.30)	279.35 (±50.63)
	0.90 Coverage ↓	266.04(±69.99)	279.76(±65.51)	252.08(±56.19)	256.40(±67.84)	231.54 (±46.40)
	Overlap Ratio ↓	0.671(±0.109)	0.172 (±0.198)	0.401(±0.199)	0.306(±0.177)	0.316(±0.174)

TABLE II: **Adaptability to different sensor’s FoV.** All tests are conducted with 4 agents and sensor range $d_s = 10m$.

FoV	Metrics	Nearest	APF	NBVP	Learnt-Greedy	MARVEL
90°	Trajectory Length. ↓	414.64(±82.28)	472.92(±87.65)	455.28(±82.03)	511.77(±103.13)	378.99 (±57.61)
	0.90 Coverage ↓	347.46(±84.07)	388.45(±80.47)	362.53(±85.70)	366.97(±102.75)	313.67 (±54.57)
	Overlap Ratio ↓	0.594(±0.125)	0.059 (±0.130)	0.203(±0.199)	0.180(±0.181)	0.147(±0.155)
180°	Trajectory Length. ↓	422.80(±108.04)	389.08(±89.53)	380.55(±84.31)	424.74(±105.14)	328.36 (±61.31)
	0.90 Coverage ↓	348.05(±95.57)	320.36(±70.86)	320.88(±83.17)	315.65(±83.23)	273.18 (±59.95)
	Overlap Ratio ↓	0.807(±0.124)	0.098 (±0.175)	0.321(±0.228)	0.216(±0.189)	0.207(±0.183)

a single-head attention mechanism with \tilde{h}^c and its enhanced neighboring features \tilde{h}_e^n , using the resulting attention scores directly as the final output policy $\pi_\theta = (a_{i,t} | o_{i,t}) = w_{i,j}$. This policy guides the agent in selecting the next viewpoint (v_j, a_q) to execute next. This design eliminates the need for a fixed policy size, allowing it to dynamically adjust to the varying number of neighboring nodes.

Critic decoder: We developed our critic network, based on [32], to selectively extract pertinent information for each agent by incorporating other agents’ actions. The critic network Q_ϕ , parameterized by ϕ , utilizes the same encoder structure as the policy network π_θ but features a distinct decoder. During training, the critic receives full ground-truth information, such as the entire map or graph (see Fig. 2), enhancing training stability through *privileged learning*. Using the enhanced node features \tilde{h}^n from the encoder, we retrieve the features of other agents, $\tilde{h}_{\psi_{-i}(t)}^n$, and their corresponding features at the next timestep, $\tilde{h}_{\psi_{-i}(t+1)}^n$, based on their current actions a_t . These are concatenated as key-value pairs to augment the current node feature h^c . The improved current node features \tilde{h}^c are combined with enhanced neighboring features \tilde{h}_e^n , computed similarly to the policy decoder. The improved current node features \tilde{h}^c are combined with enhanced neighboring features \tilde{h}_e^n , computed in the same manner as in the policy decoder. Finally, these concatenated features are mapped to state-action values $Q_\phi(s_t, a_t)$ using a feed-forward layer. The networks are trained using soft actor-critic (SAC) algorithm [33].

Training details: We use a training dataset of 5,663 randomly generated large-s maps (see Fig. 3) and an additional 100 unseen maps for testing. Each environment measures 90m by 90m, with a sensor range of $d_s = 10m$ and a 120° FoV. Training parameters include a maximum episode length

of 128 steps, a discount factor of $\gamma = 1$, a batch size of 256, and an episode buffer size of 10,000. Training starts after accumulating over 2,000 steps in the buffer. We set the target entropy to $0.01 \cdot \log(k)$, perform one iteration per training step after each episode, and use the Adam optimizer with a learning rate of 10^{-5} for both policy and critic networks. The target critic network updates every 256 steps. The complete code and model will be provided upon paper acceptance.

