

Efficient Multi-Agent Exploration in Area Coverage Under Spatial and Resource Constraints

Maram Hasan^a and Rajdeep Niyogi^b

Indian Institute of Technology Roorkee, Roorkee, 247667, India
{mhasan1, rajdeep.niyogi}@cs.iitr.ac.in

Keywords: Multi-Agent Reinforcement Learning, Curiosity-Based Rewards, Exploration, Coverage Path Planning, Constrained Environments.

Abstract: Efficient exploration in multi-agent Coverage Path Planning (CPP) is challenging due to spatial, resource, and communication constraints. Traditional reinforcement learning methods often struggle with agent coordination and effective policy learning in such constrained environments. This paper presents a novel end-to-end multi-agent reinforcement learning (MARL) framework for area coverage tasks, leveraging the centralized training and decentralized execution (CTDE) paradigm with enriched tensor-based observations and curiosity-based intrinsic rewards, which encourage agents to explore under-visited regions, enhancing coverage efficiency and learning performance. Additionally, prioritized experience adaptation accelerates convergence by focusing on the most informative experiences, improving policy robustness. By integrating these components, the proposed framework facilitates adaptive exploration while adhering to the spatial, resource, and operational constraints inherent in CPP tasks. Experimental results demonstrate superior performance over traditional approaches in coverage tasks under variable configurations.


1 INTRODUCTION


Multi-agent systems (MAS) are increasingly applied in diverse domains that require coordinated operations across complex environments. These systems enable agents to collaboratively achieve tasks demanding extensive spatial coverage, adaptability, and efficient information gathering, often exceeding the capabilities of individual agents. One prominent application is Coverage Path Planning (CPP), where agents develop optimal routes to ensure thorough area coverage while minimizing gaps and overlapping (Tan et al., 2021). The main goal of CPP is to ensure that every location within an environment is visited at least once, while adhering to optimization constraints (Orr and Dutta, 2023). It has become indispensable in fields like autonomous cleaning, precision agriculture, space exploration, and search-and-rescue operations, where systematic coverage is essential for operational efficiency and high-quality outcomes (Yanguas-Rojas and Mojica-Nava, 2017).

Implementing efficient multi-agent coordination in CPP poses significant challenges, particularly due

to spatial and operational constraints, resource limitations, and communication challenges. These factors shape how agents navigate and interact to achieve the goal of full area coverage while minimizing resource consumption and optimizing efficiency. In structured environments, such as warehouses, spatial constraints require agents to operate within physical defined boundaries like walls and other structural elements, necessitating precise navigation and strategic path planning. Additionally, they encompass coverage completeness, ensuring all regions within the environment are visited at least once with minimal overlap, requiring paths that avoid redundancy.

Resource efficiency is equally critical in CPP, especially in mission-critical scenarios, where agents must minimize energy consumption and time by selecting paths that achieve full coverage while avoiding unnecessary detours or delays (Ghaddar and Merei, 2020). Furthermore, dynamic obstacles, such as human workers, other robots, or moving machinery, introduce further unpredictability. Agents must continuously adapt their paths in real time to avoid collisions, recalibrating routes in response to these obstacles to ensure safety and operational effectiveness. Together, spatial constraints, resource efficiency, and dynamic obstacles create a challenging environment

^a  <https://orcid.org/0000-0001-9040-5842>

^b  <https://orcid.org/0000-0003-1664-4882>

for traditional path planning, requiring innovative approaches to improve adaptability and efficiency.

In recent years, reinforcement learning (RL) has emerged as a promising solution for dynamic robotic decision-making. It enables agents to learn behaviors through trial-and-error interactions with the environment rather than relying on explicit manual programming. Advancements in multi-agent reinforcement learning (MARL) extend this capability, offering robust solutions to tackle diverse challenges by allowing multiple agents to collaborate effectively, adapt to environmental changes in real time, and achieve coordinated coverage through learning-based approaches.

Multi-agent exploration in area coverage using reinforcement learning algorithms can be categorized into end-to-end and two-stage approaches (Garaffa et al., 2021). End-to-end methods treat exploration as a unified process, where raw or processed sensor data is input directly into an RL policy, which generates control actions for the agent (Chen et al., 2019b). This approach entrusts RL with all aspects of the exploration task. In two-stage approaches, RL is integrated with conventional methods by dividing decision-making into distinct components. One usage involves RL determining target locations, with classical algorithms like Dijkstra or A* (Stentz, 1994) handling path planning independently. Another applies RL exclusively to path planning, where partitioning algorithms such as dynamic Voronoi assign targets, leaving RL to navigate to these destinations (Hu et al., 2020). A third variation employs separate RL models for target selection and path planning in a layered structure, enabling agents to address intricate exploration tasks at the cost of higher computational overhead (Jin et al., 2019).

Our work focuses on developing an advanced MARL framework that addresses these spatial and resource constraints, as well as communication limitations, to enhance exploration and coverage in multi-agent systems. By integrating enriched state representations, intrinsic motivation and prioritized experience adaptation, we aim to enhance agents' exploration and learning efficiency under these constraints.

2 RELATED WORK

2.1 Classical Optimization Methods

Classical optimization and heuristic methods have laid the foundation for coverage path planning (CPP) in multi-agent systems especially given the NP-hard nature of this problem (Chen et al., 2019a). Frontier-based exploration (Yamauchi, 1997), a systematic

spatial exploration method, involves agents identifying the boundary between explored and unexplored areas, known as frontiers, and move toward these regions to maximize coverage. Cooperative frontier-based strategies extend this approach by enabling agents to share information and coordinate movements, thereby reducing redundant exploration and improving execution efficiency (Burgard et al., 2005).

