Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Collaborative Deep Q-Network for Multi-Robot Coverage Path Planning in Complex Environments

Wan Xu

xuwan@mail.hbut.edu.cn

Hubei University of Technology

Jinduan Liu

Hubei University of Technology

Yujie Wang

Hubei University of Technology

Shize Wang

Hubei University of Technology

Research Article

Keywords: multi-robot systems, deep reinforcement learning, collaborative decision-making, coverage path planning

Posted Date: November 6th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11425388/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Collaborative Deep Q-Network for Multi-Robot Coverage Path Planning in Complex Environments

Wan Xu[1,2,*], Jinduan Liu[1], Yujie Wang[1], and Shize Wang[1]

1 School of Mechanical Engineering, Hubei University of Technology, Wuhan, Hubei, P. R. China, 430068.

2 Hubei Key Laboratory of Modern Manufacturing Quality Engineering, Wuhan, Hubei, P. R. China, 430068.

* Corresponding author. E-mail: xuwan@mail.hbut.edu.cn

Abstract: Existing multi-robot coverage methods in unknown environments suffer from inadequate use of inter-robot interaction information, limited collaborative decision-making capabilities, excessive repeated coverage, and insufficient adaptability to different team sizes and map environments. To address these issues, this paper proposes a collaborative deep Q-network (CD-DQN) framework. The framework constructs a task-state representation integrating robot-specific information, local environmental observations, regional task information, and inter-robot collaborative relationships. A multi-head collaborative attention module uses the current robot's features as the query and the team members' features as the keys and values to adaptively aggregate interaction information relevant to the current coverage decision. On this basis, action-value estimation using a dueling deep Q-network (Dueling DQN), dynamic region allocation, and a multi-level reward mechanism are combined to guide robots toward unexplored areas while reducing repeated coverage and motion conflicts. A fixed-length representation of collaborative features is also adopted, and the framework is trained with varying numbers of robots and randomly generated map layouts to improve policy adaptability to different team sizes and map environments. Experimental results show that, within the tested configurations, CD-DQN maintains high coverage performance and demonstrates adaptability to changes in map size and team size. Compared with the baseline algorithms, CD-DQN achieves

higher coverage rates and fewer completion steps in both tested scenarios, with competitive redundancy and collaboration performance.

Keywords: multi-robot systems; deep reinforcement learning; collaborative decision-making; coverage path planning

1 Introduction

In recent years, multi-robot coverage and exploration have received increasing attention. Representative studies have investigated collaborative complete coverage and exploration [1], air–ground collaborative mapping and reinforcement-learning-based navigation in disaster environments [2], Voronoi-GRU-based collaborative exploration in unknown environments [3], failure-resilient cooperative coverage [4], coverage planning for the three-dimensional reconstruction of large structures [5], online concurrent multi-robot coverage planning [6], robust simultaneous exploration and coverage with robot-failure recovery [7], and DQN-based multi-robot coverage in unknown environments [8]. Recent comparative research has also examined different learning-based and decentralized strategies for dynamic multi-robot path planning [9]. These studies show that multi-robot systems can improve task execution through concurrent coverage [6], reduce repeated coverage through learning-based decision-making [8], and maintain coverage robustness under robot failures [4][7][15]. Further improvements have been achieved through minimum-length coverage-path planning [10] and integrated task allocation and online coverage-path planning [11]. Related research on unmanned aerial vehicles has investigated hazardous flight-region prediction in urban environments [12] and integrated path planning and trajectory-tracking control under environmental and system uncertainties [13]. Although these UAV navigation studies address safety, obstacle avoidance, and motion control, their objectives differ from the complete multi-robot coverage of unknown environments considered in this study. Therefore, achieving efficient collaborative coverage in complex unknown environments still requires the coordinated treatment of task allocation, inter-robot interaction, path conflicts, and repeated coverage.

Extensive research has been conducted on multi-robot coverage path planning algorithms. For the two major categories of known and unknown environments, numerous classical methods and improved strategies have been proposed. A known environment refers to a situation in which the multi-robot system, or each individual

robot, has obtained complete map information of the workspace before the task begins, including all reachable areas, obstacle locations, and obstacle geometries. An unknown environment refers to a situation in which the system has little or no prior information about the workspace before the task begins. The environmental map is updated online from robot observations as coverage proceeds.

In known environments, robots can use prior maps for task-region partitioning [17] and path planning, and related studies have mainly focused on coverage-path optimization and task allocation to improve coverage efficiency [18]. Existing studies have improved multi-robot coverage efficiency by reducing the number of turns and optimizing path allocation [19], employed mathematical programming methods to optimize coverage-path length or task-completion time [20][23], and improved coverage-planning efficiency through map decomposition and enhanced backtracking strategies [21][22]. Chen et al. proposed an ant colony system-based algorithm for coverage path planning of heterogeneous UAVs across multiple regions [16].

Coverage in unknown environments requires robots to update the environmental map in real time without complete prior map information while performing the coverage task. Existing studies have mainly focused on exploration strategies, online planning, and frontier selection. Jiao et al. proposed an online Multi-Agent Coverage Path Planning (MACPP) algorithm based on proximity interaction and a cooperation mechanism, achieving efficient coverage in unknown environments while minimizing coverage time and repeated paths [24]. However, its collaborative information is mainly derived from the prediction results of Model Predictive Control (MPC) within local neighborhoods, without directly exploiting state relationships among non-neighboring robots. Yuan et al. proposed a two-stage area-coverage method based on Multi-Agent Deep Reinforcement Learning (MADRL) to improve coverage efficiency and practical applicability [25]. However, this method performs regional planning and collaborative control in two separate stages and does not jointly model inter-robot interaction features within a unified decision model. Wang et al. proposed the MAC-Planner framework, which improves system efficiency and robustness through dynamic task allocation and path planning and is applicable to cleaning, monitoring, and other scenarios [11]. However, this framework first requires the construction of a coarse-grained map and uses K-means clustering and pairwise optimization for task allocation; consequently, its coverage performance is closely related to the initial clustering and regional

partitioning results. Wang et al. proposed APF-CPP, an artificial-potential-field-based online multi-robot coverage path planning method. This method derives a coverage strategy using an artificial potential field, determines movement directions using repulsive forces among robots, and combines a dead-end recovery mechanism with a policy-update mechanism to achieve efficient task allocation and regular coverage [26]. However, its collaboration and task guidance mainly depend on predefined potential fields and policy-update rules, rather than adaptively extracting inter-robot relationships in a data-driven manner. Yu et al. proposed a multi-robot waypoint planning system consisting of a region-decomposition-based local waypoint planner and a submap-based global waypoint planner for target search in unknown environments [27]. The local planner generates coverage waypoints through area decomposition to accelerate global information acquisition, whereas the global planner uses global information to optimize waypoint planning to avoid repeated searches. Their combination ensures both a high coverage rate and efficient search. Nevertheless, this study focuses on target search in unknown environments rather than complete coverage of all reachable grid cells. Zhao et al. proposed a clustering-based hyper-heuristic algorithm for the Heterogeneous UAV Coverage Path Planning problem (HUCPP) [28]. Under schedulability and optimality constraints, the method provides optimized trajectories for UAVs within a given planning space.

To address the requirements of online navigation and collaborative decision-making in unknown environments, deep reinforcement learning provides a feasible approach for robots to learn action policies through interactions with the environment [14]. Deep Q-Networks (DQN) use deep neural networks to approximate action-value functions, providing an effective method for sequential decision-making in high-dimensional state spaces [29]. Double DQN separates action selection from value evaluation to alleviate value overestimation [30], whereas Dueling DQN decomposes the action value into a state value and an action advantage to improve the discrimination among different action values [31]. In addition to action-value estimation, multi-robot collaborative decision-making requires the effective extraction and fusion of inter-robot interaction information. In specific applications, Tai et al. used deep reinforcement learning to map range measurements and relative target positions to continuous control commands, thereby realizing mapless navigation in previously unseen environments [37]. However, its decision-making process focuses on reaching a given target, and

continuous coverage of unknown areas still requires additional task organization and target-assignment mechanisms. Long et al. applied deep reinforcement learning to decentralized multi-robot collision avoidance and generated motion commands from local observations [38]. However, its state and reward designs mainly serve goal reaching and collision avoidance and do not directly characterize team coverage history or regional task overlap. Therefore, they are insufficient to directly guide collaborative behaviors that simultaneously consider coverage completeness and low redundancy. Multi-Head Attention (MHA) can perform weighted aggregation of input features in multiple representation subspaces [32], and attention mechanisms have also been applied to information selection and interaction modeling in multi-agent reinforcement learning [33]. Chen et al. introduced a self-attention mechanism into deep reinforcement learning and improved navigation efficiency in crowded environments by aggregating robot-pedestrian and pedestrian-pedestrian interaction features [39]. However, its interaction representation mainly focuses on motion states and their influence on goal navigation and does not establish relationships among coverage history, responsibility regions, and task progress. Therefore, when applied to multi-robot coverage, a collaborative information representation supporting task division and redundancy suppression is still required. Overall, the task objectives, reward designs, and interaction representations of the aforementioned navigation studies mainly focus on goal reaching and motion safety. For complete multi-robot coverage in unknown environments, coverage progress and regional task information must be incorporated into the decision-making process, and inter-robot relationships must be used to coordinate exploration responsibilities and suppress redundant coverage.

Compared with multi-robot map coverage in known environments, coverage in unknown environments is more consistent with practical application requirements and has broad application prospects in disaster rescue, geological exploration, military reconnaissance, and other fields. However, it also presents greater challenges. Although substantial progress has been made in unknown-environment coverage, the following limitations remain for complete coverage tasks involving variable-sized multi-robot teams. First, some methods rely on predefined regional partitioning, clustering results, or specific task settings [10][25][28], and the adaptability of their task-allocation processes to changes in robot number and map structure remains constrained by the corresponding planning mechanisms. Second, online methods based on neighborhood

interaction or artificial potential fields mainly rely on local neighborhood information and predefined guidance rules [24][26] and have not simultaneously optimized coverage completeness, redundant coverage, and task-completion time within a unified learning policy. Third, the state representations of some learning-based navigation methods focus mainly on local observations and motion interactions and make limited use of the relationships among coverage history, responsibility regions, and team task progress [38][39]. For collaborative coverage, task-related information must be further integrated so that inter-robot interactions can directly support exploration-task division and redundant-coverage suppression.

To address the above issues, this paper proposes a multi-robot coverage path planning framework based on a Collaborative Deep Q-Network (CD-DQN). The framework incorporates local environmental observations, coverage states, and regional task information into the state representation, and uses inter-robot multi-head collaborative attention to fuse task-related interaction features. Dynamic region allocation and a multi-level reward mechanism are further combined to coordinate exploration-task division and reduce redundant coverage. By training in environments with varying numbers of robots and randomly generated maps, the multi-robot system can autonomously adapt to different team sizes and complex environments and achieve efficient collaborative coverage. The main contributions of this paper are summarized as follows:

1. To address the difficulty of conventional DQN in effectively modeling inter-robot relationships during multi-robot coverage, a CD-DQN framework for unknown grid environments is constructed. The framework designs a task-state representation containing robot-specific information, local environmental observations, regional task information, and collaborative relationships. An inter-robot multi-head collaborative attention module is used to adaptively fuse team interaction information, and coverage actions are determined through a Dueling value-advantage architecture;

2. A collaborative guidance mechanism combining dynamic region allocation and multi-level rewards is designed. Dynamic region allocation updates each robot's responsibility region according to its current position and uses the geometric centroid of the responsibility region to provide task guidance. The reward function

comprehensively considers first visits, independent coverage, team coverage growth, repeated coverage, regional deviation, and collisions, thereby guiding robots to explore unknown areas while reducing task overlap and motion conflicts;

3. An experimental validation system consisting of baseline comparisons and key-component ablation studies is established. Under unified map-generation methods, training budgets, and task-termination conditions, the proposed framework is compared with DQN, MADRL, MACPP, and APF-CPP. Component ablation studies are also conducted by removing the attention mechanism, Dueling architecture, dynamic region allocation, and multi-level reward mechanism, and the contribution of each component is evaluated in terms of coverage rate, redundancy rate, completion steps, and collaboration efficiency. In addition, a coverage scenario involving three E-puck robots is constructed in Webots to further verify the execution of the discrete CD-DQN actions under continuous differential-drive motion.

