IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

# CQLite: Communication-Efficient Multi-Robot Exploration Using Coverage-biased Distributed Q-Learning

Ehsan Latif Ramviyas Parasuraman

*Abstract*— **Frontier exploration and reinforcement learning have historically been used to solve the problem of enabling many mobile robots to autonomously and cooperatively explore complex surroundings. These methods need to keep an internal global map for navigation, but they do not consider the high costs of communication and information sharing between robots. This study offers CQLite; a novel distributed Q-learning technique designed to minimize data communication overhead between robots while achieving rapid convergence and thorough coverage in multi-robot exploration. The proposed CQLite method uses ad hoc map merging, and selectively shares updated Q-values at recently identified frontiers to significantly reduce communication costs. The theoretical analysis of CQLite's convergence and efficiency and extensive numerical verification on simulated indoor maps utilizing several robots demonstrate the method's novelty. With over 2x reductions in computation and communication alongside improved mapping performance, CQLite outperformed cutting-edge multi-robot exploration techniques like Rapidly Exploring Random Trees and Deep Reinforcement Learning.**

*Index Terms*— **Multi-Robot, Exploration, Communication**

## I. Introduction

Map-based coverage and exploration is a significant problem of interest in the robotics and multi-robot systems (MRS) community [1]. In this problem, robots continuously explore to obtain the full environmental map in a new bounded environment without prior information. It can be helpful in various applications, including search and rescue, domestic service, survey and operations, field robotics, etc. Autonomous exploration and surveillance solutions can also demonstrate the adaptability of the MRS since robots can carry out these missions in different and uncharted areas.

Recent works have been influential in realizing an efficient exploration objective. For example, information-based methods (e.g., [2]) typically use the Shannon entropy to describe the uncertainty of the environmental map and construct the optimization problems such that the robot's control variable (e.g., velocity) is continuously optimized during the exploration process. On the other hand, frontier-based methods (e.g., [3]) involve deciding the robot's next move (or path) by searching the frontier points on the border of free and unknown points. Often, these methods only produce approximate solutions due to optimization.

Integrating learning with planning solutions is promising, especially for robot exploration [4], [5]. In the reinforcement learning (RL) paradigm, robots can continuously improve competence and adapt to the dynamics of natural surroundings by observing the results of navigational choices made in the actual world [6]. On the other hand, cooperation among robots in an MRS can help achieve a complex mission through simple distributed approaches [7].

School of Computing, University of Georgia, Athens, GA 30602, USA
Author emails: {ehsan.latif;ramviyas}@uga.edu

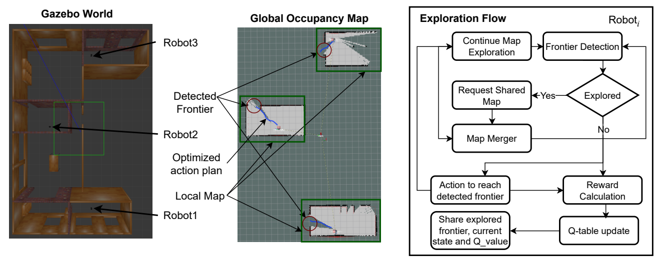


Fig. 1. Overview of the distributed CQLite method for efficient multi-robot exploration, shown with an illustrative simulation in the ROS framework.

This paper explores the intersection between learning and cooperation for efficient map exploration. We leverage the benefits of learning-based paradigms for joint exploration. We aim to create a distributed algorithm that gains knowledge through robot-robot information sharing while minimizing communication and computing overheads. Specifically, we utilize a distributed Q-learning methodology with a coverage-biased reward function and a light communication and information fusion strategy. Fig. 1 provides an overview of the proposed method implemented in the Robot Operating Systems (ROS) framework. The main contributions of this paper are summarized below.

- We propose a novel distributed coverage-biased Q-learning approach (*CQLite*) for efficient multi-robot map exploration, leading to a significant reduction of the communication and computation overheads at each robot. In our approach, we reduce communication complexity by sharing only the current state information, i.e., Q-value, instead of the complete Q-table as done in [8] and apply map merging in an ad-hoc manner.
- We substantiate the potential of our method with theoretical proofs of fast convergence in learning.
- With extensive simulation experiments, we evaluate the performance of our approach against two state-of-the-art (SOTA) multi-robot exploration methods: Rapidly-exploring Random Trees (RRT) for Optimized Exploration [9] and Deep Reinforcement Learning (DRL) for Voronoi-based Exploration [10].
- We open source[1] the CQLite as a ROS package for use and further development by the robotics community.
- We confirm the findings through **real-world robot experiments,** as demonstrated in the attached video.

[1] https://github.com/herolab-uga/cqlite.git

## II. Related Work

Map exploration problems focus on frontier-based and learning-based coverage planning approaches. A robot can be greedily pushed in an occupancy grip-map to the closest boundaries [11] or to the most uncertain (or informative) regions [12]. In frontier-based strategies, the robots will look to expand coverage into the unexplored regions by choosing the next waypoints based on the frontier of the explored map boundaries. For instance, in [9], the multi-robot map exploration objective is integrated into an optimization framework incorporating Rapidly-exploring Random Trees (RRTs) to increase the effectiveness and efficiency of multi-robot map exploration. However, the constraints of such frontier-based approaches are the computing expense of the optimization methods and the possibility of non-optimal outcomes resulting from RRTs' stochastic characteristics.

Researchers also presented communication-efficient solutions for exploration in multi-robot systems. For instance, Zhang et al. [13] introduced the MR-TopoMap based on a topological map, which can independently explore the robot's surroundings while sporadically exchanging topological maps when communication is possible. However, path planning through topological mapping can lead to a sub-optimal path and, specifically, in the case where robots start exploring from the same position, exploring the same map, making it difficult to divide the map into topologies. Masaba and Li [14] proposed an exploration algorithm using the topology of the generalized Voronoi graphs made efficient through a recurrent and lean communication strategy. Corah et al. [15] use information-based distributed planning considering communication restrictions. However, the planner's finite-horizon nature could lead to suboptimal exploration paths because it doesn't consider long-term planning beyond the given horizon, making it more difficult for the system to make decisions based on knowledge in the future. This might prevent robots from efficiently exploring or discovering key regions of interest. More recently, Gao et al. [16] reduced inter-robot communication costs by utilizing a mission-based protocol and centralized planning, where the former can actively disconnect robots to proceed with distributed (independent exploration) and the latter will help them achieve rendezvous to reconnect and share information. However, computing the super-frontier information is computationally expensive, and the active disconnection strategy may limit the robots from sharing other critical data during the mission.