Sweeping-based methods enable agents to systematically cover areas in coordinated patterns, typically moving in parallel or predefined formations to ensure comprehensive coverage with minimal overlap (Tran et al., 2022). These approaches are effective in both communication-enabled and communication-free scenarios, maximizing coverage while minimizing redundancies (Sanghvi et al., 2024). Meanwhile, biologically inspired swarming algorithms leverage local interaction rules to achieve complex and stable coordinated behaviors (Gazi and Passino, 2004). Building on this, decentralized swarm-based approaches have been developed for dynamic coverage control, allowing agents to adapt to environmental changes (Atinç et al., 2020). This adaptability proves particularly valuable for tasks requiring real-time re-allocation of coverage areas (Khamis et al., 2015).

Classical methods, while useful, are limited in dynamic and complex environments due to their reliance on static rules. They struggle with redundant coverage, limited adaptability to dynamic obstacles and unexpected change, and coordination challenges as agent numbers increase. These limitations highlight the need for learning-based approaches that enable autonomous adaptation, improved coordination, and effective handling of multi-agent coverage tasks.

2.2 Learning-Based Methods

Recently, learning methods have been increasingly applied to coverage path planning tasks, with reinforcement learning enabling agents to learn and adapt autonomously in complex environments (Zhelo et al., 2018). Early studies focused on single-agent RL, such as the application of Double Deep Q-Network (DDQN) to train individual agents in a simulated grid-world environment (Li et al., 2022), improving navigation without explicit inter-agent coordination. Meanwhile, centralized approaches like (Jin et al., 2019) combine deep Q-networks (DQN) for target selection with DDPG for adjusting agents' rotations, facilitating coordinated movements through centralized target selection. These approaches faced scalability challenges as the number of agents increased.

Traditional reinforcement learning have been extended to multi-agent settings, enabling agents to

collaboratively learn strategies that enhance coverage and adaptability in real time. End-to-end MARL algorithms such as Multi-agent Proximal Policy Optimization (Chen et al., 2019b), directly learn coordinated exploration strategies from raw sensory data leveraging convolutional neural networks to process multi-channel visual inputs. Similarly, a QMIX-based algorithm with a modified loss function and sequential action masking (Choi et al., 2022) has been applied to improve coordination among Automated Guided Vehicles in cooperative path planning. Two-stage MARL approaches divide tasks into high-level decision-making and low-level execution layers. For instance, hierarchical cooperative exploration (Hu et al., 2020) uses dynamic Voronoi partitioning to assign unique exploration areas to agents, while (Setyawan et al., 2022) employs two levels of hierarchies with Multi-agent Deep Deterministic Policy Gradient (MADDPG) at each layer for effective coordination in multi-agent coverage tasks.

On another note, exploration in multi-agent reinforcement learning is critical for efficient learning and faster convergence, particularly in complex environments where traditional methods like epsilon-greedy or noise-driven exploration fall short. Alternative methods have been proposed such as employing graph neural networks in (Zhang et al., 2022) for coarse-to-fine exploration, while a combination of DQNs and graph convolutional networks was utilized in (Luo et al.,) for sequential node exploration on topological maps. Furthermore, curiosity-based intrinsic reward mechanisms have emerged as a promising technique for exploration where it was integrated with the asynchronous advantage actor-critic (A3C) algorithm to enable effective mapless navigation in single-agent systems (Zhelo et al., 2018), demonstrating significant improvements over traditional exploration methods. Therefore, we are motivated to incorporate a curiosity-based intrinsic reward mechanism to enhance exploration in multi-agent learning.

Our proposed end-to-end MARL architecture extends MADDPG (Lowe et al., 2017) to facilitate adaptive exploration that adheres to spatial and resource constraints. By leveraging intrinsic rewards, enriched state representations, and priority buffer adaptation, it effectively addresses these challenges and achieves superior performance in complex coverage tasks.

3 PROBLEM DEFINITION

We consider the task of coverage path planning where the goal is to effectively and fully cover a given environment while reducing resource consumption

and redundancy. The desired path should (a) visit all previously unvisited locations to ensure complete coverage, (b) effectively navigates around obstacles, and (c) minimizes the total operational time. Each agent operates autonomously under the following constraints: *limited perception range*, *constrained battery capacity*, and *dynamic obstacle avoidance*.

We consider a structured indoor environment E , such as a warehouse or industrial facility, characterized by obstacles that influence agent movement. The environment has dimensions $M \times N$, where M and N denote the length and width, respectively. Specific environment attributes, such as obstacle count, location, and size, are generated based on a predefined probability distribution \mathcal{P} . In the environment E , we define I a set of k agents, with a specified diameter d_a . The environment is discretized into cells of size d_a , where each cell represents a fixed spatial unit. Figure 1 shows the discretization process of the multi-agent environment with dynamic and static obstacles.

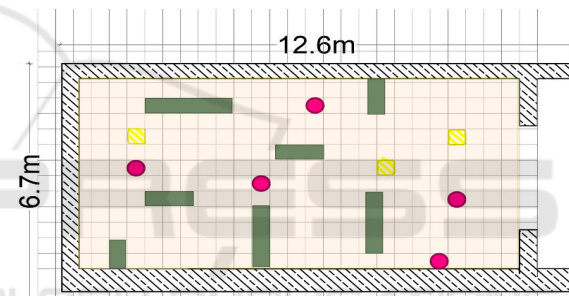


Figure 1: A discretized indoor structured environment, featuring a team of five agents (purple circles) navigating dynamic and static obstacles (yellow and grey, respectively) to achieve efficient exploration and coverage.