V. EXPERIMENTS

A. Comparison Analysis

Many prior studies evaluate exploration planners in a limited number of scenarios, often fewer than 10. However, we observed that performance can vary significantly across different scenarios, necessitating evaluation across a broad range of environments. Therefore, we test our model and baselines in 100 unseen environments. We compare MARVEL with the following baselines: **Nearest** [9], which selects the nearest frontier as the global goal using a merged map of all agents; **APF** [15], which computes a potential field based on the explored map and agent positions, guiding agents to frontiers via steepest descent, with a resistance force to prevent redundancy. To decrease the computation complexity, frontiers will be gathered into multiple clusters. In every iteration, agents will choose different cluster as target; **NBVP** [8], which samples and evaluates trajectories based on information gain, executing the most optimal one; and **Learnt-Greedy** [34], which employs deep reinforcement learning with graph-based attention to select the next sampling location, adapted for multi-agent systems with greedy headings based on observable frontiers.

We present the average and variance of *trajectory length* of complete exploration, *90% coverage*, and *overlap ratio*

TABLE III: **Adaptability to different sensor ranges.** All tests are conducted with 4 agents and 120° FoV.

Metrics	12m	15m
Trajectory Length \downarrow	299.57(± 60.33)	258.35(± 54.96)
0.90 Coverage \downarrow	247.93(± 58.7)	212.85(± 49.5)
Overlap Ratio \downarrow	0.223(± 0.188)	0.208(± 0.190)

in Table I. The *overlap ratio* measures the shared sensing area among agents per timestep, reflecting collaborative effectiveness. Exploration is considered complete when 99% of the environment is covered. Our results show that MARVEL outperforms all baselines in average trajectory length and achieving 90% coverage, surpassing the best baseline, NBVP, by 14.2% with four agents. MARVEL also demonstrates superior stability with the lowest variance in trajectory metrics, highlighting its strong generalization. Computation times are not included as the baselines were not optimized for efficiency, focusing instead on their core implementations.

In terms of *overlap ratio*, MARVEL ranks second in most tests, with APF emerging as the top performer. This is expected, as APF’s design naturally disperses agents using strong repulsive forces when agents are in close proximity, encouraging them to explore independently. Despite this, MARVEL demonstrates superior multi-agent coordination, excelling in trajectory metrics. In contrast, the Nearest method perform poorly, likely due to their short-sighted strategies, which is more prominent in our large-scale environments. Also, APF’s focus on local exploration further hinders effective agent coordination. NBVP faces challenges in larger environments, and while increasing its sampling rate could improve performance, it comes with high computational costs. In comparison, MARVEL achieves consistent computation times under $0.2s$ per action across all scenarios.

Upon closer inspection, we noticed that learnt-greedy methods delivers faster coverage of 90% of the environment than nearest and APF, indicating the strong performance of graph-based attention networks in learning spatial relationship for exploration tasks. However, this method struggles with small scattered frontiers, produced due to suboptimal viewpoint orientations, leading to repeated revisits and longer trajectories for task completion. As a result, learnt-greedy do not perform well when comparing the trajectory length for task completion. In contrast, MARVEL’s attention-based neural network, with intelligent fusion of frontiers and orientation information, enhances the environmental understanding and provides efficient reasoning about large partial maps, enabling non-myopic viewpoints planning.

B. Adaptability to different sensor configurations

We extensively evaluated our models across various FoV configurations, as detailed in Table II, to understand the impact of sensor parameters on performance. By avoiding re-training with different FoV settings, we isolated performance variations to sensor differences. Our method consistently outperformed all baselines in trajectory metrics, demonstrating robust and efficient exploration planning under varying FoV conditions. Additionally, we tested the model with different

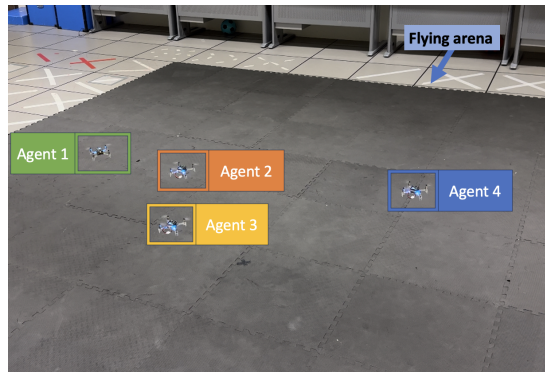


Fig. 4: **Experimental validation on four nano drones.** The grey mat demarcates the 4m by 4m flying arena.