A body of research concentrates on Reinforcement Learning (RL) and Q-learning for multi-robot tasks, modifying the learning mechanism in low communication scenarios for better navigation and exploration [17] and utilizing deep reinforcement learning to achieve optimality in robotic exploration [18]. However, this method calls for frequent map merging, which raises the cost of updates. A Deep RL (DRL) approach for cooperative multi-robot exploration using Voronoi cells was proposed in [10]. Despite its intriguing concept, it was constrained by training difficulties and sub-optimal solution tendencies. Further, DRL has shown promise in some problem spaces, but they frequently offer less-than-ideal solutions outside those contexts. They cannot guarantee convergence in infinite horizons.

To address these gaps in the literature, CQLite considers an efficient information transfer mechanism combined with distributed Q-learning with a coverage-biased reward function for efficient multi-robot cooperation to solve map exploration tasks. CQLite departs from RRT (frontier-based) and DRL (learning-based) regarding exploration strategy by reducing recurrent frontier exploration to avoid mapping overlap and Q-learning update strategy for communication efficiency by only sharing and utilizing recently calculated Q-value to the robots, respectively. Additionally, in both RRT and DRL, robots share locally explored maps on every iteration and apply map merging, which consequently gives rise to computational complexity. We reduce this overhead by only sharing and applying map merging in an ad-hoc manner. By incorporating these novelties, our approach provides significant improvement in computation and communication efficiency, even in cases of limited connectivity scenarios.

It is worth noting that the objectives of cooperative simultaneous localization and mapping (SLAM) techniques [19] and exploration approaches are fundamentally different. The SLAM problem focuses on accurately building and merging the map, while the exploration problem focuses on using the available map to determine waypoints to maximize coverage area. In our work, we use an existing map merging method[2] from the literature to perform multi-robot SLAM. At the same time, our proposed CQLite is designed to maximize exploration and lower communication and computation.

## III. Proposed Approach

We consider $n$ connected robots $r \in V, |V| = n$ deployed at random starting locations in an unknown environment. The connectivity of the robots is expressed by a graph $G = (V, E)$, and a robot $i$ can share data with their immediate neighbors $\mathcal{N}_i = \{j \in V | (i, j) \in E\}$. The robots must find and navigate toward the frontier position on their locally built map as a standard map exploration strategy. To accomplish this efficiently, a robot must decide which frontier to navigate after leaving its current explored region and is expected to reduce the number of steps to take and the size of data exchanges between robots.

Robots only share updated Q-value and the newly explored frontier with other robots in its range (neighbor set). Each robot keeps track of its local and shared frontiers to avoid re-exploration. Robots continue to generate local maps and share the newly developed map only when asked by other robots in case of the already explored frontier. Robots who cannot find new frontiers merge their local map with a map received from peers to build a global map using the feature similarity-based map merging technique [20]. The robot's decisions regarding an action plan are based on the shared information and Q-learning computation. The whole procedure concerning robot $i$ can be visualized in Fig. 2.

[2] https://wiki.ros.org/multirobot_map_merge

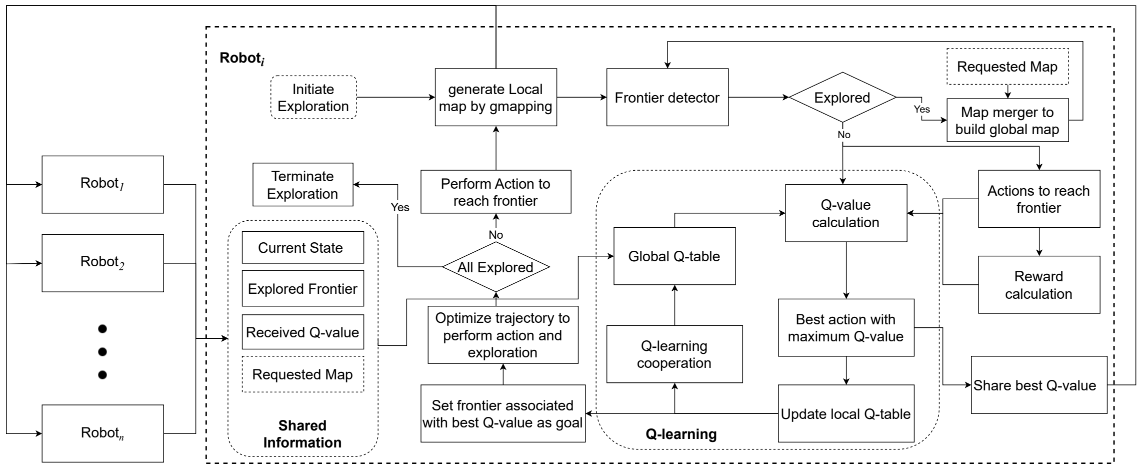


Fig. 2. System architecture of CQLite distributed across several robots. It shows the Robot $i$'s process showing the mapping, frontier detection, and Q-learning operations along with the communication of local map and updated Q-value information to $n$ connected robots.

### *A. Q-learning*

Markov's decision processes frequently model the robot's interaction with the environment. A robot's state is $(x, y, \theta, active/inactive)$ in a global frame. Robots are localized and initialized in a global frame, and positions are known concerning virtually defined bounded regions that can be expanded based on exploration requirements. We consider the frontier's position as states for exploration by applying efficient frontier detection [21]. We consider the current frontier's position as the current state $s_t$ and the newly explored frontier as the next state $s_{t+1}$. A robot can transition from state $s_t \in S$ to state $s_{t+1}$ due to acting $a_t \in A$ based on its state at time $t$. Robot action $a_t$ to reach $s_{t+1}$ from $s_t$ can be determined using discrete-time Hopfield function [22]. The transition probability is defined as $T : S \times A \times S \rightarrow [0, 1]$. The robot will receive a reward for each action using a reward function $R : S \times A \times S \rightarrow R$ specific to the task. The robot will have learned the course of action to take in each state and will be able to maximize the reward of the entire interaction process.