3.1 Problem Formalization

In our cooperative multi-agent coverage path planning environment, each agent operates with a policy guided solely by local observations rather than a global state s , which remains unknown to all agents. To capture this limited information access, we formalize the problem using a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) framework. The objective is to learn a joint policy $\pi = \{\pi_1, \pi_2, \dots, \pi_k\}$, comprising individual policies that collectively optimize the cumulative discounted rewards over a defined planning horizon h .

Starting at an initial state s_0 , at each time step t , the environment has a global state $s_t \in S$, each agent $i \in I$ selects actions based solely on its own local observations $o_i \in O_i$. Each agent i interacts independently within its observation space O_i , choosing ac-

tions according to its policy π_i . After executing the joint action $a_t = (a_t^1, a_t^2, \dots, a_t^k) \in A$, where A denotes the joint action space, the environment transitions to the next state $s_{t+1} \in S$ as per a transition probability $T(s_{t+1}|s_t, a_t)$. Each agent receives a reward reflecting quality of action (section 5.1). Dec-POMDP embodies a fully decentralized structure, where agents independently execute actions and receive local rewards based on their actions and observations.

4 METHODOLOGY

4.1 Multi-Agent Deep Deterministic Policy Gradient (MADDPG)

We extend MADDPG (Lowe et al., 2017) to learn efficient policies for coverage path planning. While MADDPG inherently supports continuous action spaces, our framework employs a posteriori discretization using a grid-based approach to facilitate coverage tracking and reward computation. During training, the centralized critic accesses the global state and actions of all agents, capturing inter-agent dependencies. However, decentralized execution enables agents to act independently based on local observations, ensuring scalability and adaptability. While centralized approaches are computationally demanding and impractical for real-time execution, fully decentralized methods often result in suboptimal performance due to limited awareness of global context. MADDPG addresses these challenges by employing a centralized critic during training and decentralized actors for execution. Figure 2 shows the components and interactions of the proposed framework.

The Actor Network π_{θ_i} : Each agent i selects action $a_i = \pi_{\theta_i}(o_i)$ based on local observations o_i , thereby facilitating decentralized decision-making. The objective of the actor network is to learn a deterministic policy π_{θ_i} that maximizes the expected cumulative reward by minimizing the following loss function L_{π_i} :

$$L_{\pi_i} = -\mathbb{E}_{o_i \sim D} [Q_{\phi_i}(s, \pi_{\theta_i}(o_i), a_{-i})],$$

where D is the replay buffer storing past experiences for sampling, Q_{ϕ_i} is the centralized critic’s estimate of the Q-value, and a_{-i} denotes the actions taken by all agents except agent i . This loss function encourages each actor to take actions that maximize the centralized Q-value estimates, thereby aligning the individual agent’s actions with the overall system’s objective.

The Critic Network Q_{ϕ_i} : It evaluates the quality of the joint actions by estimating the expected cumula-

tive reward for a given state-action pair. It takes as input the full state s (global information) and the joint action vector $\mathbf{a} = (a_1, a_2, \dots, a_N)$ and produces as output the estimated Q-value $Q_{\phi_i}(s, \mathbf{a})$. The critic minimizes the temporal difference (TD) error L_{Q_i} :

$$L_{Q_i} = \mathbb{E}_{(s, \mathbf{a}, r, s') \sim D} [(Q_{\phi_i}(s, \mathbf{a}) - y)^2]$$

where the target value y is defined as:

$$y = r_i + \gamma \mathbb{E}_{\mathbf{a}' \sim \pi} [Q_{\phi_i}(s', \mathbf{a}')]]$$

Here, r_i is the agent’s local reward, γ is the discount factor that balances the influence of future and immediate rewards, and s' represents the next state after taking the joint action. Minimizing TD error improves Q-value estimation accuracy, providing meaningful feedback for actor training. This feedback ensures the development of strategies that enhance overall system performance.

4.2 Curiosity-Based Exploration

The curiosity-driven mechanism further complements this framework by incentivizing agents to explore unvisited areas, overcoming the limitations of local observations in decentralized execution. Traditional exploration strategies, such as epsilon-greedy or noise-driven methods, often prove insufficient for environments that require comprehensive coverage and thorough exploration, as they may fail to guide agents effectively through obstacles and complex layouts.

The curiosity reward is learned using a self-supervised approach, driven by the prediction error between the agent’s anticipated state features and the actual observed features. Specifically, it uses a feature extraction network f_{ψ} parameterized by ψ , to process each state s and generate a feature vector $\phi(s) = f_{\psi}(s)$ focusing on relevant aspects of the state while reducing complexity. Next, the extracted features serve as input to a forward dynamics model \hat{f}_{θ} which predicts the next state’s feature vector based on the current state and action $\hat{\phi}(s') = \hat{f}_{\theta}(\phi(s), a)$.

The intrinsic reward $r_{\text{curiosity}}$ is then computed as the prediction error between the predicted and the actual observed feature vector $\hat{\phi}(s')$:

$$R_{\text{curiosity}} = \|\phi(s') - \hat{\phi}(s')\|^2$$

where $\|\cdot\|$ denotes the Euclidean norm. This mechanism encourages agents to explore novel states where the prediction error is high, effectively guiding them toward under-explored areas. By integrating this curiosity reward with extrinsic rewards defined by the environment, the actor network learns policies that balance exploration and exploitation. The centralized critic further refines these policies by incorporating

global state information during training, ensuring the learned strategies align with the overall objective.