2 Problem Description and Modeling

2.1 Problem Description

This paper considers a multi-robot system consisting of n mobile robots, denoted by $R = \{r_i\}_{i=1}^{n}$, equipped with identical square coverage tools with a side length of d. For ease of computation and decision-making, the continuous two-dimensional workspace Ω is discretized into a regular grid map. Specifically, the entire workspace is divided into an H×W grid. The physical size of each grid cell is exactly d×d, which completely matches the size of the robots' coverage tools. When the center of a robot's coverage tool is located at the center of a grid cell, that cell is completely covered.

Therefore, the multi-robot coverage path planning problem can be defined as follows: given a continuous two-dimensional workspace Ω, which is discretized into an H×W grid map G, and a robot set R, a real-time planner enables all reachable areas in the workspace to be covered by at least one robot. On this basis, optimal efficiency is achieved by minimizing the total time required to complete the entire coverage task, reducing the total travel distance of the robots, lowering energy consumption, minimizing repeated coverage among multiple robots, and improving the overall system efficiency. The system must also exhibit good robustness to ensure that it can cope with environmental changes and potential robot failures. Finally, this paper assumes that each robot in the multi-robot system can accurately obtain the global

coordinates of the grid cell in which it is located. The local sensing radius of each robot is one grid cell; that is, each robot can obtain information within a 3×3 grid region centered on its current position, while grid cells outside this range remain unknown. The robots share their positions and coverage states. The communication range covers the entire experimental map, and the information is updated synchronously at each time step.

2.2 Initial Environment

For environmental modeling, we adopt a multi-level representation method capable of simultaneously capturing the physical characteristics of the environment, the coverage status, and the robots' cognitive information. The first level describes cell types. Each cell is in one of the following three basic states at any given time:

1) Free Cell: Free cells represent open space that can be freely accessed and covered by robots. These cells form the target set for the coverage task and are the primary areas robots need to visit.

2) Obstacle Cell: Obstacle cells represent spaces occupied by obstacles. Robots cannot pass through or cover these cells. Static obstacles are used in the experimental environment, and the positions of the obstacle cells remain unchanged within each episode.

3) Unknown Cell: Robots do not initially have complete information about the entire environment. Unknown cells represent areas that robots have not yet sensed or explored. As robots move and cover, unknown cells are gradually converted into free cells or obstacle cells.

The second level describes map types. To comprehensively describe the environmental state and coverage progress and to facilitate computation, this paper defines the following two types of maps:

1) Environmental Cognition Map $K \in \{-1,0,1\}^{H\times W}$: This map represents the robot system's current cognition of the environment. At the initial time, most areas may be unknown, with unknown cells represented by -1. Specifically, K_t=-1, 0, 1 indicate that a grid cell is unknown, confirmed to be free, and confirmed to be an obstacle at time t, respectively. When a grid cell enters a robot's sensing range, the system updates its state from unknown to free or obstacle according to the sensing result. As exploration proceeds, the robots gradually update this map through coverage.

2) Coverage Map Tensor $C \in \{0,1\}^{H\times W\times n}$: This is a three-dimensional tensor that

records in detail the coverage history of each robot at each location. In addition, a visit count matrix $V_t \in \mathbb{N}_0^{H\times W}$ is defined to record the total number of visits to each location by all robots, facilitating the subsequent calculation of redundant coverage.

The initial environment is set as follows. In each training episode, a 30×30 map is randomly generated, in which 50–90 grid cells are randomly selected as static obstacle cells. The obstacle cells account for approximately 5.6%–10.0% of the total number of grid cells. After the map is generated, a connectivity check is performed on all free cells. Only maps in which the free space forms a single connected region are retained, ensuring that the robots can reach all grid cells to be covered. To improve the model's adaptability to changes in team size, the number of robots $n\in\{2,3,\cdots,9\}$ is randomly generated in each training episode, and the initial positions of the robots are randomly initialized. The multi-robot system stops after collaboratively completing the coverage task or reaching the maximum step limit.

2.3 Dynamic Region Allocation

The Voronoi diagram is a classic concept in computational geometry, and its core idea is to partition space according to the distances to a specific set of points. In CD-DQN, this paper applies Voronoi partitioning to dynamic region allocation, with each robot using its current position as a seed point of the Voronoi diagram. For each free cell that has been confirmed as uncovered in the current map, its Euclidean distances to all robots are calculated, and the cell is then assigned to the nearest robot. When the distances are equal, the robot ID is used as a secondary criterion. Region partitioning is always based on the real-time positions of the robots, ensuring the dynamic adaptability of the allocation. The geometric centroid of the responsibility region is set as the target position, and the direction from the robot's current position to the target position is used as the initial movement direction. The specific movement strategy is described in the next section.

The dynamic updating of region allocation includes two mechanisms. The first is periodic updating, in which the region allocation is updated every 50 time steps. As the robots move and explore the environment, the region partitioning is automatically adjusted. The second is event-triggered updating. In addition to periodic updating, certain events also trigger immediate updating, including when new environmental information is perceived, when a robot discovers a relatively large unknown area, or when the state of a grid cell in the cognition map is updated from unknown to obstacle;

when the responsibility region is empty, that is, when a robot completes the coverage of its responsibility region or becomes trapped in a local deadlock; and when the number of robots changes at the beginning of each episode, in which case the system repartitions the responsibility regions according to the current robot set to adapt to different team sizes. These updating mechanisms allow the responsibility regions to be adjusted according to robot positions, environmental cognition, and coverage progress, thereby improving the consistency between region allocation and the current coverage state.

2.4 Movement Strategy

In terms of movement strategy, CD-DQN has nine discrete actions, including eight movement actions and one wait action. Each movement action corresponds to a specific direction and displacement, as illustrated in Fig. 1. Within each time step, a robot can move only to one grid cell adjacent to its current grid cell. When the wait action is selected, the robot remains at its current position.

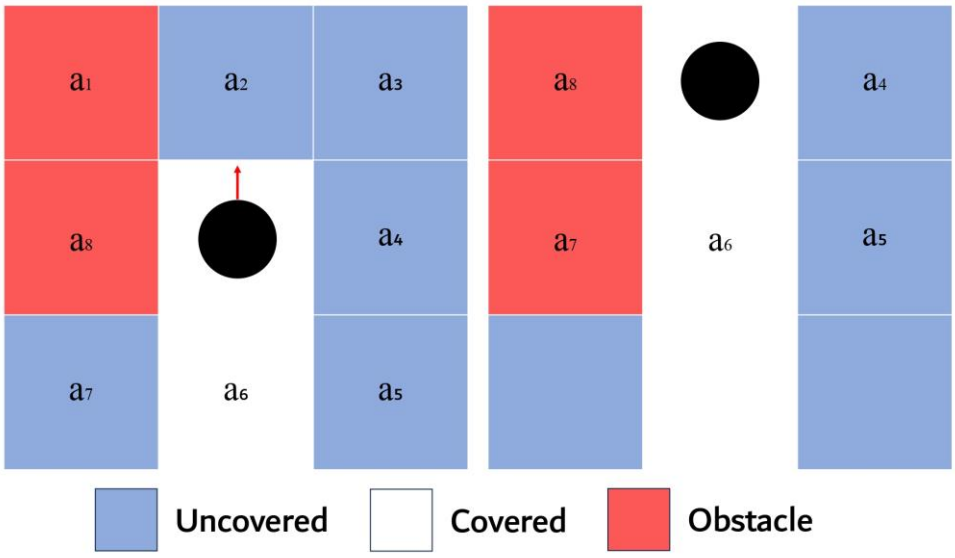


Fig. 1 Robot movement strategy. Actions a_1–a_8 represent the eight movement directions. The robot selects the action with the highest Q-value

Before executing an action, the system must verify the feasibility of the action. This includes a boundary check to ensure that the target position is within the map boundaries and an obstacle check to determine whether the target position is occupied by an obstacle. The obstacle check is based on the known map K, which means that a robot can make decisions only on the basis of known information. The action feasibility

checks mainly include the map boundary check, known-obstacle check, diagonal corner-cutting check, and robot occupancy conflict check. In this paper, each robot is simplified as a point robot occupying a single grid cell. Each robot selects an action according to the current environmental state, and the robots execute their actions sequentially according to their IDs. If the target grid cell is already occupied, the action is invalid and the robot remains at its current position. When multiple robots select the same grid cell, the robot with the smaller ID enters the grid cell first. A corresponding action is considered feasible only when the candidate position satisfies the above conditions. The network outputs the Q-values of all actions according to the current state and selects a candidate action, after which the environment determines the legality of the candidate action. If the action causes the robot to move beyond the map boundary, enter an obstacle cell, conflict with another robot, or be blocked during diagonal movement, the current movement is cancelled and the corresponding penalty is assigned. Otherwise, the action is executed and the robot state is updated. When a robot selects the wait action, or when the selected movement action cannot be executed because of a map boundary, obstacle, or robot occupancy conflict, the robot remains at its current position during the current time step.

3 CD-DQN-Based Path Planning Method

3.1 Overview of the CD-DQN Method

The Collaborative Deep Q-Network (CD-DQN) method proposed in this paper is a multi-robot coverage path planning framework for unknown environments. Its core idea is to fuse team interaction information through inter-robot multi-head collaborative attention, together with dynamic region allocation and a multi-level reward mechanism, to guide robots in forming task divisions and reducing repeated coverage, thereby completing the map coverage task efficiently and with low redundancy.

At each time step, the system constructs a state representation based on robot-specific information, local environmental perception, regional task information, and the shared team state. Dynamic region allocation provides each robot with a task region and an exploration direction as task guidance for coverage decision-making. Through the inter-robot multi-head collaborative attention module, the CD-DQN network

aggregates team-state and task information relevant to the current robot's coverage decision and uses the fused features for action-value estimation to output the Q-value of each candidate action. During training, the multi-level reward mechanism provides feedback for policy learning through new-region exploration rewards, cross-robot redundancy penalties, regional-consistency rewards, and other terms, guiding robots to complete regional tasks and reduce repeated coverage.

Experiences generated through interaction with the environment are stored in the replay buffer for iterative network-parameter updates. Team-information fusion, regional-task guidance, and reward feedback jointly constitute the collaborative coverage mechanism proposed in this paper. The overall system framework is shown in Fig. 2.