In Q-learning, all possible states and actions are created using the Q table, which then updates each value through iterative learning. The robot then chooses the best course of action for each state based on the values in the table. This approach is frequently utilized in path planning, chess, card games, and other activities.

**Assumptions:** For simplicity, we assume a flat ground terrain environment for exploration.

- The robots have omnidirectional sensors to detect obstacle boundaries within a maximum sensing range $r_s$.
- Each robot has a communication range, $r_c >> r_s$ forming a neighbor set $\mathcal{N}_i$ with constant connectivity to send and receive information about its relative position.
- The connectivity graph $G$ is connected throughout exploration, which is practical to achieve in a multi-robot application. Since the proposed solution is distributed, it ensures maximum coverage even in partial disconnectivity, but at the cost of increased re-exploration.

Here, we introduce CQLite as a distributed method for robot $i$, which is now at state $s_t$ at time $t$ and selects the following state as $s_{t+1}$ to explore independently. CQLite begins with an empty Q-table and updates its table values as exploration proceeds; CQLite aims to achieve fast convergence based on the optimized Q-learning and reward function. Finding the action $a$ that maximizes the Q-value for a specific state $s$ is the goal of the maximum optimization function for Q-learning, i.e., $a^* = \arg\max_a Q_i(s, a)$, where $a^*$ is the optimal action for a given state $s$.

The Q-learning algorithm updates the Q value as

$$\begin{aligned} &Q_{i,t+1}\left(s_t, a_t\right) \\ &= (1-\alpha) Q_{i,t}\left(s_t, a_t\right) + \alpha\left[r_{i,t} + \gamma Q_{i,t}\left(s_{t+1}, a^*\right)\right], \end{aligned} \quad (1)$$

where $r_{i,t}$ is the reward received for taking action $a$ in state $s_t$. The $\alpha \in (0, 1]$ controls the balance between the coverage and delay, and $\gamma$ is the discount factor to prioritize present vs. future rewards. This optimization function is used in the action selection step of the Q-learning, where the agent selects the action that maximizes its expected future reward.

The objective of the CQLite is to perform maximum coverage in less time and avoid overlapping exploration, which can be numerically defined as

$$max_{\pi}\{P_a^{\pi}(t) - \lambda_i E_t(a|\pi)\}, \quad (2)$$

where $P_a^{\pi}(t)$ is the probability to cover the unexplored region using for action $a$ using policy $\pi$ at time $t$, $E_t(a|\pi)$ is estimated time to reach the state $s_t$ by taking action $a$ at time $t$ in policy $\pi$ and $\lambda_i$ is the cost associated with each step taken by robot $i$. We have a vector $path$ extracted by containing position waypoints connecting $s_t$ to $s_{t+1}$ associated with $a$ [23]. For each dimension of $path$ at each control instant $t = t_j$, we first compute the velocity command as:

$$v_{t_j} = K_p \cdot e_{t_j} + K_I \sum_{t=t_0}^{t_j} (e_{t_j}), \quad (3)$$

where $e_{t_j} = s_{t,j} - s_{t,j-1}$ represents the instantaneous error between the intermediate states associated with action $a$ (i.e., the feedback) at time $t = t_j$. Further, $K_p$ and $K_I$ are the so-called proportional and integral gains of the motor controller that regulate the contributions of the corrections induced by the actual error and the error accumulated over time, respectively. These constant values determination is based on the motion constraints of our differential drive robots as discussed by Li et al. [23]. They can be different based on the robot's physical and motion characteristics. In our case, we predetermined values of $K_P$ and $K_I$ as 2 and 0.5, respectively. Now we apply simple kinematic to find the estimate $E_t(a|\pi)$ as:

$$E_t(a|\pi) = \sum_{j=1}^{m} \frac{(e_{t_j})^2}{v_{t_j}}, \quad (4)$$

where $m = |\mathcal{N}_i|$ is the number of intermediate neighbors of the robot $i$. To avoid the exploration of an already explored region for state $s_t$, we determine $P(s_t \cap ES_t)$ as:

$$P(s_t \cap ES_t) = \sum_{j=1}^{m} \frac{P(s_t \cap es_j)}{m}, \quad (5)$$

where $ES_t$ is the set of explored states at time $t$, $es_j \in ES_t$, and $m$ is the number of explored states and overlap probability of each explored state in $ES_t$ can be determined as:

$$P(s_t \cap es_j) = \begin{cases} 1 & dist(s_t, es_j) \leq r_{i,s} \\ 0 & dist(s_t, es_j) > r_{i,s} \end{cases} \quad (6)$$

At each discrete time step $t$, the robot $i$ acquires an observation $s_t$ from the environment, selects a corresponding action $a_t$, then receives feedback from the environment in the form of a reward $r_{it} = R(s_t, a_t)$ as shown below:

$$r_{it} = \begin{cases} -\lambda_i & s_t \in ES_t \\ \lambda_i - Q_{i,t} + \rho(1 - P(s_t \cap ES_t)) + \sigma r_{i,c} & s_t \notin ES_t \end{cases} \quad (7)$$

Where $P(s_t \cap ES_t)$ is the probability of overlap between the current state $s_t$, and the already explored states $ES_t$ by $robot_i$ and other robots, $\rho$ is a scaling factor that controls the importance of minimizing the overlap, $r_c$ is the communication range, and $\sigma$ is the scaling factor that determines the importance of maximizing the communication range. $\sigma$ depends upon the robot's sensing capabilities and makes the reward function modular for heterogeneous robots with different sensing capabilities. Then the state information is updated $s_{t+1}$. The goal of the RL is to select policy $\pi$ that maximizes the discounted sum of future rewards, i.e., $Q_\pi(s_1) = \sum_{\tau}^{t=1} \gamma^t R(s_t, a_t)$, which according to the Bellman optimality principle satisfies.