4.3 Tensor-Like State Representation

For each agent, the observation is represented as a multi-layered tensor to enhance spatial awareness and environmental understanding. This representation comprises three channels: an occupancy map that indicating agent positions and dynamic obstacles, an obstacle map highlighting static obstacles, and a visitation map recording the frequency of cell visits. By structuring these features in an image-like format, agents gain spatial context, allowing them to distinguish between frequently and infrequently visited regions. The multi-layered configuration facilitates the use of convolutional processing, enabling agents to leverage spatial patterns more effectively, and ultimately implement more efficient coverage strategies.

4.4 Priority Buffer Adaptation

In reinforcement learning, standard experience replay buffers employ random sampling assigning equal probability to all experiences, treating them as equally valuable for agent’s learning process. While effective in simpler tasks, this approach can delay learning in complex multi-agent environments where experiences vary significantly in their impact on exploration and coordination strategies. To address this, we implement a Prioritized Experience Replay (PER) buffer (Schaul, 2015) that assigns higher sampling probabilities to more informative experiences, such as successful exploration steps or collisions, over repetitive navigation. Experiences with larger temporal-difference (TD) errors where predictions diverge most from latest observed outcomes are prioritized, accelerating convergence in challenging tasks.

The priority p_i of an experience i is computed as:

$$p_i = (|\delta_i| + \epsilon)^\alpha$$

where δ_i is the TD error, ϵ is a small constant ensures non-zero priority, and $\alpha \in [0, 1]$ controls the level of prioritization, the higher values the more focus on high-error experiences. The TD error δ_i is defined as:

$$\delta_i = r + \gamma Q(s', a'; \theta') - Q(s, a; \theta)$$

where r is the reward, s and a are the state-action pair, s' and a' are the next state and action, γ is the discount factor, and Q represents the action-value function parameterized by θ (current) and θ' (target network). The sampling probability from the prioritized buffer for experience i is computed as:

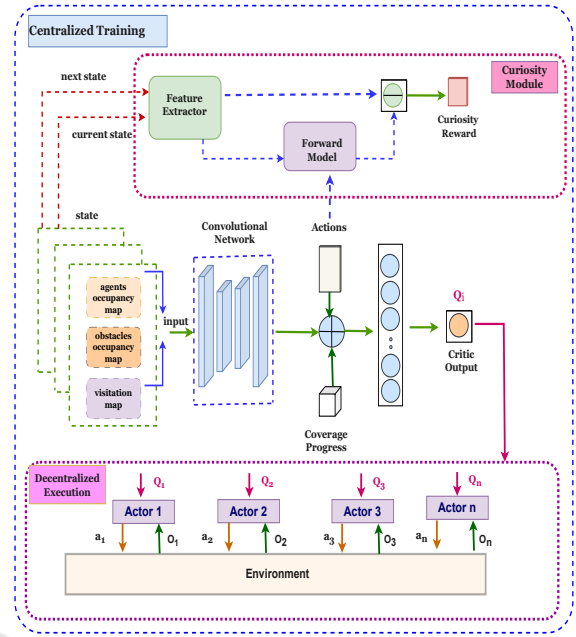


Figure 2: Our framework architecture showing the critic and actors interactions, and the curiosity module.

$$P(i) = \frac{p_i}{\sum_j p_j}$$

To correct for sampling bias introduced by prioritization, importance-sampling (IS) weights are used as :

$$w_i = \left(\frac{1}{N \cdot P(i)} \right)^\beta$$

where N is the total number of experiences, and $\beta \in [0, 1]$ gradually increases to 1 to reduce bias as learning progresses. These IS weights adjust the loss, ensuring unbiased gradient updates. By prioritizing experiences with greater informational value, the buffer enhances the model’s ability to learn robust policies and accelerates convergence, particularly in environments where specific state-action pairs have a disproportionate impact on overall performance.

5 EXPERIMENTS

5.1 Task Description

In this coverage problem, we consider an environment that simulates a real-world warehouse area where agents move efficiently to ensure complete coverage. This coverage may include inspection, surveillance, or cleaning operations across the warehouse. The warehouse layout is converted into a grid map. Transforming the continuous space and actions of the environment into discretized equivalents, as shown in

Figure 1. Each grid cell (i, j) represents a unique spatial location that requires coverage, with a fixed size equal to the agent diameter d_a , for example, $(0.5 \text{ m} \times 0.5 \text{ m})$. The dynamic obstacles within the grid introduce complexity by changing its positions over time.

Agent Configuration. Each agent a_k in the environment has the following characteristics:

- **Battery Capacity B_k :** Maximum energy the agent can utilize before recharging is required.
- **Perception Range P_k :** Fixed distance within which the agent observes nearby obstacles, agents, and grid cells.
- **Neighborhood Set $\mathcal{N}_k(t)$:** All observable entities within agent's perception range P_k , including dynamic and static obstacles and other agents.

Observation Space. While the centralized critic has full-observability of the environment, the Actor for each agent operates with partial observability, relying only on localized information rather than complete global awareness. Each agent is limited to its perception range, able to observe information within its $z \times z$ perception window centered around a_k 's position. For example, within a 5×5 grid, the agent observes obstacles, other agents, cell coverage status within this window. Agents occupy one grid cell at a time and can move to adjacent cells if they are valid cells. Figure 3 illustrates the observations accessible to agents within their perception range across various scenarios of a discretized environment.