sensor ranges, as shown in Table III. Originally trained with $d_s = 10m$, the model successfully utilized extended ranges of $d_s = 12m$ and $15m$, showing significant performance improvements. This adaptability without retraining underscores the model’s resilience to sensor configuration changes and its ability to leverage additional sensory data effectively, making it suitable for diverse environments.

C. Experimental Validation

To assess MARVEL’s potential for real-world deployment, we conducted experiments with Crazyflie 2.1 drones in a 4m by 4m arena (see Fig. 4). Using the swarm manager from [35], we simulated sensor coverage based on the drones’ real-time positions, enabling live to virtual mapping tests to be conducted. These trials confirmed that MARVEL can be successfully implemented on physical hardware and adapts well to the motion dynamics of the drones.

VI. CONCLUSION

In this work, we introduce a neural framework tailored for multi-robot exploration with constrained field-of-view (FoV) sensors in large-scale environments. Our approach employs a graph attention network, with intelligent fusion of frontier and orientation features, to enhance environmental understanding and allow agents to produce non-myopic decisions. In particular, MARVEL handles the extensive action space of viewpoint planning using an information-driven action pruning method. Our evaluations show that MARVEL outperforms existing state-of-the-art multi-agent exploration planners, adapts well to different team sizes and FoV configurations, and performs reliably across diverse environments. Additionally, MARVEL has been successfully validated on real drones, demonstrating its potential for deployment on actual robots.

While MARVEL excels in 2D indoor settings, we acknowledge the added complexity of 3D environments with intricate obstacles. To address this, we plan to extend MARVEL to handle full 3D action spaces by integrating height information and associated FoV changes, thereby enhancing its application on platforms such as drones for indoor mapping. There, to ensure efficient 3D exploration, we plan to employ a sparse graph representation to manage computational demands effectively.