The reward function in Eq. (7) produces a negative reward whenever the agent has looped back, and the calculated reward is based on the step-cost, Q-value, probability of overlap, and scaling factor otherwise.

**Multi-Robot Lite Cooperation:** We reduce the communication overhead amongst individual exploration-capable robots through a distributed approach, allowing each robot to make independent decisions based on local information and with little interaction from other robots. In our lite version of Q-learning, only the current state and Q-value are communicated amongst nearby robots to encourage cooperation. When another robot receives the information, it will update the received Q value in its Q table and update the local map. We develop a discovery approach based on the distance between simulated robots to replicate the network range in which we only share the current position of a robot $i$, its Q-Value for each direction, and mark the current situation as explored to avoid repetitive exploration.

### *B. Exploration Strategy*

Robots create a global Q table for each cell and action after searching the map and experiencing several experiences. The Q table is then turned into a weighted graph $G = (\mathcal{S}, \mathcal{E}, \mathcal{C})$, where $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ denotes the set of states, and $\mathcal{E} \in |\mathcal{S}| \times |\mathcal{S}|$ signifies the set of edges whose elements indicate whether or not a path exists between the center points of each pair of states. It is assumed that robots do not exchange nodes during exploration, and Voronoi boundaries are fixed. Furthermore, $\mathcal{C}$ is the weight matrix indicating the edge metric cost. The primary goal of discovering this study's reduced graph and significant states is to optimally disperse robots over the coverage region by minimizing the relevant cost function. Because robots move at varying speeds, we formulate the cost as a function of the defined traveling time as $t_{(s_{p_i}, s_q)} = \frac{d_{(s_{p_i}, s_q)}}{v_i}$, where $v_i$ is the $i^{th}$ robot's speed, and $d_{(s_{p_i}, s_q)} \in \mathcal{C}$ is the Euclidean distance between the $i^{th}$ robot's current state $p_i$ and state $q$. Furthermore, knowing the optimal path from state $p_i$ to state $s$, each robot's overall optimal traveling time is the sum of the trip times (costs) from state $p_i$ to state $s$. This study's shortest path between each pair of states is computed using the A* algorithm. Then the total time $\tau$ is calculated by knowing path $\mathcal{P} = \{p_1, p_2, ..., p_n\}$ as

$$\tau_{(S_p, S_q)} = t_{(s_p, s_{p_1})} + t_{(s_{p_1}, s_{p_2})} + ... + t_{(s_{p_n}, s_q)}. \quad (8)$$

After determining the shortest time between each pair of states, the field is partitioned into M Voronoi subgraphs $g_{r_i}$ for $i \in \{1, 2, ..., M\}$ to distribute work proportionally among $M$ robots. To that aim, the ideal Voronoi diagram $g_{r_i}$ for $i^{th}$ robot, according to Lloyd's algorithm, is a split of the area determined as:

$$g_{r_i} = \{s_q \in S | \tau_{(s_{p_i}, s_q)} \leq \tau_{(s_{p_j}, s_q)}, \forall i \neq j\}. \quad (9)$$

Where $j$ is the other connected robot. The $i^{th}$ robot is responsible for covering the state $s$ (associated robot) in its sub-graph $g_{r_i}$ using the Voronoi partitioning result. The entire cost is then calculated as

$$\lambda_{i,(p,g_r)} = \sum_{q \in g_{r_i}} \tau_{(s_{p_j}, s_q)} \phi_q, \quad (10)$$

where $\phi_q$ is the priority value associated with state $s_q$. As the map turns into a graph, higher priority values are assigned to target states, while lower priority values are assigned to states

**Algorithm 1:** Distributed CQLite Exploration

Data: Reward matrix $R$, learning rate $\alpha$, discount factor $\gamma$, step cost $\lambda_i$;
Number of iterations: $t = 0$;
Initialize empty Q-table as $Q_i$ for robot $i$; Initialize empty explored frontier list $ES_t$;
Generate a local map using range sensor;
Initialize Explored Frontier Detected $s_d$ as 0;
**while** $(t \leq t_{max})$ *and* $s_d$*<2* **do**
  $S_t \rightarrow S_{start}$, step = 1;
  Find new frontiers at new $ES_t$ and update $S_t$ using locally explored map;
  $Q_{i,t}$ = list(0);
  **for** *each frontier* $s$ *in* $S_t$ **do**
    **if** $s$ *not in* $ES_t$ **then**
      Calculate Q-value as $q_s$ for actions $a_s$ to reach frontier $s$ using Eq. (1);
      Append $Q_{i,t}(s, a_s)$;
    **else**
      Request for explored Maps;
      Merge maps into local maps;
      $s_d$ = $s_d$ + 1;
  **if** $f_d < 2$ **then**
    Set updated Q-value $q_{update}$ as $max_a(Q_{i,t}(s, a))$;
    Update $Q_i$ with $q_{update}$;
    Take action $a_t$ associated with $Q_i$;
    Share $q_{update}$ with connected robots;
    Receive Q-value for $ES_t$ from connected robots $j \in \mathcal{N}_i$;
    Receive explored frontiers $es_{1:n-1}$ and update $ES_t$ with $s_t$ and $es_{1:n-1}$;
    Set new state associated with $a_t$ as $S_t$;
    Reset $s_d$ as 0;

far and already explored from the current state. The entire travel time (cost) will therefore be minimized, and an optimal solution will be obtained only when the current distance between the robot $i$ and the target state $s_q$, $d_{(s_{p_i}, s_q)}$ converges to zero. Algorithm 1 provides the pseudocode description of CQLite for efficient map exploration implemented in a distributed manner on each robot $i$.

### C. Convergence Analysis

We analyze the convergence of the target Q value update function Eq. (1). We denote the error ratio $\delta_t = \frac{MSE(Q_t)}{E_t(a|\pi)}$, where $MSE(Q_t)$ is the calculated mean square error for Q-table at time $t$ and $E_t(a|\pi)$ is the average number of steps to cover the region by taking action $a$ at time $t$ for policy $\pi$.