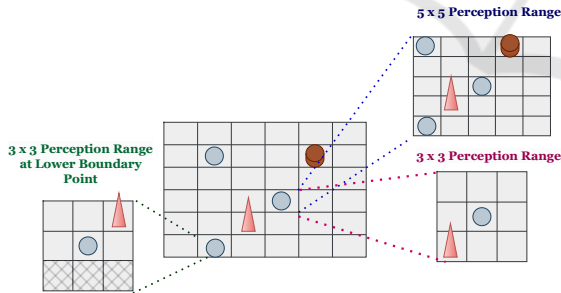


Figure 3: The observations and perception range of each agent in a discretized environment.

Action Space. Each agent has a discrete action space A_i , representing the set of all possible actions it can take in the environment. This includes the primitive actions of moving in the standard 2D directions, as well as the NoOp (No Operation) action, used when no other action is appropriate or necessary.

$$A_i = \{Up, Down, Left, Right, NoOp\}$$

At each time t , agent a_k selects an action to navigate the grid. All agents share the same travel velocity for the purpose of simplicity.

Dynamics. State transitions are influenced by the actions of all agents and the movement of dynamic obstacles. The state $s(t+1)$ depends on the joint action $\{a_k(t)\}$ for all agents and the obstacle dynamics. Agents experience transitions to specific cells based on their chosen actions.

Action Masking. Agents face significant challenges in avoiding collisions, such as moving into walls or static obstacles. To address this, we employ an action masking mechanism that preemptively restricts infeasible actions, ensuring safer navigation and minimizing uninformative experiences that could hinder training. However, collisions with dynamic obstacles are managed separately through the reward mechanism.

Reward Signal. Our environment operates under dense rewards setting, where each agent receives reward signal at each timestep. This signal encourages efficient coverage by rewarding agents for covering new cells, penalizing multiple visits to already-covered cells, and discouraging excessive energy usage. Thus, incentivizing agents to prioritize uncovering new cells until full coverage is achieved.

- **Coverage Reward R_{cover} :** A positive reward for covering an uncovered cell (i, j) where $V_{tot}(i, j)$, the total visitation count of cell (i, j) across all agents, is equal to zero. This encourages agents to prioritize new areas for coverage.

$$R_{cover} = \begin{cases} +10 & \text{if } V_{tot}(i, j) = 0, \text{ (cell is uncovered)} \\ 0 & \text{otherwise} \end{cases}$$

- **Overlapping Penalty $R_{overlap}$:** A negative penalty is applied when an agent visits a cell (i, j) already covered by any agent, with λ controlling the severity. This discourages frequent overlaps, encouraging agents to minimize revisits to previously covered areas:

$$R_{overlap} = \begin{cases} -\min(0.5 \cdot V_{tot}(i, j)^{1.5}, 5) & \text{if } (i, j) \text{ is covered} \\ 0 & \text{otherwise.} \end{cases}$$

- **Energy-Aware Penalty R_{energy} :** To account for resource constraints, a negative penalty proportional to energy usage is applied to promote efficient path choices. The energy consumption $E_k(t)$ for agent a_k is constrained by:

$$\sum_{t=0}^T E_k(t) \leq B_k, \quad \forall k \in I$$

where $E_k(t)$ denotes the total energy used along its path up to timestep t , based on movement and turns. The penalty R_{energy} is defined as:

$$R_{\text{energy}} = \begin{cases} -(\lambda_1 \cdot d + \lambda_2 \cdot \theta) & \text{if } a_k \text{ moves or turns} \\ -\lambda_3 & \text{if } a_k \text{ stays NoOP} \end{cases}$$

Here, d is the per-timestep distance traveled, θ is the turning angle, and constants λ_1 , λ_2 and λ_3 represent the energy consumed per unit distance, per degree of turn, and for NoOp actions, respectively. A small penalty λ_3 encourages agents to move instead of getting stuck. For this work, we use $\lambda_1 = 0.11$, $\lambda_2 = 0$ and $\lambda_3 = 0.15$ as rotation actions are not included in the action space.

- **Collision Penalty $R_{\text{collision}}$:** A penalty is applied when the distance between agent a_k and any obstacle within its perception range $\text{obs} \in O_k$ falls below a predefined margin d_{margin} , or if the agent's new position is occupied by other agents or obstacles. This mechanism encourages agents to avoid collisions and maintain safe navigation.

$$R_{\text{collision}} = \begin{cases} -1 & \text{if } d_{\text{Manhattan}}(a_k, \text{obs}) \leq d_{\text{margin}} \\ -10 & \text{if } (x_k(t+1), y_k(t+1)) \text{ occupied} \\ 0 & \text{otherwise} \end{cases}$$

where $d_{\text{Manhattan}}(a_k, \text{obs}) = |x_k - x_{\text{obs}}| + |y_k - y_{\text{obs}}|$ represents the manhattan distance between the agent and any obstacle in its perception range.

The cumulative reward $R_k(t)$ for agent a_k at each timestep t is defined as:

$$R_k(t) = R_{\text{cover}} + R_{\text{overlap}} + R_{\text{energy}} + R_{\text{collision}} \quad (1)$$

Finally, the total reward signal incorporates the curiosity-based intrinsic reward to promote efficient exploration during training.

$$R_{\text{total}} = R_k(t) + \alpha_{\text{curiosity}} \cdot R_{\text{curiosity}} \quad (2)$$

The weight, $\alpha_{\text{curiosity}} \in [0, 1]$, controls the contribution of the curiosity reward to the final reward signal.