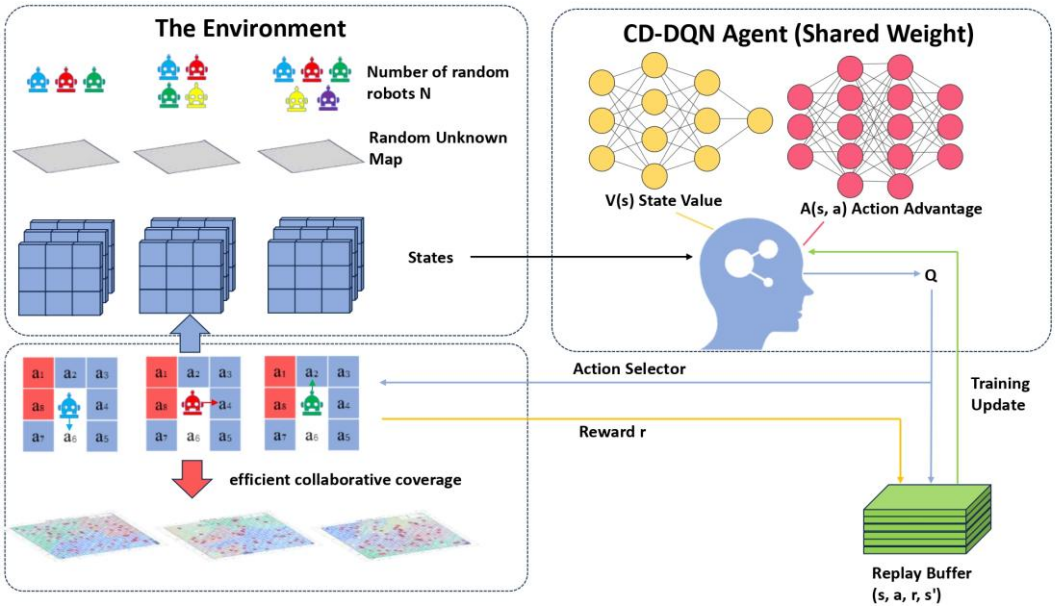


Fig. 2 System Framework

3.2 CD-DQN Network Architecture

To address the difficulty of traditional DQN in fully representing inter-robot state dependencies and task-coupling relationships during multi-robot coverage, this paper constructs the CD-DQN network architecture. The network uses Dueling DQN [31] as the basis for action-value estimation and, on top of a multi-source state representation for the coverage task, designs an inter-robot multi-head collaborative attention module: the current robot's features are used as the query, and the features of all robots in the team are used as keys and values to adaptively extract robot-interaction information relevant to the current action decision, after which the fused collaborative features are

used for action-value estimation. Unlike the existing MHA-DQN method for personalized route planning [34], CD-DQN further combines inter-robot interaction modeling with dynamic region allocation and a multi-level reward mechanism, forming a collaborative decision-making framework for complete multi-robot coverage in unknown environments. The overall network architecture of CD-DQN is shown in Fig. 3.

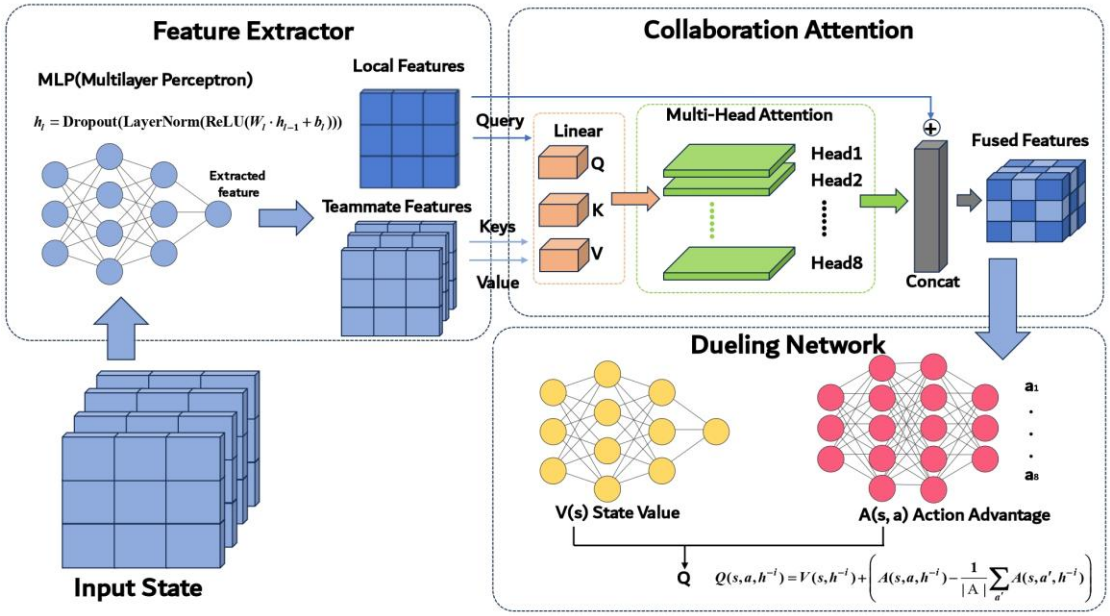


Fig. 3 Network architecture of the proposed CD-DQN framework

The feature extraction network of CD-DQN adopts a multilayer perceptron (MLP) structure. The first layer is responsible for converting the raw state vector into a high-dimensional feature representation. The second and third layers further refine these features, gradually constructing a more abstract representation. This forms a feed-forward neural network whose core operation is the linear transformation and non-linear activation between layers:

$$h^{l+1} = \text{Dropout}(\text{LayerNorm}(\text{ReLU}(W^l h^l + b^l))) \tag{3.1}$$

Where h^l is the output activation of the l-th layer (and input to the next), W^l is the weight matrix of the l-th layer, and b^l is the bias vector. ReLU (Rectified Linear Unit), defined as $\max(0, x)$, introduces non-linearity. LayerNorm normalizes all features for a single sample. Dropout, with a probability $p = 0.1$, randomly sets a fraction of neuron outputs to zero during training for regularization, preventing overfitting.

The multi-head attention mechanism [32] computes feature correlations in different representation subspaces through multiple parallel attention heads and performs weighted aggregation of the input information. This mechanism has been used for information selection and interaction modeling in multi-agent reinforcement learning. On this basis, this paper designs an inter-robot multi-head collaborative attention fusion module to address the complexity of interaction information and the mutual influence of robot decisions during multi-robot coverage in unknown environments. Unlike MHA-DQN for personalized route planning, this paper uses the attention mechanism to model state dependencies, coverage relationships, and regional-task relationships among multiple robots. Specifically, for robot i, its feature vector hi is used as the query, and the team feature matrix Z composed of all robots' features is used as the keys and values. Different attention heads compute feature correlations between the current robot and the team members and adaptively aggregate interaction information relevant to the current coverage decision. The resulting team collaborative feature is then input into the Dueling value–advantage branches to estimate the long-term value of each candidate action. This design enables CD-DQN to dynamically adjust the degree of attention to information according to the state and task relationships among robots, providing feature support for multi-robot collaborative coverage decisions.

Let h_i denote the feature vector of robot i. Stacking the feature vectors of all n robots by row gives the team feature matrix:

$$Z = [h_1; h_2; \dots; h_n] \in R^{n \times d_f} \tag{3.2}$$

where d_f denotes the dimension of the feature vector of an individual robot.

$$q_i = h_i W^Q, \quad K = ZW^K, \quad V = ZW^V \tag{3.3}$$

$$\text{Attention}(q_i, K, V) = \text{softmax}\left(\frac{q_i K^T}{\sqrt{d_k}}\right) V \tag{3.4}$$

where q_i is the query vector of the current robot i, K is the key matrix, and V is the value matrix; h_i denotes the feature vector of the current robot i, and Z is the team feature matrix obtained by stacking the feature vectors of all n robots. W^Q, W^K, and W^V are projection matrices. $q_i K^T$ calculates the similarity between the query and all keys. A higher similarity indicates that the information corresponding to that key is more relevant to the current query. d_k is the dimension of the key vector, and $\sqrt{d_k}$ is the scaling factor. Softmax is the normalized exponential function, which converts the similarity scores into a probability distribution whose sum is 1, namely the attention

weights. The attention output is essentially a weighted sum of the value vectors V, where the weights are determined by the similarity between the query q_i and the keys K. The final Multi-Head Attention concatenates the feature vectors output by multiple attention heads.

$$\text{MultiHead}(q_i, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_{m_h})W^O \tag{3.5}$$

$$\text{head}_j = \text{Attention}\left(q_i W_j^Q, KW_j^K, VW_j^V\right), \qquad j = 1,2,\ldots,m_h \tag{3.6}$$

where W_j^Q, W_j^K, W_j^V are the unique trainable projection matrices of the j-th head; m_h is the number of attention heads, and $\boldsymbol{W}^O$ is the trainable output projection matrix that projects the concatenated high-dimensional vector back to the original dimension.

CD-DQN also adopts the Dueling network architecture, the core idea of which is to decompose the Q-value function into a state-value function $V(s)$ and an advantage function $A(s,a)$. This decomposition is meaningful for the multi-robot coverage task. $V(s)$ represents the expected return in the current state, regardless of the specific action chosen. In the coverage task, some states, such as a uniform distribution of robots, inherently have high value, regardless of the specific next action. $A(s,a)$ represents the advantage of choosing a specific action relative to the average. In many coverage scenarios, multiple actions may have similar values (e.g., moving in different directions in an open area). The advantage function helps the network focus on learning these subtle differences. The final Q-value is calculated as:

$$V(s,h^{-i}) = W_v \cdot (\text{ReLU}(W_{v1} \cdot h_{attn})) + b_v \tag{3.7}$$

$$A(s,a,h^{-i}) = W_a \cdot (\text{ReLU}(W_{a1} \cdot h_{attn})) + b_a \tag{3.8}$$

$$Q(s,a,h^{-i}) = V(s,h^{-i}) + \left(A(s,a,h^{-i}) - \frac{1}{|\text{A}|}\sum_{a'} A(s,a',h^{-i}) \right) \tag{3.9}$$

where h_{attn} represents the feature vector enhanced by the attention mechanism. $\mathbf{w}_v, \mathbf{w}_a$ are the weight parameters for the value and advantage streams, and b_v, b_a are the corresponding biases. s is the current state, and h^{-i} is the collaborative team feature, which is the aggregated information from all other robots' feature vectors processed by the collaborative attention mechanism. $|\text{A}|$ represents the size of the

action space (9 in this system), and subtracting the mean advantage centers the advantage values.

3.3 State Representation

The agent's state representation is the basis for its decision-making. To achieve effective collaboration and efficient exploration in the multi-robot coverage task, this framework constructs a high-dimensional state vector for each robot. This vector integrates the robot's own proprioceptive information, local environmental perception, collaborative relationships with other robots, and global task progress, totaling 106 dimensions. The specific components of the state vector are as follows:

- Robot Proprioception (4 dims): Includes a 2D normalized position (x, y) and 2D normalized velocity, derived from the difference between the current and previous time step's pose.
- Local Environment Perception (36 dims): A 3×3 local perception window centered on the robot's current position. For each cell within this window, 4 key features are extracted: terrain information (value from the Known Map K), coverage status (whether it has been covered by any robot), visit frequency (number of times visited), and other-robot coverage (whether it has been covered by a robot other than itself). This forms a 36-dimensional feature block.
- Region Task Allocation Info (4 dims): Based on the dynamic Voronoi region allocation, this provides task guidance to the robot. It consists of the target direction (2D vector), region ownership (a flag), and target distance.
- Multi-Robot Collaboration Features (54 dims): To promote collaboration and avoid redundancy, each robot needs to perceive the states of other robots within its communication range. To support a variable number of robots, the maximum number of robots allowed by the system is set to $n_{max} = 10$, and the collaboration-feature portion of the state vector is padded to a fixed length. For each of up to $n_{max} - 1$ other robots, the following six features are calculated: communication link (indicating whether the robot is within communication range), normalized distance (distance to the other robot), shared coverage count (number of cells covered by both robots), other-robot coverage count (number of cells covered by the other robot), intrusion flag (indicating whether the robot

itself is within the other robot's task region), and region-size ratio (ratio of the size of its own task region to that of the other robot). This gives a total of 54 dimensions.

- Global & Individual Progress Metrics (8 dims): Includes 7 indicators: global coverage rate, individual contribution, individual redundancy, time progress, exploration direction (2D vector), coverage efficiency, and wait counter. This provides the robot with a macroscopic view of task progress to coordinate global behavior.

3.4 Reward Function

The reward function is at the core of reinforcement learning, defining "good" and "bad" behavior for the agent and directly guiding the policy's learning direction. To achieve high coverage, low redundancy, and strong collaboration simultaneously, the reward function for CD-DQN is designed as a multi-level reward function that provides dense feedback, integrating immediate exploration, efficiency penalties, collaborative guidance, and safety constraints.