**Theorem 1 (Convergence of Q-values):** Using Eq. (1) for Q-value updates, then if $0 \leq \delta_t \leq 1$, with probability $1 - e$, we have the estimated time to reach a given state as:

$$E_t(a|\pi) \leq \omega E_1(a|\pi) + \sqrt{\frac{\ln(1/e) \sum_{i=0}^{t-1} \psi_i^2(\delta_{t-i:t})}{2}} \quad (11)$$

Here, $\psi_i(\delta_{t-1:t}) = \frac{\prod_{j=t-i}^{t-1}(j+\gamma\delta_j)}{\prod_{j=t-i}^{t} j}$, $\alpha_t = \frac{\prod_{j=1}^{t-1}(j+\gamma\delta_j)}{\prod_{j=2}^{t} j}$ and $\gamma = 0.95$.

*Proof:* Our analysis is derived based on the subsequent (synchronous) Q-learning. In contrast to the conventional Q-learning, we swap out the current $Qt$ for the independent Q-function $Q'(s, a)$ for the target $Q_t(s_t, a_t)$ and note that if $Q'_t(s, a) = Q^*_{source}$, we know that $0 \leq \delta_t \leq 1$.
First, we break down the update role into:

$$\begin{aligned} &Q_t(s_t, a_t) \\ &= \left(\frac{t-1}{t}\right) Q_{t-1}(s, a) + \frac{1}{t}\big(r_t + \gamma max_{a'} Q'_{t-1}(s', a') \\ &+ \gamma max_{a'} Q^*_{t-1}(s', a') - \gamma max_{a'} Q^*_{t-1}(s', a')\big) \end{aligned}$$

Let $\varepsilon_t(s, a) = Q_t(s, a) - Q^*(s, a)$ and $\xi(s') = \gamma \times max_{a'}\left(Q^*_{t-1}(s', a')\right)$ then recall the definition of $\delta_t$, we will have $\varepsilon_t(s, a)$

$$\leq \frac{t-1}{t}\varepsilon_{t-1}(s, a) + \frac{1}{t}\big(\xi(s') - E_{s'}\xi(s')\big) + \frac{1}{t}\gamma\delta_t E_{t-1}$$

As we know $\varepsilon_t(s, a) \leq E_t$, by applying maximization and recursion of $E$, we will have:

$$\begin{aligned} E_t &\leq \frac{t-1+\gamma\delta_t}{t} E_{t-1} + \frac{1}{t}\big(\xi(s') - E_{s'}\xi(s')\big) \\ &\leq \frac{\prod_{j=1}^{t-1}(j+\gamma\delta_j)}{\prod_{j=2}^{t} j} E_1 + \sum_{i=1}^{t-1} \frac{\prod_{j=t-i}^{t-1}(j+\gamma\delta_j)}{\prod_{j=t-i}^{t} j} \\ &\times \big(\xi(s') - E_{s'}\xi(s')\big) \\ &= \alpha_t E_1 + \sum_{i=1}^{t-1} \psi_i(\delta)\big(\xi(s') - E_{s'}\xi(s')\big) \end{aligned}$$

According to weighted Hoeffding inequality [24], with probability $1 - e$, we can prove Eq (11) for Theorem 1. ■

This convergence result demonstrates the influence of the error ratio on the convergence rate. In other words, learning will go more quickly for our chosen Q value update function. Even though CQLite shares only updated Q-value, it still achieves the required convergence and provides an optimal strategy for robots to explore the map efficiently.

**Time Complexity:** Assume that the grid factor is $k_g$ (resolution of the grid map on which the grid is divided) and that the target sub-map is $k \times l$ in size. The grid map's size is $k_g k \times k_g l$, and the total number of points is $k_g^2 kl$. The operations to find Q-values must be carried out cyclically $k_g^2 kl$ times. $kl$ can calculate and represent the CQLite's state space size; however CQLite doesn't perform a merging and searching strategy at every iteration; hence the length of the Q-value table is significantly less than $(kl)$ through selecting the training process, and the computing complexity of the algorithm is considerably less than $O(kl)$.

## IV. Experimental Validation

Turtlebot3 robots are used to carry out the exploration plan, implemented using the ROS framework. The open-source *openslam-gmapping*[3] technique of the ROS *gmapping*

[3] https://openslam-org.github.io/gmapping.html

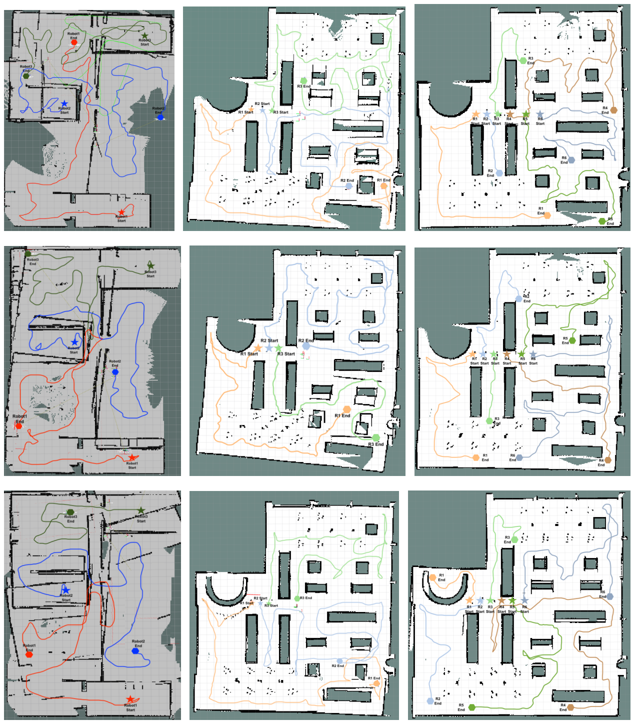

Fig. 3. A depiction of the outcome in a sample trial. It shows the map generated by three robots in the house world (left column) and three and six robots in the bookstore world (canter and right column, respectively) created by the three compared approaches; RRT (top), DRL (center), and CQLite (bottom), with robots moving in a simulated House and Bookstore worlds along with the following trajectories, start and end locations.

package is used to create 2D maps. It uses odometer data and a particle filter method as its foundation. The local maps created by each robot are combined to create the global map. Feature-based map merging[4] is employed to merge maps when required. Frame conversion between the local map frames is necessary for map merger. The coordinate transformation correlation between the robots must be calibrated before combining local maps. In the current work, the global frame is one robot's frame, and the relative positions and orientations of the robots are initialized to a known state. The ROS *movebase*[5] package allows the robot to move toward the goal point while securely avoiding barriers between robots. The Dijkstra algorithm for global path planning and the Dynamic Window Approach (DWA) for local dynamic obstacle avoidance are both implemented in this package. In this study, the units of time and distance are in seconds and meters, respectively.