5.2 Implementation Details

In this section, we detail the implementation setup of our experiments, focusing on the application of our framework in a warehouse environment. The experiments were conducted on two discretized environments of sizes 10×10 and 20×20 representing a $5m \times 5m$ and $10m \times 10m$ warehouses. These environments features three agents in the smaller grid and five in the larger one, along with static obstacles (15 and 40, respectively). The agents' objective was to

achieve complete coverage of the grid while minimizing redundant exploration under spatial constraints.

In our experiments, we consider two levels of observation:

- **Proximal Information Level (PIL)** provided agents with basic positional information about their immediate neighborhood, denoted as $\mathcal{N}_k^i(t)$. Agents only perceive their immediate surrounding cells, leading to limited situational awareness.
- **Enriched Tensor-based Representation (ETR)** introduced an extended observational details, incorporating our proposed tensor-like state representation in addition to historical visitation frequencies and coverage.

In MADDPG, each agent has actor and critic networks. In the Proximal Information Level, both networks are feedforward: the actor has three layers with ReLU activation and a softmax output, while the critic uses four layers with ReLU activation to estimate Q-values for joint actions. In the Enriched Tensor-Based Representation (ETR), convolutional layers process tensor-like observations, capturing spatial and temporal dependencies. The actor processes local observations via CNNs (16, 32 filters, 3×3 , stride 1, ReLU), followed by a 64-unit LSTM layer and fully connected layers with 256, 128 units, producing action probabilities via a softmax layer. The LSTM addresses partial observability. The critic processes global state information through two convolutional layers and fully connected layers (256, 128, and 64 units), integrating a coverage progress tensor for Q-value computation.

The curiosity module includes a feature extraction network with two convolutional layers, followed by a fully connected layer producing a 64-dimensional embedding. This embedding is used by the forward dynamics model, which consists of two fully connected layers with 64 and 32 units, followed by ReLU activation, to predict the next state's features. All models are jointly optimized using Adam optimizers with learning rates of 5×10^{-4} for actors, 10^{-3} for critics, and 10^{-3} for the curiosity module. Training is conducted with a replay buffer size of 10^5 , a batch size of 128, a discount factor $\gamma = 0.95$, and a soft update factor $\tau = 0.01$. Each episode comprises of 90 or 140 steps depending on the environment size.

The experimental evaluation covered the following configurations:

1. **MADDPG (Proximal Information Level):** Standard MADDPG using basic positional information, without intrinsic rewards or enhancements. Agents received limited local observations from a 3×3 and a 9×9 window.

Table 1: Comparison of coverage percentages between MADDPG and our approach across different observation levels and window sizes.

Configuration	Proximal Information Level PIL		Enriched Tensor-Based Representation ETR	
	3x3	9x9	5x5	17x17
MADDPG	56.47 (%)	65.88 (%)	50.88 (%)	35.67 (%)
Our Approach	72.94 (%)	86.24 (%)	85.94 (%)	90.66 (%)

2. **Our framework (Enriched Tensor-based Representation):** Enhanced MADDPG utilizing enriched observation, intrinsic curiosity-based rewards, and Prioritized Experience Replay (PER). The observation window is 5×5 and 17×17 .

Average cumulative rewards and coverage percentages were recorded over 5000 episodes. Coverage was measured as the ratio of visited cells to the total number of grid cells, excluding obstacle cells. Agent behaviors were visualized using visitation maps, which highlighted the distribution of agent visits and identified areas with insufficient coverage.

6 RESULTS AND DISCUSSIONS

In our initial experiments under proximal information level (PIL), each agent is provided with a limited observation window containing only basic positional information of agent’s neighborhood set $\mathcal{N}_k(t)$ within its perception range. This local observation offers a minimal situational awareness, lacking any global context or historical visitation data. Despite extensive training across multiple episodes, as shown in 4b, agents under this configuration consistently fail to achieve effective coverage of the environment, capped at 56.47% for a 3x3 window and 65.88% for a 9x9 window, as indicated in Table 1. Their movement patterns are highly repetitive, and they often become confined to specific areas, resulting in significant gaps in overall coverage as shown in Figure 4a. This suboptimal behavior indicates that, even with the exploration noise embedded in MADDPG, the agents struggle to explore the environment effectively. The limited observational information in PIL restricts each agent’s perspective to its immediate surroundings, making it challenging to make informed movement decisions that promote efficient coverage. Consequently, agents often revisit previously explored cells rather than discovering unvisited parts of the grid, impeding comprehensive exploration.

Furthermore, in the more challenging 20x20 environment, MADDPG with PIL showed a similar trend of limited coverage. Using a 5x5 window, MADDPG agents achieved only 50.88% coverage, struggling to adapt their strategies effectively in a larger

space as shown in Figure 5a and Figure 5b. When the window was further expanded to 17x17, coverage unexpectedly dropped to 35.67%, a significant 29.89% decrease from the performance in 5x5 configuration. This decline could be attributed to the sparsity of useful information in the PIL observation space. The larger observation window introduces a majority of zero values, representing empty space, with only limited non-zero values for obstacles and agent IDs. This representation lacks the necessary variation for learning meaningful policies, effectively overwhelming the learning process and reducing the agents’ ability to extract relevant environmental features. Consequently, agents failed to prioritize unexplored regions, leading to confined and unproductive movement patterns.