Let p be the robot's current position and p' be the new position after executing action a. We define the following symbols: $V(p')$: total visit count for the cell. $C(p')$: total number of robots that have covered the cell. R_i: the responsibility area (a set of cells) assigned to robot i. T_i: the target centroid for robot i's area. $d(p, p')$: Euclidean distance. $\mathrm{I}(\cdot)$: indicator function (1 if true, 0 otherwise). The immediate reward r_i for robot i is the sum of the following components:

1. Core Exploration Reward ($R_{explore}$). This is the main driver for the coverage task.

Base Exploration Reward: When the robot moves to a cell p' that is visited for the first time, let the first-visit reward term be r_{visit}=2.0, as defined in Eq. (3.10). This reward term is used to guide the robot to preferentially explore uncovered regions.

$$r_{visit} = 2.0 \cdot \mathbb{I}(V(p') = 1) \tag{3.10}$$

Large Reward for Independent Exploration: To improve collaboration efficiency, when a robot covers a cell that has not yet been covered by the team for the first time, that is, when $V(p') = 1$ and $C(p') = 1$ are both satisfied, let the independent-coverage reward component be r_{unique}=20.0. This reward term is used to encourage robots to

explore regions that have not yet been covered by the team.

$$r_{unique} = +20.0 \cdot \mathbb{I}(V(p') = 1 \wedge C(p') = 1) \tag{3.11}$$

2. Efficiency and Redundancy Penalty ($R_{penalty}$). This part penalizes inefficient and repetitive work. Revisit Penalty: When a robot moves to a previously visited cell p', the revisit reward component is defined as $r_{revisit}$. Accordingly, the absolute value of this component increases linearly with the number of visits to the cell and is used to suppress repeated movements by the robot within a local region.

$$r_{revisit} = -(V(p') - 1) \cdot 1.0 \tag{3.12}$$

Redundant Coverage Penalty: When a robot moves to a cell that has already been covered by another robot, a redundancy penalty is triggered. The penalty is defined as follows:

$$r_{redundant} = -\mathbb{I}(C(p') > 1) \cdot 2.0 \cdot (C(p') - 1) \tag{3.13}$$

3. Collaboration and Guidance Reward (R_{collab}). This part fine-tunes the robot's cooperative behavior to adhere to the dynamic region allocation. Region Consistency Reward: According to the dynamic region-allocation result, when robot i moves into its responsibility region R_i, let the region-consistency reward term be r_{region}=1.0. When the robot enters the responsibility region R_j of another robot j, where $j \neq i$, let r_{region}=−1.0. In all other cases, let r_{region}=0, as defined in Eq. (3.14).

$$r_{region} = \begin{cases} +1.0 & \text{if } p' \in R_i \\ -1.0 & \text{if } \exists j \neq i, p' \in R_j \\ 0 & \text{otherwise} \end{cases} \tag{3.14}$$

Target-Guidance Reward: To encourage robots to complete their assigned regions, the system rewards movement toward the target centroid. If the distance between the robot and the target centroid Ti of its responsibility region decreases after movement, let the target-guidance reward component be r_{target} = 1.0. If the distance increases, let r_{target} = −0.3. If the distance remains unchanged, let r_{target} = 0.

$$r_{target} = \begin{cases} +1.0 & \text{if } d(p', T_i) < d(p, T_i) \\ -0.3 & \text{if } d(p', T_i) > d(p, T_i) \\ 0 & \text{otherwise} \end{cases} \tag{3.15}$$

4. Basic Behavior and Safety Penalty (R_{safe}). These penalties discourage unsafe actions and unnecessary waiting. Wait Penalty: When a robot selects the wait action await, let the wait reward component be r_{wait} = −5.0 to reduce the probability that the

robot will continue waiting when it is unnecessary.

$$r_{wait} = -\mathbb{I}(a = a_{wait}) \cdot 5.0 \tag{3.16}$$

Invalid-Move Penalty: When a robot executes an invalid action, such as attempting to move beyond the map boundary or colliding with another robot, and its position remains unchanged after the action, that is, $p' = p$, let the invalid-move reward component be $r_{invalid} = -8.0$.

$$r_{invalid} = -\mathbb{I}(a \text{ is invalid and } p' = p) \cdot 8.0 \tag{3.17}$$

Collision Penalty: When an action executed by a robot causes it to collide with a known obstacle, let the obstacle-constraint reward component be $r_{obstacle} = -15.0$. When a diagonal movement does not satisfy the physical-feasibility condition because it is blocked by adjacent obstacles, let $r_{obstacle} = -12.0$. In all other cases, let $r_{obstacle} = 0$.

$$r_{obstacle} = \begin{cases} -15.0 & \text{if } a \text{ hits known obstacle} \\ -12.0 & \text{if } a \text{ is diagonally blocked} \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

In summary, the total reward r_i for robot i when it executes action a in state s at position p and transitions to state s′ at position p′ is the sum of all the above components:

$$r_i = R_{explore} + R_{penalty} + R_{collab} + R_{safe} \tag{3.19}$$

The definitions and coefficient values of the reward components are summarized in Table 1.

Reward Component	Current Value
First visit r_{visit}	2.0
Team first coverage r_{unique}	20.0
Revisit $r_{revisit}$	A penalty of 1.0 for each additional visit
Redundant coverage $r_{redundant}$	A penalty of 2.0 for each additional overlapping robot
Region consistency r_{region}	+1.0 in the robot's own region; −1.0 in another robot's region; 0 otherwise.
Target guidance r_{target}	1.0 when moving closer and −0.3 when moving farther away

Wait r_{wait}	-5.0
Invalid movement $r_{invalid}$	-8.0
Obstacle collision $r_{obstacle}$	-15.0 or -12.0

Table 1 Summary of Reward Parameters

By combining the four types of reward components—core exploration, efficiency and redundancy, collaborative guidance, and safety constraints—CD-DQN can learn to balance individual exploration and team collaboration from complex immediate feedback. The reward coefficients are hierarchically set according to the importance of different behaviors in the coverage task, and the method ultimately converges to an efficient, low-redundancy coverage policy. Among them, team first coverage is the core behavior that promotes global coverage and is therefore assigned a relatively large positive reward. First visits, region consistency, and target guidance are local guidance signals, and their reward magnitudes are relatively small. Waiting, invalid movement, and collisions reduce coverage efficiency or compromise motion safety, and therefore increasingly strong negative rewards are assigned to them.

3.5 Training Process

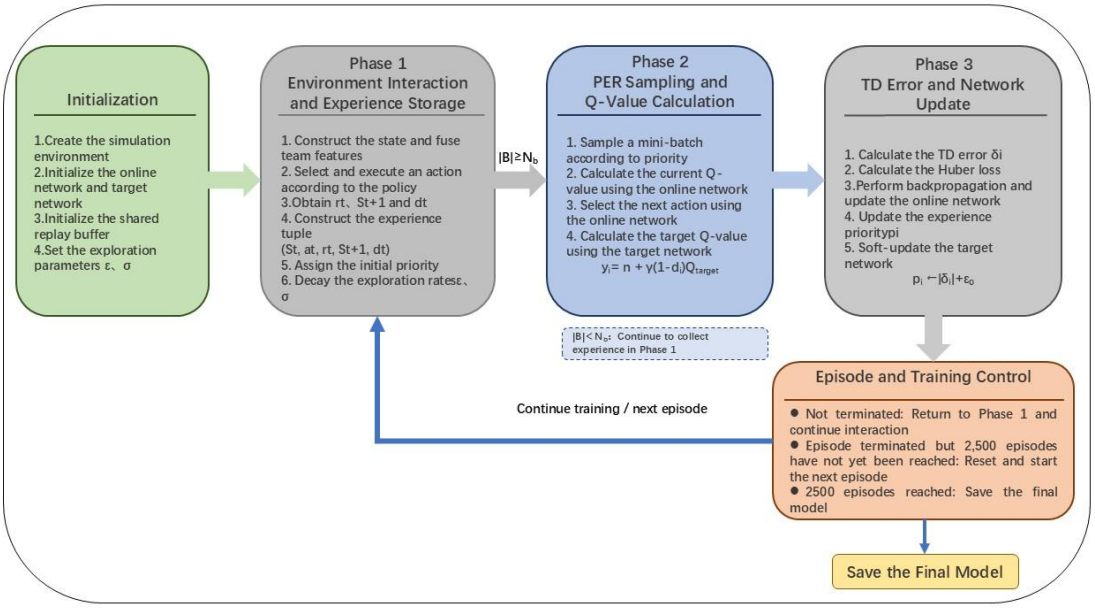


Fig. 4 Training Flowchart

As shown in Fig. 4, in addition to initialization, each training episode sequentially

executes three phases: interaction with the environment and experience storage, Prioritized Experience Replay (PER) sampling and Q-value calculation, and Temporal-Difference (TD) error calculation and network updating. These phases run continuously in a loop to iteratively improve the multi-robot collaborative policy.

Phase 1: Interaction and Experience Collection

At each time step t, the robot agents interact with the environment based on the current policy. To balance exploration and exploitation, we employ a dual-exploration mechanism. The agent selects an action a_t according to an ε-greedy policy, where the exploration rate ε decays exponentially from 1.0 to 0.01 over time. To further encourage diverse behaviors in complex scenarios, Gaussian noise N is injected into the Q-value estimates:

$$\text{Action Selection}: a_t = \arg\max_a (Q(s_t, a) + \mathrm{N}(0, \sigma)) \tag{3.20}$$

where σ is the noise standard deviation, which also decays continuously. After executing action a_t, the system receives a reward r_t (as defined in Section 3.4) and transitions to the next state s_{t+1}. This transition tuple $e_t = (s_t, a_t, r_t, s_{t+1}, \text{done})$ is stored in the Experience Replay Buffer D, which has a fixed capacity of 200,000.

Phase 2: Prioritized Experience Sampling and Current/Target Q-Value Calculation

Once the data in the buffer exceeds the batch size N (set to 128), the network update process is triggered. Instead of uniform sampling, we utilize Prioritized Experience Replay (PER) to improve learning efficiency [35]. Transitions with higher Temporal-Difference (TD) errors are considered more valuable and are assigned higher sampling probabilities $P(i)$:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}, \quad p_i = |\delta_i| + \text{ò}_0 \tag{3.21}$$

where p_i is the priority of the i-th transition, determined by its TD-error $|\delta_i|$. The hyperparameter α controls the degree of prioritization.

For each sampled mini-batch, the Current Q-value is computed by the online network using the CD-DQN architecture (including Attention and Dueling layers): $Q_{online}(s_i, a_i)$. Simultaneously, the Target Q-value is calculated using the Double DQN

technique [30] to mitigate overestimation. The online network selects the action with the highest Q-value in the next state s_{i+1}, and the target network evaluates the selected action:

$$y_i = r_i + \gamma \cdot Q_{target}(s_{i+1}, \arg\max_{a'} Q_{online}(s_{i+1}, a'; \theta); \theta^-) \cdot (1 - d_i) \tag{3.22}$$

where θ and θ^- represent the parameters of the online and target networks, respectively.

Phase 3: TD-Error Calculation and Network Update

The network is optimized by minimizing the discrepancy between the predicted Q-value and the target Q-value. We employ the Smooth L1 Loss (Huber Loss), which is robust to outliers and reduces the influence of large TD errors. The loss is weighted by importance sampling weights w_i from PER to correct the bias introduced by prioritized sampling:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} w_i \cdot \text{SmoothL1}(y_i - Q_{online}(s_i, a_i; \theta)) \tag{3.23}$$

Gradients are calculated via backpropagation, clipped to a maximum norm of 1.0, and the Adam optimizer is used to update the online network parameters θ.

Finally, to ensure training stability, the target network parameters θ^- are not updated directly but track the online network through a Soft Update mechanism. Every 500 training steps, the target network absorbs a small fraction of the online network's weights:

$$\theta^- \leftarrow \tau\theta + (1 - \tau)\theta^- \tag{3.24}$$

The soft-update coefficient τ determines how rapidly the target network tracks changes in the parameters of the online network. The soft target-network update method proposed by Lillicrap et al. [36] requires τ to be much smaller than 1 and sets it to 0.001 in the relevant experiments. Considering the relatively large fluctuations in the joint states and reward signals during multi-robot collaborative training, this study adopts a more conservative value of τ = 0.0005 within the same order of magnitude. The entire training process consists of 2,500 episodes.