**Simulation Setup:** A closed simulation environment based on the ROS Gazebo simulator with two indoor template environments is used: the Gazebo's house world ($\approx 250m^2$ area) and the Amazon AWS bookstore world ($\approx 100m^2$ area). The robots may quickly finish the map exploration in a closed environment. Each robot has a laser scanner to gather data about its surroundings. The robots form a fully connected graph through a WiFi communication channel with a standard range of 40-60m. The robot's trajectory is determined based on the fusion of wheel odometry and laser scan information. The following parameters are used in the experiments in the simulated environment. The laser scanner's range and $r_{i,s}$ are set to $15m$ and $1m$, respectively. Additionally, the robot's maximum linear and angular speeds in the simulation are set to $0.5ms^{-1}$ and $\frac{\pi}{4}rads^{-1}$, respectively. The global detector's growth factor $\eta$ and the local detector's growth factor $\eta_1$ in the RRT detector are set to $5m$ and $3m$, respectively. The weight parameters, $\alpha = 0.6$, $\gamma = 0.95$ and $\lambda_i = 2$ for $1m$ distant step. Each experiment was run for ten trials, with average observations reported. We evaluate the performance in the following three scenarios to validate the robustness and scalability of the proposed solution: 1) **3 robots** in the house world, 2) **3 robots** in the bookstore world, and 3) **6 robots** in the bookstore world.

### A. *Evaluation Metrics*

The proposed CQLite and the methods put forward by RRT [9] and DRL [10] are compared in our experiments. We use the below metrics for a comprehensive evaluation: 1) **Mapping Time**: The amount of time spent mapping is a gauge of how effectively a multi-robot map is explored; 2) **Path Length:** This term refers to the length of the robot's trajectory as a whole, which is necessary for a multi-robot system to explore the entire map. The entire trajectory length gives an idea of the robot's energy usage while subtly describing its investigation's effectiveness; 3) **Exploration Percentage:** The percentage of the generated map with time elapsed; 4) **Overlap Percentage:** The percentage of the overlap of the explored map with time elapsed; 5) **Map SSIM:** Structural similarity index measure of generated maps compared with ground truth map to measure map correctness; 6) **CPU Utilization:** The maximum percentage consumption of the processor of a robot throughout the trajectory; 7) **Memory Consumption (RAM):** The maximum occupied memory by the robot throughout the trajectory; 8) **COM payload:** The size of the data communicated by a robot averaged over iterations.

### B. *Results and Discussion*

We have reported each approach's average performance after ten trials in each condition to reduce the measurement noise and analyze the statistical details. A sample of the mapping outcomes of the compared approaches with the trajectories followed by three robots in the simulated environment is shown in Fig. 3 and generated maps also delineate the map correctness. The outcome should be stated considering the average mapping time, distance traveled, and mapping efficiency. Mapping efficiency is determined by comparing with the original map, and reported percentages are normalized with gazebo world dimensions.

Table I provides a comparative analysis of different methods on all the performance metrics and the statistical data from the results. It also lists the theoretical (algorithmic) computational complexity. Figs. 4 shows the comparison of the approaches in the three key performance metrics: computation, communication, and exploration.

[4]http://wiki.ros.org/multirobot_map_merge
[5]https://github.com/ros-planning/navigation