REFERENCES

- [1] P. Petracek, V. Kratky, M. Petrlik, T. Baca, R. Kratochvil, and M. Saska, "Large-scale exploration of cave environments by unmanned aerial vehicles," vol. 6, no. 4, pp. 7596–7603. [Online]. Available: <https://ieeexplore.ieee.org/document/9492802/>
- [2] A. Kleiner, J. Prediger, and B. Nebel, "RFID technology-based exploration and SLAM for search and rescue," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 4054–4059. [Online]. Available: <http://ieeexplore.ieee.org/document/4059043/>
- [3] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3d reconstruction," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3477–3484. [Online]. Available: <http://ieeexplore.ieee.org/document/7487527/>
- [4] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, and J. A. Castellanos, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1686–1705, 2023.
- [5] B. Zhou, H. Xu, and S. Shen, "Racer: Rapid collaborative exploration with a decentralized multi-uav system," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1816–1835, 2023.
- [6] B. 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'*, 1997, pp. 146–151.
- [7] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, 2000, pp. 476–481 vol.1.
- [8] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1462–1468.
- [9] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the Second International Conference on Autonomous Agents*, ser. AGENTS '98. New York, NY, USA: Association for Computing Machinery, 1998, p. 47–53. [Online]. Available: <https://doi.org/10.1145/280765.280773>
- [10] M. Juliá, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," vol. 33, no. 4, pp. 427–444. [Online]. Available: <http://link.springer.com/10.1007/s10514-012-9298-8>
- [11] M. Kulich, J. Faigl, and L. Přeučil, "On distance utility in the exploration task," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4455–4460.
- [12] A. N. Nazif, A. Davoodi, and P. Pasquier, *Multi-Agent Area Coverage Using a Single Query Roadmap: A Swarm Intelligence Approach*, 2011, pp. 95–112.
- [13] T. Dang, M. Tranzatto, S. Khattak, F. Mascarih, K. Alexis, and M. Hutter, "Graph-based subterranean exploration path planning using aerial and legged robots," *Journal of Field Robotics*, vol. 37, no. 8, pp. 1363–1388, 2020.
- [14] Z. Xu, D. Deng, and K. Shimada, "Autonomous uav exploration of dynamic environments via incremental sampling and probabilistic roadmap," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2729–2736, 2021.
- [15] J. Yu, J. Tong, Y. Xu, Z. Xu, H. Dong, T. Yang, and Y. Wang, "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)*, 2021, pp. 8779–8785.
- [16] 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. 14413–14423, 2020.
- [17] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.
- [18] D. Zhu, T. Li, D. Ho, C. Wang, and M. Q.-H. Meng, "Deep reinforcement learning supervised autonomous exploration in office environments," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 7548–7555.
- [19] H. Li, Q. Zhang, and D. Zhao, "Deep reinforcement learning-based automatic exploration for navigation in unknown environment," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 6, pp. 2064–2076, 2019.
- [20] A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid, and J. Davidson, "Visual representations for semantic target driven navigation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8846–8852.
- [21] J. F. Henriques and A. Vedaldi, "Mapnet: An allocentric spatial memory for mapping environments," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8476–8484.
- [22] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," in *International Conference on Learning Representations (ICLR)*, 2020.
- [23] C. Yu, X. Yang, J. Gao, H. Yang, Y. Wang, and Y. Wu, "Learning efficient multi-agent cooperative visual exploration," in *Computer Vision – ECCV 2022*. Cham: Springer Nature Switzerland, 2022, pp. 497–515.
- [24] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey," vol. 55, no. 2, pp. 895–943. [Online]. Available: <https://link.springer.com/10.1007/s10462-021-09996-w>
- [25] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '18. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2018, p. 2085–2087.
- [26] I.-J. Liu, U. Jain, R. A. Yeh, and A. Schwing, "Cooperative exploration for multi-agent deep reinforcement learning," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 6826–6836. [Online]. Available: <https://proceedings.mlr.press/v139/liu21j.html>
- [27] S. Iqbal and F. Sha, "Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning," 2021. [Online]. Available: <https://arxiv.org/abs/1905.12127>
- [28] Q. Long, Z. Zhou, A. Gupta, F. Fang, Y. Wu, and X. Wang, "Evolutionary population curriculum for scaling multi-agent reinforcement learning," in *International Conference on Learning Representations*, 2020.
- [29] W. Wang, T. Yang, Y. Liu, J. Hao, X. Hao, Y. Hu, Y. Chen, C. Fan, and Y. Gao, "From few to more: Large-scale dynamic multiagent curriculum learning," vol. 34, no. 5, pp. 7293–7300. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/6221>
- [30] C. Wakiipoor, P. J. Martin, C. Rebhuhn, and A. Vu, "Heterogeneous multi-agent reinforcement learning for unknown environment mapping," 2020. [Online]. Available: <https://arxiv.org/abs/2010.02663>
- [31] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 09–15 Jun 2019, pp. 2961–2970. [Online]. Available: <https://proceedings.mlr.press/v97/iqbal19a.html>
- [32] Y. Wang, Y. Cao, J. Chiun, S. Koley, M. Pham, and G. A. Sartoretti, "Viper: Visibility-based pursuit-evasion via reinforcement learning," in *8th Annual Conference on Robot Learning*.
- [33] P. Christodoulou, "Soft actor-critic for discrete action settings," 2019. [Online]. Available: <https://arxiv.org/abs/1910.07207>
- [34] Y. Cao, R. Zhao, Y. Wang, B. Xiang, and G. Sartoretti, "Deep reinforcement learning-based large-scale robot exploration," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4631–4638, 2024.
- [35] J. Chiun, Y. R. Tan, Y. Cao, J. Tan, and G. Sartoretti, "Star: Swarm technology for aerial robotics research," *arXiv preprint arXiv:2406.16671*, 2024.