4 Experiments and Discussion

4.1 Experimental Setup

To address the previously identified issues concerning adaptability to changes in team size and map, repeated-coverage control, and the use of inter-robot interaction information, this paper conducts performance tests under different map sizes and team sizes, baseline-algorithm comparisons, and key-component ablation experiments. Coverage rate, redundancy rate, and average completion steps are used to evaluate coverage completeness, task overlap, and completion efficiency, while ablation of the attention module is used to examine the role of team-interaction information fusion. In addition, Webots physics-engine simulation is used to further examine the execution feasibility of discrete coverage actions under continuous differential-drive motion.

All experiments were conducted on a high-performance computing platform equipped with an Intel i7-12700K CPU, 32 GB of RAM, and an NVIDIA RTX 3080 GPU. The software environment used Python 3.8.5, PyTorch 1.12.0, and CUDA 11.3 for GPU acceleration. Visualization and data analysis were performed using Matplotlib 3.5.2 and NumPy 1.21.0. The physics-engine simulation uses Webots R2021a and the E-puck differential-drive robot model.

The training environment is designed as a 30×30 grid map to simulate coverage scenarios in real physical space. Fifty to ninety obstacles are randomly distributed in the map, and a connectivity check ensures that all free grid cells are mutually connected. A variable number n ($2 \leq n \leq 9$) of homogeneous mobile robots are used for collaborative coverage. The robots are simplified as point robots occupying a single grid cell; the maximum movement distance per step is one grid cell, the local sensing radius is set to one grid cell, and the communication radius is set to 80 grid-cell units. Both the number of robots and the map are randomly generated in each episode. The model is trained for a total of 2,500 episodes. In each episode, five training iterations are performed, and the maximum number of steps per iteration is set to 1,500.

4.2 Learning Process and Scale Adaptability Analysis

4.2.1 Learning Curve Analysis

To evaluate the performance of the system, we define the following key metrics based on the environment's state at the end of each episode. Let $C(p)$ be the number

of robots that covered cell p:

Coverage Rate: The ratio of unique free cells covered by at least one robot ($C(p) \geq 1$) to the total number of free cells in the map.

$$\text{Coverage Rate} = \frac{|\text{Cells}_{\text{Covered}}|}{|\text{Cells}_{\text{Free}}|} \tag{4.1}$$

Redundancy Rate: The ratio of cells covered by more than one robot ($C(p) > 1$) to the total number of covered cells. This measures the degree of wasted effort.

$$\text{Redundancy Rate} = \frac{|\text{Cells}_{\text{Redundant}}|\,(\text{C(p)} > 1)}{|\text{Cells}_{\text{Covered}}|} \tag{4.2}$$

Collaboration Efficiency: The ratio of cells covered by exactly one robot (C(p) = 1) to the total number of covered cells. This is the complement of the Redundancy Rate and measures the system's ability to effectively divide labor.

$$\text{Collaboration Efficiency} = \frac{|\text{Cells}_{\text{Effective}}|\,(\text{C(p)} = 1)}{|\text{Cells}_{\text{Covered}}|} \tag{4.3}$$

Completion Steps: The number of steps consumed when each coverage task ends, used to measure the overall efficiency of the multi-robot system.

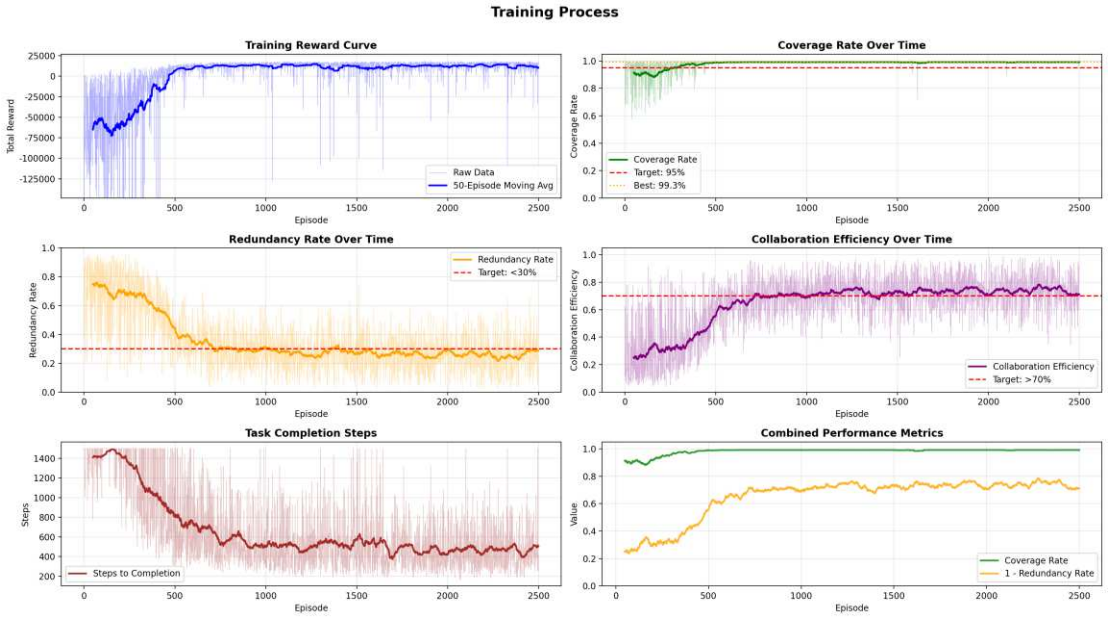


Fig. 5 Training Process of the Shared-Weight Model

Fig. 5 shows the evolution of the key performance indicators of CD-DQN during 2500 training episodes. Three distinct stages can be observed from the learning curves.

The exploration stage mainly occurs during Episodes 0-500. During this stage, the robots mainly perform random exploration and learn basic movement and obstacle-avoidance skills. The coverage rate shows a steady upward trend, but the redundancy rate remains high. The optimization stage occurs approximately during Episodes 500-1,500. During this stage, the coverage rate stabilizes above 99%. As training progresses, the robots gradually learn to make coverage decisions under the guidance of regional tasks, and the redundancy rate and completion steps generally decrease. The convergence stage occurs approximately during Episodes 1,500-2,500. All indicators begin to stabilize, and the system exhibits stable, efficient, and collaborative behavior. While the coverage rate remains at a high level, the redundancy rate and completion steps generally decrease, indicating that the training process not only improves coverage completeness but also gradually improves repeated-coverage control and task-completion efficiency.

4.2.2 Final Performance Test

The optimally trained model was used to conduct 100 independent tests with different numbers of robots, with a different random map and initial-position configuration used in each test. To examine the adaptability of the learned policy to changes in map size and team size, the coverage rate, collaboration efficiency, average completion steps, and redundancy rate under different configurations were calculated. The results are shown in Table 2.

map size	number of robots	coverage rate	collaboration efficiency	average completion steps	redundancy rate
30×30	3	99.8%	90.9%	618	9.1%
	4	99.1%	86.7%	385	13.3%
	5	99.1%	84.6%	325	15.4%
35×35	3	99.1%	89.0%	764	11.0%
	4	99.1%	88.4%	534	11.6%
	5	99.1%	87.0%	472	13.0%
40×40	3	99.5%	81.6%	1152	18.4%
	4	99.1%	85.8%	774	14.2%
	5	99.1%	79.2%	676	20.8%

Table 2 Performance of CD-DQN under Different Map Sizes and Team Sizes

Table 2 shows that, under the test configurations of 30×30, 35×35, and 40×40 maps with three to five robots, the coverage rate of CD-DQN remains within 99.1%–99.8%, indicating that the learned policy can maintain high coverage completeness when the map size and team size change. For the same map size, increasing the number of robots reduces the average completion steps, while the redundancy rate also exhibits stable performance. For example, on the 30×30 map, when the number of robots increases from three to five, the average completion steps decrease from 618 to 325, while the redundancy rate increases from 9.1% to 15.4%.

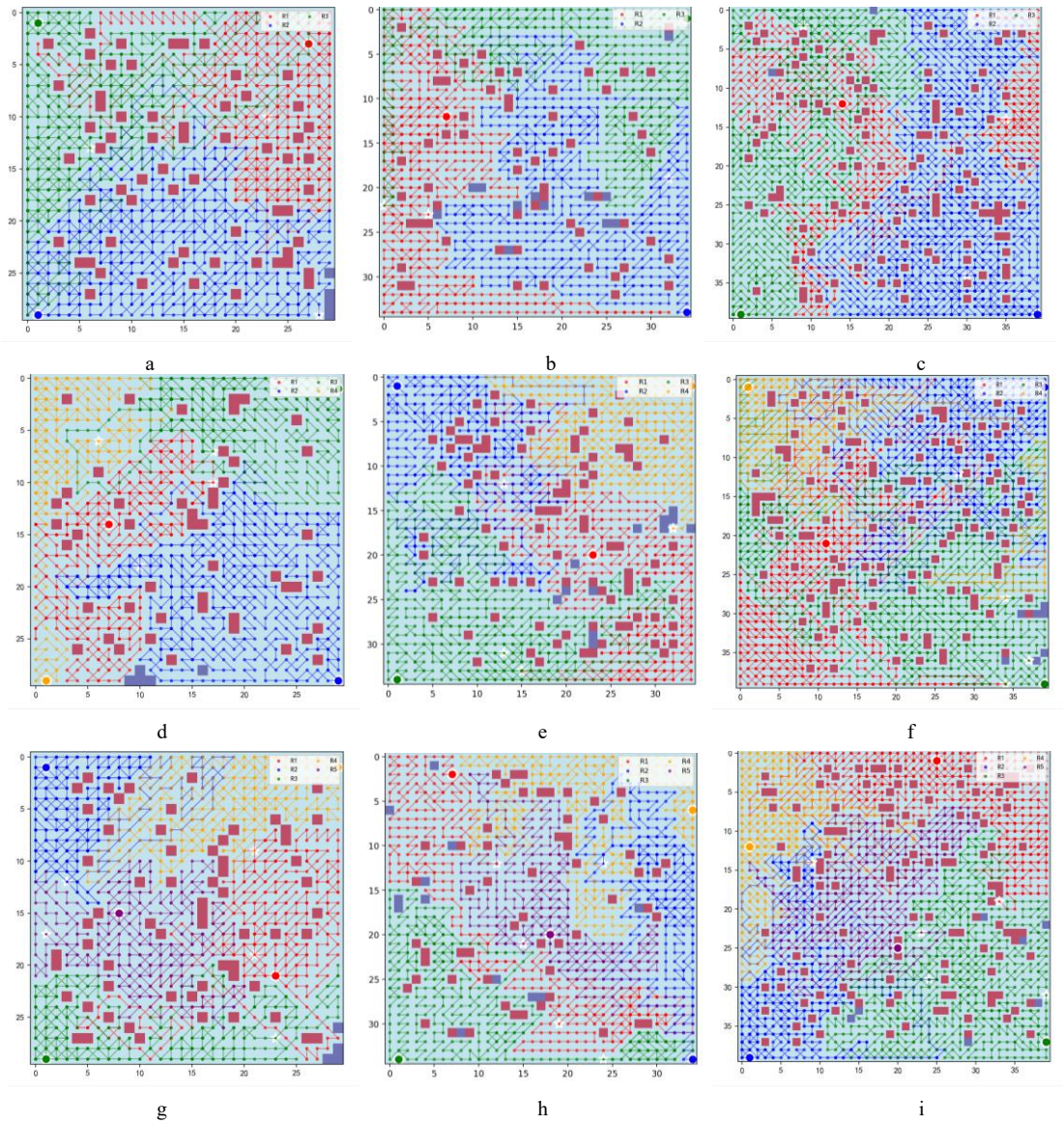


Fig. 6 Robot Trajectories. (a-c) 3 robots on 30×30, 35×35, 40×40 maps. (d-f) 4 robots on 30×30, 35×35, 40×40 maps. (g-i) 5 robots on 30×30, 35×35, 40×40 maps

Figures 6(a)–(i) show representative coverage trajectories under different combinations of map size and robot number. The paths of different colors, representing different robots, show a clear pattern of cooperation at the macroscopic level. Overlap in trajectories primarily occurs at the boundaries of different regions, indicating that the robots are effectively interacting and yielding at these borders to ensure seamless coverage, rather than creating chaotic overlap in the center of the regions. This efficient collaborative behavior emerges naturally in all scenarios, whether with 3, 4, or 5 robots. This strongly demonstrates that, by training with a variable number of robots, CD-DQN has learned a collaborative strategy that adapts to different team sizes and exhibits

excellent generalization ability. Fig. 6 provides an intuitive display of the coverage trajectories for the different configurations in Table 2 and is used to observe the spatial distribution and overlap of the paths of the individual robots. The coverage completeness and degree of repeated coverage under different configurations are evaluated in conjunction with Table 2, while the specific effects of team-information fusion and region allocation on collaborative performance are further analyzed through the ablation experiments in Section 4.4.