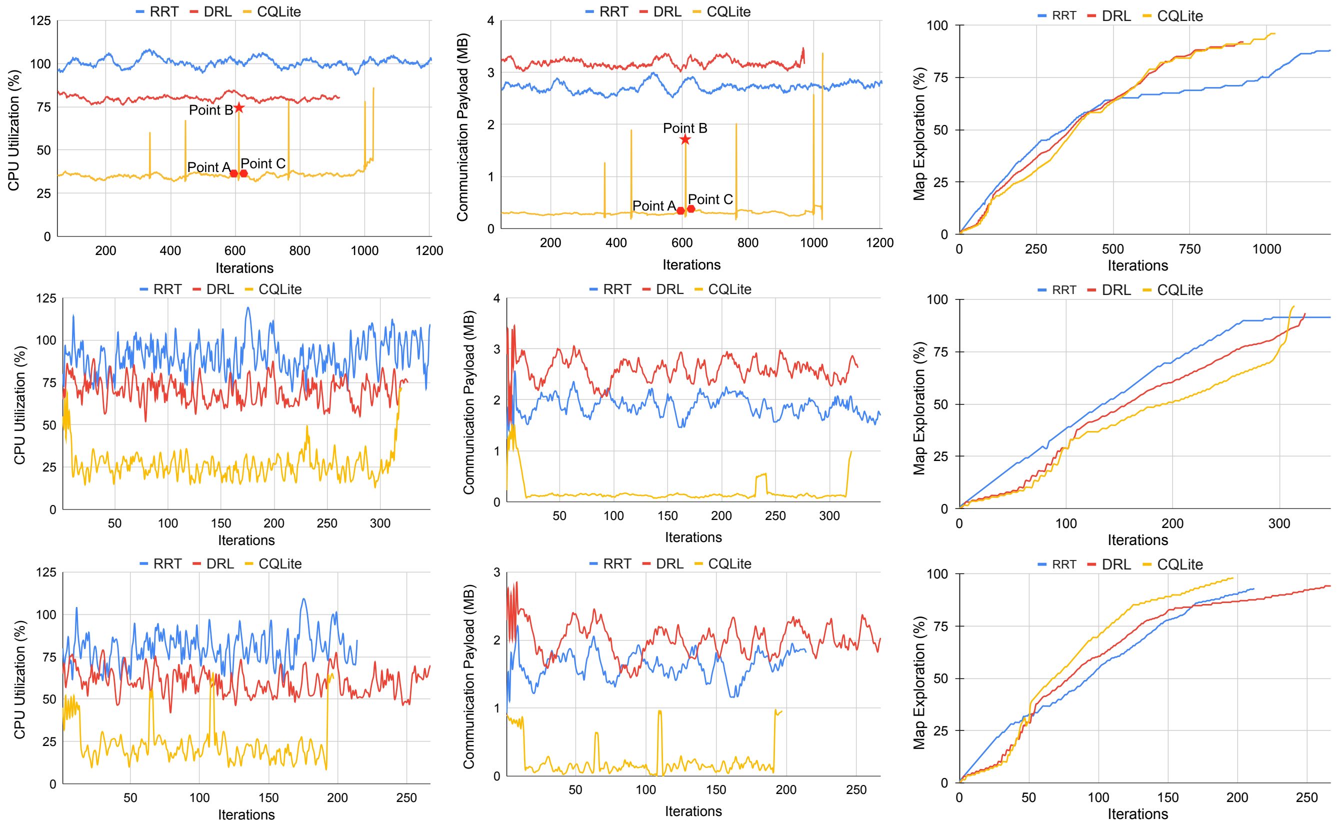


Fig. 4. Computation **(Left)**, Communication **(Center)** cost, and Exploration over time **(Right)** comparison plot of CQLite with RRT and DRL approach in three Gazebo simulated world. Row-wise: **Top** 3 robots in house world, **Middle** 3 robots in bookstore world, and **Bottom** 6 robots in bookstore world.

TABLE I. PERFORMANCE RESULTS OF RRT, DRL, AND THE PROPOSED CQLITE. * INDICATES THE BEST PERFORMER.

| Evaluation parameters | Three robots in house world | | | Three robots in bookstore world | | | Six robots in bookstore world | | |
|---|---|---|---|---|---|---|---|---|---|
| | RRT [9] | DRL [10] | CQLite (ours) | RRT [9] | DRL [10] | CQLite (ours) | RRT [9] | DRL [10] | CQLite (ours) |
| **Mapping Time (s)** | $1208 \pm 52$ | $\mathbf{924}^* \pm 67$ | $1029 \pm 59$ | $347 \pm 32$ | $324 \pm 21$ | $\mathbf{317}^* \pm 19$ | $212 \pm 18$ | $267 \pm 29$ | $\mathbf{197}^* \pm 13$ |
| **Path Length (m)** | $592 \pm 11$ | $604 \pm 19$ | $\mathbf{543}^* \pm 9$ | $278 \pm 26$ | $235 \pm 29$ | $\mathbf{147}^* \pm 21$ | $223 \pm 12$ | $196 \pm 17$ | $\mathbf{121}^* \pm 11$ |
| **Exploration Percentage (%)** | $87 \pm 4$ | $91 \pm 3$ | $\mathbf{95}^* \pm 3$ | $90 \pm 5$ | $93 \pm 2$ | $\mathbf{97}^* \pm 2$ | $93 \pm 4$ | $94 \pm 5$ | $\mathbf{98}^* \pm 2$ |
| **Overlap Percentage (%)** | $51 \pm 5$ | $46 \pm 6$ | $\mathbf{28}^* \pm 2$ | $57 \pm 8$ | $51 \pm 9$ | $\mathbf{31}^* \pm 6$ | $47 \pm 7$ | $39 \pm 8$ | $\mathbf{21}^* \pm 6$ |
| **MAP SSIM** | $0.73 \pm 0.12$ | $0.89 \pm 0.08$ | $\mathbf{0.91}^* \pm 0.06$ | $0.68 \pm 0.21$ | $0.71 \pm 0.13$ | $\mathbf{0.89}^* \pm 0.08$ | $0.71 \pm 0.17$ | $0.73 \pm 0.15$ | $\mathbf{0.93}^* \pm 0.10$ |
| **CPU Utilization (%)** | $112 \pm 22$ | $79 \pm 18$ | $\mathbf{42}^* \pm 8$ | $97 \pm 18$ | $65 \pm 15$ | $\mathbf{34}^* \pm 9$ | $68 \pm 21$ | $47 \pm 16$ | $\mathbf{26}^* \pm 9$ |
| **RAM (MB)** | $824 \pm 19$ | $1264 \pm 41$ | $\mathbf{665}^* \pm 24$ | $624 \pm 16$ | $819 \pm 33$ | $\mathbf{432}^* \pm 21$ | $452 \pm 19$ | $724 \pm 38$ | $\mathbf{319}^* \pm 18$ |
| **COM Payload (MB)** | $2.2 \pm 0.08$ | $2.4 \pm 0.06$ | $\mathbf{0.6}^* \pm 0.02$ | $1.3 \pm 0.06$ | $1.8 \pm 0.04$ | $\mathbf{0.4}^* \pm 0.01$ | $1.1 \pm 0.04$ | $1.3 \pm 0.05$ | $\mathbf{0.2}^* \pm 0.01$ |

The proposed CQLite reliably outperforms other strategies on the key performance metrics. CQLite covers a larger area in less time, improving mapping efficiency by 10% while traveling 22 fewer meters than RRT in the experiment. In three-robot scenarios, CQLite was more effective than DRL and RRT, with 9% and 8% shorter mapping times, respectively. Its path length was also less than DRL's by about 38%. The advantages became even more apparent when the trial involved six robots. While the mapping time was around 26% faster than DRL and 7% faster than RRT for the bookhouse world, the path length was about 38% shorter than with DRL. However, the mapping time of DRL is 10% better than CQLite for house world because DRL is trained for the world and has an optimized path for coverage.

CQLite had an exploration percentage that was 4% greater than DRL in the three-robot scenario. This advantage persisted in the six-robot case, where CQLite's exploration percentage was almost 4% higher than DRL's while maintaining the lowest overlap percentage. The stability and effectiveness of CQLite in multi-robot exploration tasks are highlighted by these results from various experiments.