To address these limitations, we incorporated an enriched tensor-based representation that expands the observation space to include additional layers, such as historical visitation frequency in an image-like format, and used CNN to process the input efficiently. This enriched observation along with the curiosity-based rewards significantly enhanced agent performance across all scenarios by incorporating historical visitation data and additional spatial context into the observation space. In the 10x10 environment, ETR achieved 72.94% coverage with a 3x3 observation window, marking a 29.23% improvement over MADDPG with PIL. Figure 4c and Figure 4d demonstrated an example coverage and average rewards of our framework with PIL. Expanding the observation window to 9x9 further boosted coverage to 86.24%, a 30.91% improvement over the PIL configuration. This demonstrates that historical context and spatial layering facilitate more strategic exploration, allowing agents to prioritize unexplored areas systematically and make more informed decisions.

Furthermore, with a 17x17 observation window, our proposed configuration achieved a notable coverage level of 90.66% and a stable learning and increased cumulative rewards over training episodes as shown in Figure 5c and Figure 5d respectively. This result reflects a substantial 42.63% absolute improvement under ETR in coverage over the PIL configuration under the same settings. This enhancement is attributed to ETR’s spatial encoding and extended cov-

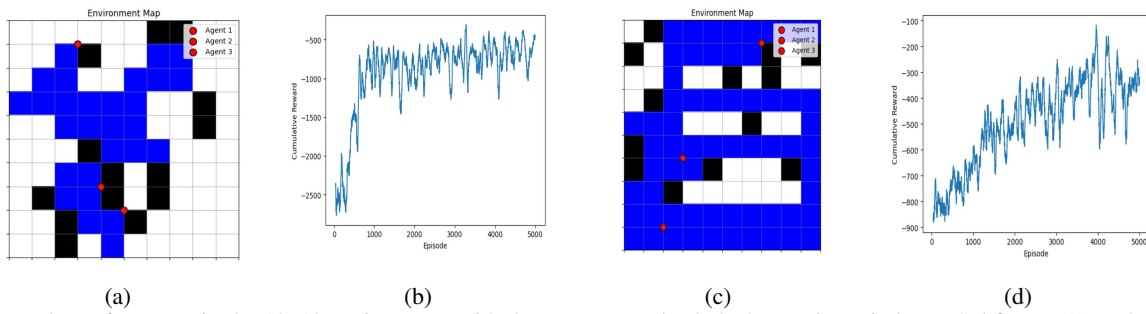


Figure 4: Performance in the 10x10 environment with three agents under 3x3 observation windows. Subfigures (a) and (b) depict area coverage and average cumulative rewards using MADDPG and PIL, respectively. While subfigures (c) and (d) show area coverage and average cumulative rewards using our framework with PIL.

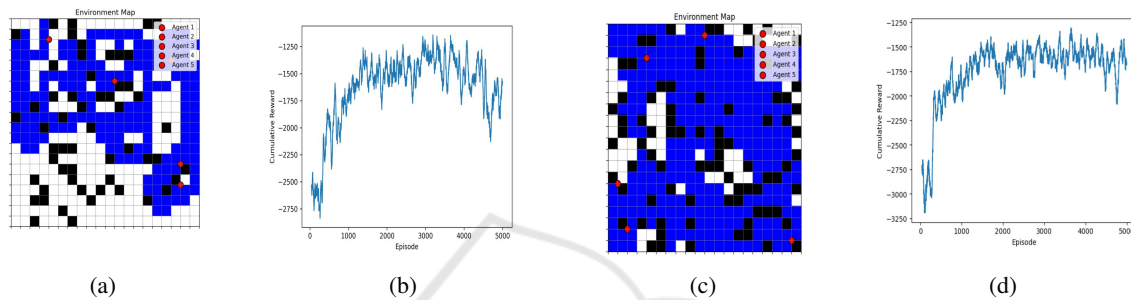


Figure 5: Performance in the 20x20 environment with five agents. Subfigures (a) and (b) depict area coverage and average cumulative rewards using MADDPG with a 5x5 observation window, while subfigures (c) and (d) show area coverage and average cumulative rewards using our framework with ETR and a 17x17 observation window.

erage observations, which effectively address the limitations associated with the sparse, zero-dominated observations characteristic of PIL. By incorporating meaningful spatial context and past visitation data, ETR enables agents to avoid redundant coverage and strategically prioritize unexplored areas. The integration of curiosity-driven intrinsic rewards further enhances exploration efficiency by incentivizing agents to seek novel states, promoting balanced and thorough exploration.

Overall, the results demonstrated an improvement of up to 42.63% in absolute coverage compared to baseline approaches, underscoring the effectiveness of combining enriched observation representations with intrinsic rewards. Our framework advances the state-of-the-art MADDPG in the area coverage problem under spatial constraints, promoting coordinated and comprehensive exploration in multi-agent environments.

7 CONCLUSION

The rising demand for automated and efficient area coverage in structured environments, such as warehouses and industrial facilities, highlights the need for robust solutions to enhance exploration and coordination in multi-agent environments under resource

limitations and complex spatial constraints. The proposed framework integrates an enriched tensor-based representation and prioritized experience replay with curiosity-driven intrinsic rewards to utilize spatial encoding and visitation history, driving agents to explore novel and less-visited areas. Prioritized experience sampling further enhances the model's ability to learn from most informative experiences to prompt successful navigation and exploration. Together, these components foster an adaptive and exploratory learning process, particularly effective in spatially constrained environments with limited information where traditional strategies fall short.