4.2.3 Comprehensive Performance Analysis

To provide a holistic view of the system's efficiency across the two key variables of team size and environment complexity, we define a unified Performance Score. This metric combines the three primary performance indicators: Coverage Rate (C), Redundancy Rate (R), and average completion steps (S). The score is calculated using the weighted formula:

$$\text{Score} = 0.5 \times \text{C} + 0.25 \times (1 - \text{R}) + 0.25 \times \left(\frac{\text{max_steps - S}}{\text{max_steps}} \right) \tag{4.4}$$

A simulation study was conducted with the number of robots set to n ∈ {2, 3, …, 9} and the map size set to H×W ∈ {20×20, …, 40×40}. The resulting performance distribution is shown in Fig. 7.

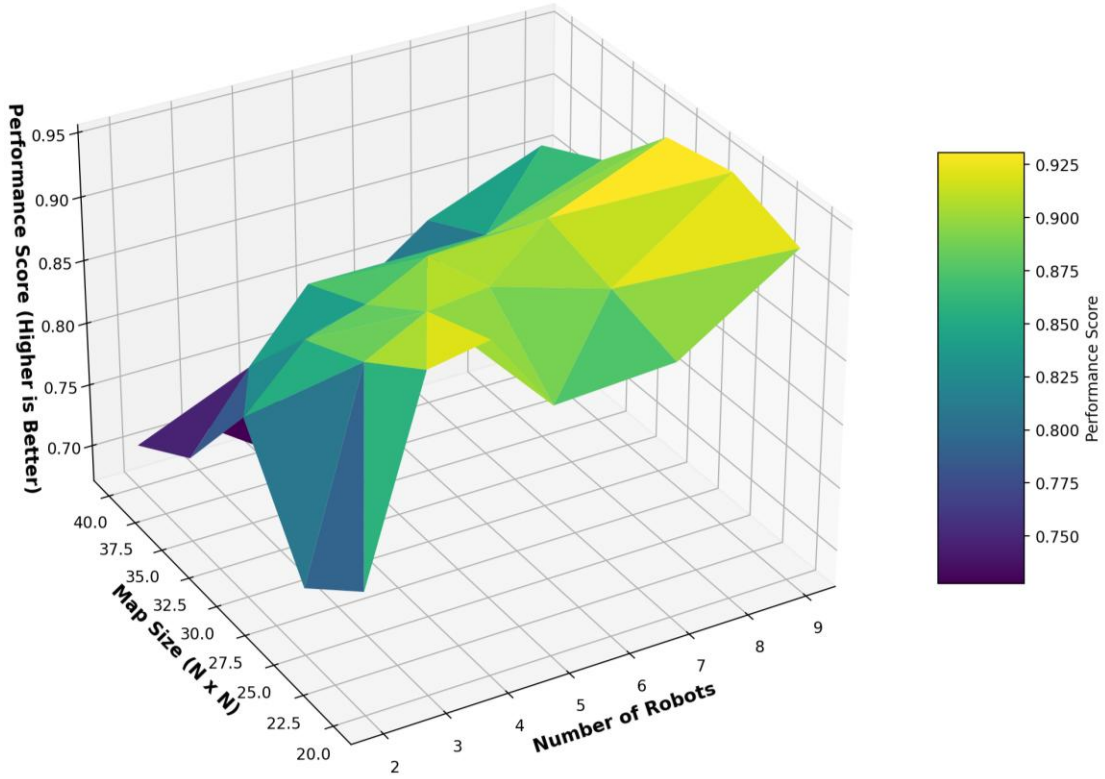


Fig. 7 Variation in System Performance with Team Size and Map Size. The Z-axis represents the comprehensive performance score, and a higher value indicates better overall performance

The 3D plot in Fig. 7 reveals the complex relationship between team size and environment scale. Generally, for a fixed map size, increasing the number of robots improves the performance score by significantly reducing completion time. The highest performance (score > 0.925) is achieved on larger maps (e.g., 35×35 to 40×40) with a sufficient number of robots (7–9), demonstrating the system's good adaptability to changes in scale. The plot also highlights critical trade-offs. A dip in performance is observable around 5 robots on a 30×30 map. This suggests that adding too many robots to a relatively small environment can increase redundancy, thereby lowering the overall efficiency score. Conversely, using too few robots for a large map (e.g., 2–3 robots on a 35×35 map) also results in a lower score, as the task becomes too difficult to complete efficiently. Fig. 7 shows that the comprehensive performance of CD-DQN varies with the combination of team size and map size, and the coverage performance is jointly affected by the task scale and the robot-number configuration. Combined with Table 2, it can be seen that the method maintains a high coverage rate under the tested

configurations, while the repeated-coverage level and completion-step data also demonstrate excellent performance.

4.3 Comparative Experiments

In this section, 100 independent tests are conducted under the same testing conditions. CD-DQN is compared with MADRL, MACPP, APF-CPP, and DQN in terms of coverage rate, redundancy rate, collaboration efficiency, and average completion steps in unknown environments. Two comparison scenarios are considered: n = 3 with a 30×30 map and n = 4 with a 40×40 map. The quantitative comparison results are summarized in Table 3, and the metric-wise performance comparison is visualized in Fig. 8.

Scenario	Algorithm	coverage rate	collaboration efficiency	average completion steps	redundancy rate	performance score
n = 3 H×W = 30×30	CD-DQN	99.8%	90.9%	618	9.1%	0.873
	MADRL	99.2%	85.5%	692	14.5%	0.844
	APF-CPP	98.4%	81.7%	753	18.3%	0.820
	MACPP	98.8%	89.2%	824	10.8%	0.830
	DQN	94.9%	65.8%	1050	34.2%	0.714
n = 4 H×W = 40×40	CD-DQN	99.1%	85.8%	774	14.2%	0.831
	MADRL	98.9%	82.6%	844	17.4%	0.810
	APF-CPP	97.8%	79.9%	910	20.1%	0.787
	MACPP	98.2%	86.9%	940	13.1%	0.802
	DQN	95.2%	64.3%	1140	35.7%	0.700

Table 3 Quantitative performance comparison between CD-DQN and the baseline algorithms.

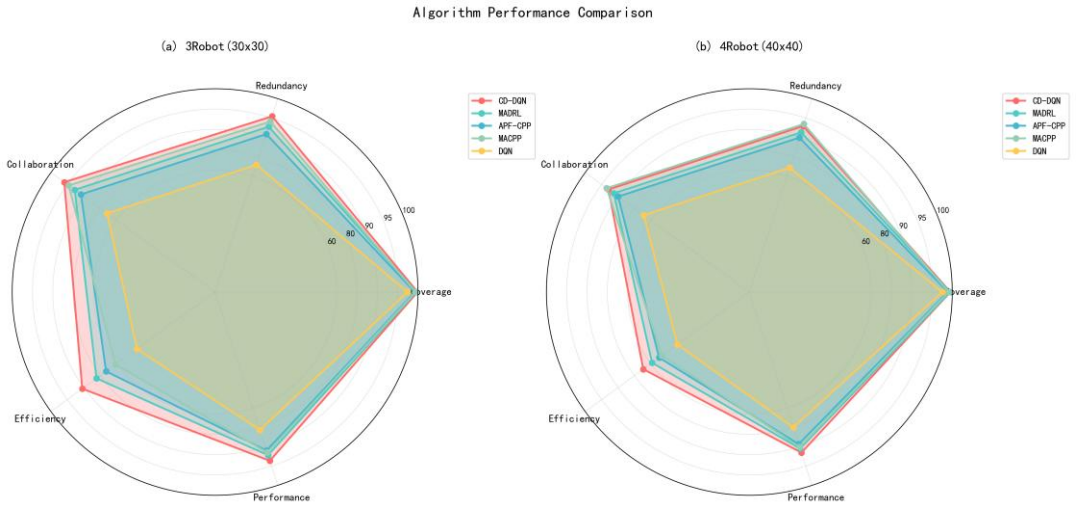


Fig. 8 Performance comparison of CD-DQN and the baseline algorithms: **(a)** three robots on the 30×30 map; **(b)** four robots on the 40×40 map

As shown in Table 3 and Fig. 8, CD-DQN achieves the smallest average number of completion steps in both test scenarios. Compared with MADRL, which ranks second in average completion steps, CD-DQN reduces the average completion steps on the 30×30 and 40×40 maps by approximately 10.7% and 8.3%, respectively, demonstrating higher coverage-task completion efficiency. In terms of collaboration and redundancy, on the 30×30 map, both the collaboration efficiency (90.9%) and redundancy rate (9.1%) of CD-DQN are at the best levels; on the 40×40 map, although MACPP has a slightly lower redundancy rate, CD-DQN achieves a coverage rate of 99.1%, higher than MACPP's 98.2%, while the average completion steps decrease from 940 to 774, a reduction of approximately 17.7%, and the comprehensive performance score increases from 0.802 to 0.831. Compared with the DQN baseline, CD-DQN achieves significant improvements in all evaluation metrics in both test scenarios. Specifically, on the 30×30 map, the coverage rate increases from 94.9% to 99.8%, the redundancy rate decreases from 34.2% to 9.1%, and the average completion steps decrease from 1,050 to 618, a reduction of approximately 41.1%. Across the two comparison scenarios, CD-DQN achieves the highest coverage rate, the smallest average completion steps, and the highest comprehensive performance score, indicating that the proposed framework effectively improves the problems of repeated coverage and insufficient completion efficiency and achieves simultaneous improvements in coverage completeness and task-completion efficiency.

4.4 Ablation Study

To evaluate the individual contributions of the key components of CD-DQN to coverage performance, ablation experiments are conducted while keeping the number of training episodes, network size, optimizer parameters, map set, and number of robots unchanged. The complete CD-DQN is used as the reference model, and four ablation variants are constructed: CD-DQN-w/o Attention removes the inter-robot multi-head collaborative attention module; CD-DQN-w/o Dueling replaces the Dueling structure with a single action-value stream; CD-DQN-w/o Region disables the dynamic region-allocation strategy; and CD-DQN-w/o Reward removes the multi-level reward-shaping terms designed in this paper while retaining only the basic rewards required for task learning. The attention-module ablation is used to evaluate the effect of team-interaction information fusion on coverage decision-making, whereas the region-allocation and reward-mechanism ablations are used to evaluate the effects of task division and reward guidance on suppressing repeated coverage and improving task-completion efficiency.

Model	Coverage rate/%	Collaboration efficiency/%	Average completion steps	Redundancy rate/%
CD-DQN	99.8	90.9	618	9.1
w/o Attention	98.1	74.5	792	25.5
w/o Dueling	99.1	85.7	698	14.3
w/o Dynamic Region	97.8	69.8	856	30.2
w/o Reward	98.4	86.3	729	13.7

Table 4 Ablation study results on the effectiveness of key framework components.