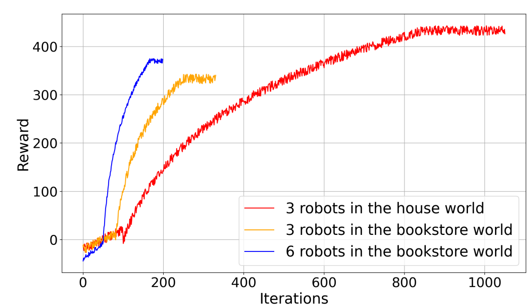


Fig. 5. Reward progression with iterations for three experiments

In the reward convergence (Fig. 5), the 3-robots in the house world show a steady progression, converging at around 850 iterations. The 3-robots in the bookstore demonstrate convergence near 280 iterations, while the 6-robots in the bookstore achieve stability quickest, by approximately 175 iterations. After these points, rewards in each experiment remain consistent, indicating optimal behavior within their respective environments. These convergence results imply the generalizability of the CQLite approach. Further, it can be noted that the local convergence of Q-values (Theorem 1) within the robot's neighborhood set will lead to the group's global convergence when the graph is connected [1], [25].

Communication-wise, CQLite's strategy is more effective. Contrary to RRT and DRL, which exchange locally explored maps continually. CQLite showed a significant reduction of more than 80% in the communication payload (average data size) shared between the robots. Notably, CQLite continues to explore at a constant rate even after reaching 60% coverage, in contrast to RRT, which slows down. This dominance carries over into a real-life three-robot bookshop scenario, where it outperformed DRL and RRT regarding reduced mapping time and shorter journey distances. Results have validated the practicality of CQLite by surpassing DRL and RRT in terms of most of the performance matrices in all scenarios. Further, demonstrating its efficacy and applicability on resource-constrained robots, CQLite maintained decreased RAM, CPU, and communication payload usage. CQLite demonstrates its power in managing a range of multi-robot exploration scenarios by offering improved map quality, as higher MAP SSIM ratings indicate. It is particularly appropriate in situations when there are significant communication and resource constraints.

## V. Conclusion

This paper proposed CQLite, a distributed Q-learning strategy for multi-robot exploration created to get around the excessive communication and computation complexity and expense of learning-based systems. CQLite, which reduces communication and processing overhead by simply sharing Q-value and mapping data over the network, performs well in practical applications. Experimental results revealed that it ensures comprehensive coverage, quick convergence, and cheaper computing costs compared to well-liked RRT and DRL techniques. The same mapping efficiency was attained with only half the CPU load and 80% less communication overhead. In the future, we will examine the reward generality of CQLite to cope with various multi-robot applications.

## References


[1] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[2] B. Fang, J. Ding, and Z. Wang, "Autonomous robotic exploration based on frontier point optimization and multistep path planning," *IEEE Access*, vol. 7, pp. 46 104–46 113, 2019.

[3] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, "Fast frontier-based information-driven autonomous exploration with an mav," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 9570–9576.

[4] R. Shrestha, F.-P. Tian, W. Feng, P. Tan, and R. Vaughan, "Learned map prediction for enhanced mobile robot exploration," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1197–1204.

[5] E. Latif, W. Song, and R. Parasuraman, "Communication-efficient reinforcement learning in swarm robotic networks for maze exploration," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2023, pp. 1–6.

[6] Z. Zhang, X. Wang, Q. Zhang, and T. Hu, "Multi-robot cooperative pursuit via potential field-enhanced reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8808–8814.

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

[8] A. K. Sadhu and A. Konar, "An efficient computing of correlated equilibrium for cooperative $q$-learning-based multi-robot planning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 8, pp. 2779–2794, 2018.

[9] L. Zhang, Z. Lin, J. Wang, and B. He, "Rapidly-exploring random trees multi-robot map exploration under optimization framework," *Robotics and Autonomous Systems*, vol. 131, p. 103565, 2020.

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

[11] W. Gao, M. Booker, A. Adiwahono, M. Yuan, J. Wang, and Y. W. Yun, "An improved frontier-based approach for autonomous exploration," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 292–297.

[12] A. Bouman, M. F. Ginting, N. Alatur, M. Palieri, D. D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick *et al.*, "Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2518–2525.

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

[14] K. Masaba and A. Q. Li, "Gvgexp: Communication-constrained multi-robot exploration system based on generalized voronoi graphs," in *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2021, pp. 146–154.

[15] M. Corah, C. O'Meadhra, K. Goel, and N. Michael, "Communication-efficient planning and mapping for multi-robot exploration in large environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1715–1721, 2019.

[16] Y. Gao, Y. Wang, X. Zhong, T. Yang, M. Wang, Z. Xu, Y. Wang, Y. Lin, C. Xu, and F. Gao, "Meeting-merging-mission: A multi-robot coordinate framework for large-scale communication-limited exploration," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 700–13 707.

[17] A. Serra-Gómez, B. Brito, H. Zhu, J. J. Chung, and J. Alonso-Mora, "With whom to communicate: learning efficient communication for multi-robot collision avoidance," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 770–11 776.

[18] R. Han, S. Chen, and Q. Hao, "Cooperative multi-robot navigation in dynamic environment with deep reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 448–454.

[19] L. Bernreiter, S. Khattak, L. Ott, R. Siegwart, M. Hutter, and C. Cadena, "Collaborative robot mapping using spectral graph analysis," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3662–3668.

[20] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2916–2923.

[21] M. Keidar and G. A. Kaminka, "Efficient frontier detection for robot exploration," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 215–236, 2014.

[22] Z. Uykan, "On the working principle of the hopfield neural networks and its equivalence to the gadia in optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3294–3304, 2019.

[23] J. Li, "Faster parallel algorithm for approximate shortest path," in *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, 2020, pp. 308–321.

[24] P. Duda, M. Jaworski, L. Pietruczuk, and L. Rutkowski, "A novel application of hoeffding's inequality to decision trees construction for data streams," in *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2014, pp. 3324–3330.

[25] R. Konda, H. M. La, and J. Zhang, "Decentralized function approximated q-learning in multi-robot systems for predator avoidance," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6342–6349, 2020.