Experimental results demonstrated the efficacy of the proposed framework, achieving up to a 42.63% improvement in grid coverage compared to vanilla PIL-based framework. This improvement was especially significant in larger and more complex environments, where ETR facilitated systematic navigation and comprehensive coverage with reduced overlap. Notably, during training, the centralized critic supports inter-agent coordination by leveraging global state information. Post-training, agents rely solely on decentralized actors policies and local observations, ensuring scalability and responsiveness for real-world robotics without computational overhead.

In conclusion, the proposed framework successfully advances the state of the art in MARL for area

coverage tasks by addressing spatial and resource constraints. Its ability to integrate enriched observations, prioritize meaningful experiences, and promote adaptive exploration makes it a promising solution for real-world applications in structured and resource-constrained environments.

ACKNOWLEDGMENTS

The second author was in part supported by a research grant from Google.

REFERENCES

- Atınc, G. M., Stipanović, D. M., and Voulgaris, P. G. (2020). A swarm-based approach to dynamic coverage control of multi-agent systems. *IEEE Transactions on Control Systems Technology*, 28(5):2051–2062.
- Burgard, W., Moors, M., Stachniss, C., and Schneider, F. E. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376–386.
- Chen, X., Tucker, T. M., Kurfess, T. R., and Vuduc, R. (2019a). Adaptive deep path: efficient coverage of a known environment under various configurations. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3549–3556. IEEE.
- Chen, Z., Subagdja, B., and Tan, A.-H. (2019b). End-to-end deep reinforcement learning for multi-agent collaborative exploration. In *2019 IEEE International Conference on Agents (ICA)*, pages 99–102. IEEE.
- Choi, H.-B., Kim, J.-B., Han, Y.-H., Oh, S.-W., and Kim, K. (2022). Marl-based cooperative multi-agv control in warehouse systems. *IEEE Access*, 10:100478–100488.
- Garaffa, L. C., Basso, M., Konzen, A. A., and de Freitas, E. P. (2021). Reinforcement learning for mobile robotics exploration: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):3796–3810.
- Gazi, V. and Passino, K. M. (2004). Stability analysis of swarms. *IEEE Transactions on Automatic Control*, 48(4):692–697.
- Ghaddar, A. and Merei, A. (2020). Eaoa: Energy-aware grid-based 3d-obstacle avoidance in coverage path planning for uavs. *Future Internet*, 12(2):29.
- Hu, J., Niu, H., Carrasco, J., Lennox, B., and Arvin, F. (2020). Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(12):14413–14423.
- Jin, Y., Zhang, Y., Yuan, J., and Zhang, X. (2019). Efficient multi-agent cooperative navigation in unknown environments with interlaced deep reinforcement learning. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2897–2901. IEEE.
- Khamis, A., Hussein, A., and Elmogy, A. (2015). Multi-robot task allocation: A review of the state-of-the-art. *Cooperative robots and sensor networks 2015*, pages 31–51.
- Li, W., Zhao, T., and Dian, S. (2022). Multirobot coverage path planning based on deep q-network in unknown environment. *Journal of Robotics*, 2022(1):6825902.
- Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., and Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30.
- Luo, T., Subagdja, B., Wang, D., and Tan, A.-H. Multi-agent collaborative exploration through graph-based deep reinforcement learning. In *2019 IEEE International Conference on Agents (ICA)*, pages 2–7. IEEE.
- Orr, J. and Dutta, A. (2023). Multi-agent deep reinforcement learning for multi-robot applications: A survey. *Sensors*, 23(7):3625.
- Sanghvi, N., Niyogi, R., and Milani, A. (2024). Sweeping-based multi-robot exploration in an unknown environment using webots. In *ICAART (1)*, pages 248–255.
- Schaul, T. (2015). Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.
- Setyawan, G. E., Hartono, P., and Sawada, H. (2022). Cooperative multi-robot hierarchical reinforcement learning. *Int. J. Adv. Comput. Sci. Appl.*, 13:35–44.
- Stentz, A. (1994). Optimal and efficient path planning for partially-known environments. In *Proceedings of the 1994 IEEE international conference on robotics and automation*, pages 3310–3317. IEEE.
- Tan, C. S., Mohd-Mokhtar, R., and Arshad, M. R. (2021). A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. *IEEE Access*, 9:119310–119342.
- Tran, V. P., Garratt, M. A., Kasmarik, K., Anavatti, S. G., and Abpeikar, S. (2022). Frontier-led swarming: Robust multi-robot coverage of unknown environments. *Swarm and Evolutionary Computation*, 75:101171.
- Yamauchi, B. (1997). 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*, pages 146–151. IEEE.
- Yanguas-Rojas, D. and Mojica-Nava, E. (2017). Exploration with heterogeneous robots networks for search and rescue. *IFAC-PapersOnLine*, 50(1):7935–7940.
- Zhang, H., Cheng, J., Zhang, L., Li, Y., and Zhang, W. (2022). H2gcn: Hierarchical-hops graph neural networks for multi-robot exploration in unknown environments. *IEEE Robotics and Automation Letters*, 7(2):3435–3442.
- Zhelo, O., Zhang, J., Tai, L., Liu, M., and Burgard, W. (2018). Curiosity-driven exploration for mapless navigation with deep reinforcement learning. *arXiv preprint arXiv:1804.00456*.