As shown in Table 4, the complete CD-DQN achieves the best results in all four indicators: coverage rate, collaboration efficiency, average steps, and redundancy rate. After the inter-robot multi-head collaborative attention module is removed, the coverage rate decreases from 99.8% to 98.1%, the redundancy rate increases from 9.1% to 25.5%, and the average completion steps increase from 618 to 792. These results indicate that the attention module is an important component for improving collaborative coverage performance. By incorporating team-interaction information into individual action-value estimation, CD-DQN effectively reduces repeated

coverage among robots and improves task-completion efficiency, thereby verifying the effectiveness of the proposed interaction-information fusion mechanism. After the Dueling structure is removed, the coverage rate decreases by 0.7 percentage points, the average completion steps increase by 80, and the redundancy rate increases by 5.2 percentage points, indicating that separating the state value from the action advantage improves action-selection efficiency to a certain extent. After the dynamic region-allocation strategy is disabled, the coverage rate decreases by 2.0 percentage points, the collaboration efficiency decreases by 21.1 percentage points, the average completion steps increase by 238, and the redundancy rate increases by 21.1 percentage points. This is the largest performance degradation among all the ablation models, indicating that the dynamic region-allocation strategy plays an important role in coordinating robot task division and suppressing repeated coverage. After the multi-level reward shaping is removed, the coverage rate decreases by 1.4 percentage points, the average completion steps increase by 111, and the redundancy rate increases by 4.6 percentage points. These results indicate that the multi-level reward mechanism guides the robots to reduce invalid movement and improve coverage-path execution efficiency. Overall, the complete CD-DQN outperforms all the ablation models, verifying the contribution of each component to the overall performance. Among these components, dynamic region allocation plays the most prominent role in coordinating task division and suppressing repeated coverage, inter-robot multi-head collaborative attention improves coverage decisions by fusing team-interaction information, and the multi-level reward mechanism further guides the robots to reduce repeated coverage and improve completion efficiency. Together, these mechanisms enable CD-DQN to reduce the redundancy rate and completion steps while maintaining a high coverage rate, demonstrating the comprehensive advantages of the proposed collaborative framework.

4.5 Coverage Execution Simulation Using the Webots Physics Engine

To further examine the execution feasibility of the proposed CD-DQN coverage policy under continuous-motion and physical collision constraints, a multi-robot coverage simulation scenario is constructed using Webots R2021a. The physical field is set to 3 m × 3 m and corresponds to a 30 × 30 grid map, with each grid cell representing a physical area of 0.1 m × 0.1 m. Three E-puck differential-drive robots are deployed in the scenario, and several static WoodenBox objects are used to construct

the obstacle environment.

The simulation system adopts a centralized-decision and distributed-execution structure. The Webots Supervisor node is responsible for maintaining the environmental state, loading the trained CD-DQN model, and constructing the 106-dimensional state vector according to Section 3.3. The model outputs Q-values for eight movement actions toward adjacent grid cells and one wait action. A movement action specifies an adjacent target grid cell, whereas the wait action causes the robot to remain stationary. The selected actions are transmitted to the individual E-puck controllers through the Emitter/Receiver communication channels and are then converted by the low-level controllers into left- and right-wheel velocity commands.

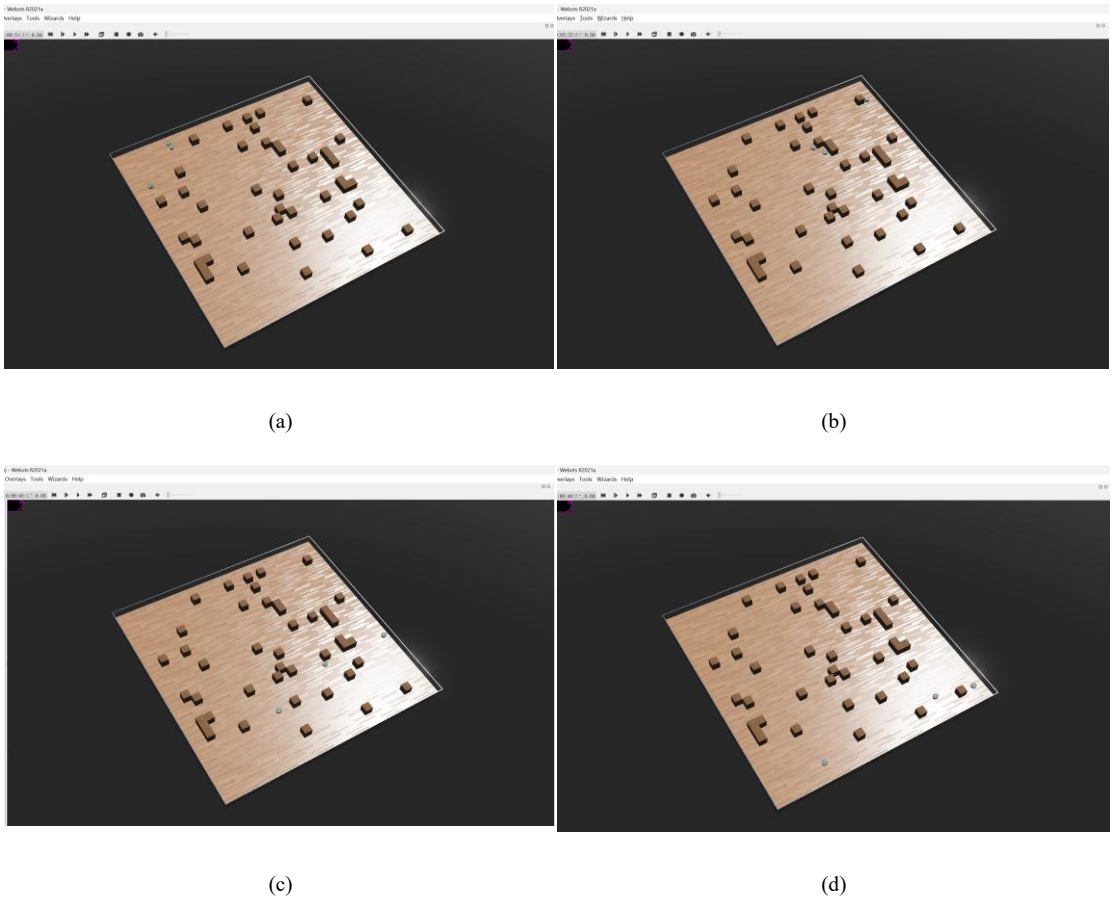
$$X_t = \left(c_t + \frac{1}{2}\right)\Delta\, d - \frac{L}{2}, \qquad Z_t = \left(r_t + \frac{1}{2}\right)\Delta d - \frac{\mathrm{L}}{2} \tag{4.5}$$

where r_t and c_t denote the row and column indices of the target grid cell, respectively; X_t and Z_t denote the physical coordinates of the center of that cell in the Webots ground coordinate system; L = 3.0 m is the side length of the field; and Δd = 0.1 m is the grid resolution.

After the target physical coordinates are obtained, the low-level controller calculates the target heading and heading error according to the robot's current position and orientation. When the absolute heading error exceeds the preset threshold, the robot first turns in place by driving the left and right wheels in opposite directions. After the heading is aligned, the robot moves toward the target position by driving the two wheels in the same direction. When the robot enters the preset target-position tolerance range, the controller returns an arrival state. After all robots have completed their current actions, the Supervisor updates the environmental state and triggers the next CD-DQN decision, thereby forming a closed-loop simulation process of “state acquisition–action decision–continuous execution–synchronous update.”

To verify the execution capability of the CD-DQN coverage policy under continuous-motion conditions, four representative moments from the same Webots simulation are selected to illustrate the coverage process of the three robots from discrete action decision-making to continuous differential-drive motion, as shown in

Fig. 9.



(a) (b)

(c) (d)

Fig. 9 Coverage Execution Process of Three E-puck Robots in the Webots Environment: (a) simulation time of 51.3 s and coverage rate of 10.6%; (b) simulation time of 216.1 s and coverage rate of 40.2%; (c) simulation time of 405.6 s and coverage rate of 70.2%; and (d) simulation time of 520.3 s and coverage rate of 90.7%

As shown in Fig. 9, as the simulation proceeds, the scene coverage rate increases from 10.6% at 51.3 s to 90.7% at 520.3 s, and the three robots continuously execute the coverage task in the static-obstacle environment. With continuous differential-drive chassis motion, robot collision volume, and action execution time taken into account, the system establishes a closed-loop connection between the coverage policy and continuous motion control through grid-target mapping, low-level differential-drive control, and synchronous state updating. The simulation results verify the execution feasibility of the discrete CD-DQN coverage actions in the constructed physics-engine scenario and demonstrate the application potential of combining the proposed framework with differential-drive robot motion control.

5 Conclusion

To address the limited adaptability to changes in team size and map, insufficient repeated-coverage control, and inadequate use of inter-robot interaction information in multi-robot coverage tasks in unknown environments, this paper proposes a Collaborative Deep Q-Network (CD-DQN) coverage path planning framework. The framework improves policy adaptability to different task configurations through training with variable numbers of robots and random maps; it fuses team-state and task information through inter-robot multi-head collaborative attention and combines a Dueling structure for action-value estimation, allowing team-interaction information to participate in individual coverage decisions; and it provides task division through dynamic region allocation and combines a multi-level reward mechanism to guide robots to explore uncovered regions and reduce repeated coverage. Experimental results show that, in the performance tests on 30×30 – 40×40 maps with three to five robots, the coverage rate of CD-DQN remains within 99.1% – 99.8%, indicating that it can maintain high coverage completeness under the tested changes in map size and team size. In the comparison scenario with a 30×30 map and three robots, compared with DQN, the coverage rate of CD-DQN increases from 94.9% to 99.8%, the redundancy rate decreases from 34.2% to 9.1%, and the average completion steps decrease from 1,050 to 618, demonstrating joint improvements in coverage completeness, repeated-coverage control, and completion efficiency. The ablation experiments further show that, after the inter-robot multi-head collaborative attention module is removed, the redundancy rate increases from 9.1% to 25.5% and the average completion steps increase by 174, supporting the effectiveness of team-interaction information fusion in collaborative coverage decision-making. Disabling dynamic region allocation causes the most significant performance degradation, indicating that regional task division is an important component of the current framework for reducing repeated coverage, while the multi-level reward mechanism further improves coverage performance. On this basis, a coverage scenario involving three E-puck robots is further constructed in Webots, and the discrete actions output by CD-DQN are converted into continuous differential-drive motion commands, verifying the execution feasibility of the coverage decisions in a physics-engine environment.

The experimental validation scope of this paper covers 20×20 – 40×40 static-

obstacle grid maps, two to nine homogeneous robots, and fully connected communication conditions. Cases involving larger maps, more robots, limited communication, dynamic obstacles, and robot failures still require further targeted validation. Future work will focus on the following aspects: (1) extending the framework to support larger robot teams and more complex unstructured environments in order to verify the scalability and stability of the algorithm under high-load conditions; (2) introducing a collaboration mechanism for heterogeneous robot teams and optimizing task allocation among robots with different functions, such as collaboration between ground robots and aerial robots; (3) because the current study mainly considers static obstacles, investigating online adaptability to effectively address environments containing dynamic obstacles and ensure the safety and efficiency of the algorithm in real dynamic scenarios; and (4) exploring deployment methods for the proposed policy on physical robot platforms and further examining the effects of real sensor noise, localization errors, communication delays, and chassis-execution errors on coverage performance.

Statements and Declarations

Author Contributions: Wan Xu: Conceptualization, Methodology, Writing—Review & Editing. Jinduan Liu: Methodology, Software, Writing—original draft, visualization. Yujie Wang: Data curation, software. Shize Wang: Data curation, software.

Funding: This research received startup funding from the Key Research and Development Program of Hubei Province under grant no. 2023BEB031.

Competing Interests: The authors declare that they have no competing interests.

Data Availability Statement: The original contributions presented in the study are included in the article material; further inquiries can be directed to the corresponding author.

Ethical Approval: Not applicable.

References

[1] Lin, H.-Y., Huang, Y.-C.: Collaborative complete coverage and path planning for multi-robot exploration. Sensors 21(11), 3709 (2021). https://doi.org/10.3390/s21113709

[2] Tao, H., Zhao, W., Zhao, L., Wang, J.: Research on disaster environment map fusion construction and reinforcement learning navigation technology based on air–ground collaborative multi-heterogeneous robot systems. Sensors 25(16), 4988 (2025). https://doi.org/10.3390/s25164988

[3] Lei, Y., Hou, J., Ma, P., Ma, M.: Voronoi-GRU-based multi-robot collaborative exploration in unknown environments. Applied Sciences 15, 3313 (2025). https://doi.org/10.3390/app15063313

[4] Song, J., Gupta, S.: CARE: Cooperative Autonomy for Resilience and Efficiency of robot teams for complete coverage of unknown environments under robot failures. Autonomous Robots 44(3–4), 647–671 (2020). https://doi.org/10.1007/s10514-019-09870-3

[5] Almadhoun, R., Taha, T., Seneviratne, L., Zweiri, Y.: Multi-robot hybrid coverage path planning for 3D reconstruction of large structures. IEEE Access 10, 2037–2050 (2022). https://doi.org/10.1109/ACCESS.2021.3139080

[6] Mitra, R., Saha, I.: Online concurrent multi-robot coverage path planning. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 8691–8698. IEEE (2025). https://doi.org/10.1109/IROS60139.2025.11247261

[7] Nair, V.G., Dileep, M.V., Guruprasad, K.R.: Robust online multi-robot simultaneous exploration and coverage path planning. IEEE Access 12, 72990–73003 (2024). https://doi.org/10.1109/ACCESS.2024.3403658

[8] Li, W., Zhao, T., Dian, S.: Multirobot coverage path planning based on deep Q-network in unknown environment. Journal of Robotics 2022, 6825902 (2022). https://doi.org/10.1155/2022/6825902

[9] Sameena, S., Nallasivan, G.: Dynamic multi-robot path planning: A comparative perspective. In: 2025 4th International Conference on Innovative Mechanisms for Industry Applicatio

ns (ICIMIA), pp. 105–111. IEEE (2025). https://doi.org/10.1109/ICIMIA67127.2025.11200646

[10] Ramesh, M., Imeson, F., Fidan, B., Smith, S.L.: Minimum-length coverage path planning for grid environments with approximation guarantees. IEEE Robotics and Automation Letters 10(10), 10674–10681 (2025). https://doi.org/10.1109/LRA.2025.3604732

[11] Wang, Z., Lyu, X., Zhang, J., Wang, P., Zhong, Y., Shi, L.: MAC-Planner: A novel task allocation and path planning framework for multi-robot online coverage processes. IEEE Robotics and Automation Letters 10(5), 4404–4411 (2025). https://doi.org/10.1109/LRA.2025.3551078

[12] Jeong, S., You, K., Seok, D.: Hazardous flight region prediction for a small UAV operated in an urban area using a deep neural network. Aerospace Science and Technology 118, 107060 (2021). https://doi.org/10.1016/j.ast.2021.107060

[13] Wang, B., Zhang, Y., Zhang, W.: Integrated path planning and trajectory tracking control for quadrotor UAVs with obstacle avoidance in the presence of environmental and systematic uncertainties: Theory and experiment. Aerospace Science and Technology 120, 107277 (2022). https://doi.org/10.1016/j.ast.2021.107277

[14] Tang, J., Gao, Y., Lam, T.L.: Learning to coordinate for a worker-station multi-robot system in planar coverage tasks. IEEE Robotics and Automation Letters 7(4), 12315–12322 (2022). https://doi.org/10.1109/LRA.2022.3214446

[15] Sun, C., Tang, J., Zhang, X.: FT-MSTC*: An efficient fault tolerance algorithm for multi-robot coverage path planning. In: 2021 IEEE International Conference on Real-time Computing and Robotics (RCAR), pp. 107–112. IEEE (2021). https://doi.org/10.1109/RCAR52367.2021.9517650

[16] Chen, J., Ling, F., Zhang, Y., You, T., Liu, Y., Du, X.: Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system. Swarm and Evolutionary Computation 69, 101005 (2022). https://doi.org/10.1016/j.swevo.2021.101005

[17] Zhou, X., Wang, H., Ding, B., Hu, T., Shang, S.: Balanced connected task allocations for multi-robot systems: An exact flow-based integer program and an approximate tree-based genetic algorithm. Expert Systems with Applications 116, 10–20 (2019). https://doi.org/10.1016/j.eswa.2018.09.001

[18] Fevgas, G., Lagkas, T., Argyriou, V., Sarigiannidis, P.: Coverage path planning methods focusing on energy efficient and cooperative strategies for unmanned aerial vehicles. Sensors 22(3), 1235 (2022). https://doi.org/10.3390/s22031235

[19] Lu, J., Zeng, B., Tang, J., Lam, T.L., Wen, J.: TMSTC*: A path planning algorithm for minimizing turns in multi-robot coverage. IEEE Robotics and Automation Letters 8(8), 5275–5282 (2023). https://doi.org/10.1109/LRA.2023.3293319

[20] Li, L., Shi, D., Jin, S., Yang, S., Zhou, C., Lian, Y., Liu, H.: Exact and heuristic multi-robot Dubins coverage path planning for known environments. Sensors 23, 2560 (2023). https://doi.org/10.3390/s23052560

[21] Miao, X., Lee, H.-S., Kang, B.-Y.: Multi-cleaning robots using cleaning distribution method based on map decomposition in large environments. IEEE Access 8, 97873–97889 (2020). https://doi.org/10.1109/ACCESS.2020.2997095

[22] Wang, C., Dong, W., Li, R., Dong, H., Liu, H., Gao, Y.: An improved STC-based full coverage path planning algorithm for cleaning tasks in large-scale unstructured social environments. Sensors 24, 7885 (2024). https://doi.org/10.3390/s24247885

[23] Tang, J., Ma, H.: Mixed integer programming for time-optimal multi-robot coverage path planning with efficient heuristics. IEEE Robotics and Automation Letters 8(10), 6491–6498 (2023). https://doi.org/10.1109/LRA.2023.3306996

[24] Jiao, L., Peng, Z., Xi, L., Ding, S., Cui, J.: Multi-agent coverage path planning via proximity interaction and cooperation. IEEE Sensors Journal 22(6), 6196–6207 (2022). https://do

i.org/10.1109/JSEN.2022.3150098
[25] Yuan, G., Xiao, J., He, J., Jia, H., Wang, Y., Wang, Z.: Multi-agent cooperative area cove rage: A two-stage planning approach based on reinforcement learning. Information Science s 678, 121025 (2024). https://doi.org/10.1016/j.ins.2024.121025
[26] Wang, Z., Zhao, X., Zhang, J., Yang, N., Wang, P., Tang, J., Zhang, J., Shi, L.: APF-CP P: An artificial potential field based multi-robot online coverage path planning approach. I EEE Robotics and Automation Letters 9(11), 9199–9206 (2024). https://doi.org/10.1109/LR A.2024.3432351
[27] Yu, L., Su, X., Tian, Y., Huang, J., Shen, C.: Multirobot target searches in unknown envi ronments via waypoint planning system. IEEE Transactions on Industrial Electronics 71(7), 7511–7519 (2024). https://doi.org/10.1109/TIE.2023.3301541
[28] Zhao, B., Huo, M., Li, Z., Yu, Z., Qi, N.: Clustering-based hyper-heuristic algorithm for multi-region coverage path planning of heterogeneous UAVs. Neurocomputing 610, 128528 (2024). https://doi.org/10.1016/j.neucom.2024.128528
[29] Mnih, V., Kavukcuoglu, K., Silver, D., et al.: Human-level control through deep reinforce ment learning. Nature 518, 529–533 (2015). https://doi.org/10.1038/nature14236
[30] van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double Q-learning. Proceedings of the AAAI Conference on Artificial Intelligence 30(1), 2094–2100 (2016). https://doi.org/10.1609/aaai.v30i1.10295
[31] Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., de Freitas, N.: Dueling ne twork architectures for deep reinforcement learning. In: Proceedings of the 33rd Internation al Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 1995–2003. PMLR (2016). https://proceedings.mlr.press/v48/wangf16.html
[32] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., P olosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Sy stems, vol. 30, pp. 5998–6008 (2017). https://papers.nips.cc/paper/7181-attention-is-all-you-n eed
[33] Iqbal, S., Sha, F.: Actor-attention-critic for multi-agent reinforcement learning. In: Proceedi ngs of the 36th International Conference on Machine Learning. Proceedings of Machine L earning Research, vol. 97, pp. 2961–2970. PMLR (2019). https://proceedings.mlr.press/v97/i qbal19a.html
[34] Ayman, N., Esmail, S.A., Hamdi, A., Shaban, K., Kassab, H.: MHA-DQN: Personalized r oute planning for asthma patients using multi-head attention and deep reinforcement learni ng. In: 2025 IEEE/ACS 22nd International Conference on Computer Systems and Applicat ions (AICCSA), pp. 1–8. IEEE (2025). https://doi.org/10.1109/AICCSA66935.2025.11315371
[35] Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. In: Internatio nal Conference on Learning Representations (ICLR) (2016). https://arxiv.org/abs/1511.05952
[36] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. In: International Conference on Learning Representations (ICLR) (2016). https://arxiv.org/abs/1509.02971
[37] Tai, L., Paolo, G., Liu, M.: Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In: 2017 IEEE/RSJ International Conference on I ntelligent Robots and Systems (IROS), pp. 31–36. IEEE (2017). https://doi.org/10.1109/IRO S.2017.8202134

[38] Long, P., Fan, T., Liao, X., Liu, W., Zhang, H., Pan, J.: Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In: 2018 IEEE Internationa l Conference on Robotics and Automation (ICRA), pp. 6252–6259. IEEE (2018). https://doi.org/10.1109/ICRA.2018.8461113

[39] Chen, C., Liu, Y., Kreiss, S., Alahi, A.: Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 6015–6022. IEEE (2019). https://doi.org/10.1109/ICRA.2019.8794134



Wan Xu Graduated from Wuhan University of Technology in 2002 with a Bachelor's degree in Automotive Engineering. Obtained a Doctorate in Mechatronics Engineering from Huazhong University of Science and Technology in 2009. Currently serves as Professor at the School of Mechanical Engineering, Hubei University of Technology. Recent research focuses on mobile robotics, distributed multi-axis motion control, industrial Ethernet technology, and wireless sensor networks. He has led one National Natural Science Foundation of China project and five provincial/ministerial-level research initiatives. As a principal contributor, he has participated in and completed over ten vertical research projects, including National Natural Science Foundation grants and provincial/ministerial-level programmes. Collaborating with industry partners, he has delivered multiple commissioned research projects. His academic output comprises over 30 published papers, alongside five granted invention patents and four software copyright registrations.



Jinduan Liu is currently pursuing a master's degree at the School of Mechanical Engineering, Hubei University of Technology, Wuhan, China. His research interests include multi-robot coverage path planning, deep reinforcement learning, collaborative decision-making, and task allocation. His research focuses on attention-based inter-robot information fusion, dynamic region allocation, and reward mechanism design to improve coverage efficiency, reduce redundant exploration, and enhance adaptability to varying team sizes and environments. He received the First Prize in the 14th Hubei Provincial "Challenge Cup" College Student Entrepreneurship Plan Competition.



Yujie Wang Master's candidate at Hubei University of Technology. During postgraduate studies:Silver Award, 13th Hubei Provincial 'Challenge Cup' University Student Entrepreneurship Plan Competition ('Innovative Hubei Youth Power, Youth Contribution') National Third Prize, 23rd National University Student Robotics Competition ROBOCON 2024 'Grain Back to the Granary' Competition (Programming Challenge) National Second Prize, 2023 National University Student Robotics Innovation and Technology Exchange Camp and Competition One SCI indexed paper under review One invention patent Participated in multiple research projects under academic supervisors



Shize Wang is currently pursuing a master's degree at Hubei University of Technology, Wuhan, China. His research interests include mobile robot path planning, multi-robot coordination, and trajectory optimization, with an emphasis on collision avoidance, motion smoothness, and simulation-based evaluation of planning strategies. He received the First Prize in the 14th Hubei Provincial "Challenge Cup" College Student Entrepreneurship Plan Competition.