IJRCS

International Journal of Robotics and Control Systems
Vol. 5, No. 6, 2025, pp. 3350-3375
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

ASCEE

Adaptive Policy Switching for Efficient Multi-Robot Coordination Using Reinforcement Learning

Mohamed Nadour [a,1], Lakhmissi Cherroun [a,2], Imad Eddine Tibermacine [b,3], Abdelaziz Rabehi [c,4], Alfian Ma'arif [d,5],

[a] Applied Automation and Industrial Diagnostics Laboratory (LAADI), Faculty of Sciences and Technology, University of Djelfa, 17000 DZ, Algeria
[b] Department of Computer, Automation and Management Engineering, Sapienza University of Rome, Via Ariosto, 25, 00185 Roma (RM), Rome, Italy
[c] Laboratory of Telecommunications and Smart Systems, Faculty of Sciences and Technologies, University of Djelfa, BP 3117, Djelfa, 17000, Algeria
[e] Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[1] m.nadour@univ-djelfa.dz; [2] l.cherroun@univ-djelfa.dz; [3] tibermacine@diag.uniroma1.it; [4] rab_ehi@hotmail.fr;
[5] alfian.maarif@te.uad.ac.id
* Corresponding Author

ARTICLE INFO

Article history
Received October 01, 2025
Revised November 18, 2025
Accepted December 30, 2025

Keywords
Multi-Robot Coordination;
Task Allocation;
Reinforcement Learning;
Adaptive Policy Switching;
Context-Aware Control

ABSTRACT

Multi-robot systems operating in diverse environments require coordination strategies that balance efficiency and safety. This paper presents an adaptive framework combining heuristic planning and learning-based control to achieve that balance. The proposed system dynamically switches between a classical heuristic controller and a Q-learning-based policy according to real-time obstacle density, enabling context-aware adaptation to varying environmental complexity. The framework was evaluated in three representative scenarios of increasing difficulty, including a single robot with one task in an obstacle-free environment, a moderate case with three robots and five tasks among eight obstacles, and a complex case with five robots managing eight tasks amid fifteen obstacles. Performance was analyzed using several metrics such as task completion time, near-miss frequency, operational efficiency, and energy consumption. Results show that while the baseline policy performs best in sparse environments, the reinforcement-learning policy achieves faster completion in dense ones, though this comes at the cost of an increased frequency of near-misses due to its efficiency-driven behavior. The adaptive method effectively reconciles this trade-off, reducing near-misses by 25–40 % while maintaining competitive completion times and minimal energy usage. These findings demonstrate that adaptive policy selection provides robust, context-sensitive coordination across heterogeneous environments and can support missions in logistics, exploration, and disaster-response robotics, autonomously optimizing safety and performance according to real-time conditions.

1. Introduction

Multi-robot systems (MRS) have become a fundamental component of modern automation, enabling teams of autonomous robots to perform tasks that are difficult, inefficient, or impossible for a single robot to accomplish [1]-[3]. These systems have been successfully applied in domains such as warehouse logistics, search and rescue, precision agriculture, and environmental monitoring. Their effectiveness depends on how well multiple robots coordinate task allocation, motion planning, and collision avoidance in diverse and dynamic environments. Achieving both efficiency and safety in such coordination remains a central challenge in robotics research. Classical methods for multi-robot coordination, such as graph-based algorithms, the Hungarian method, market-driven auctions, and A*-based path planning [4]-[9], provide predictable and verifiable performance. They guarantee convergence and are computationally efficient under static or low-density conditions. However, these deterministic systems typically rely on precomputed or centralized information and lack the flexibility to adapt in real time to changing environments, uncertain dynamics, or unforeseen obstacles. As a result, their performance often degrades when deployed in complex, high-density, or partially known environments.

Reinforcement learning (RL) and other learning-based approaches have emerged as promising alternatives that enable robots to learn adaptive behaviors from experience [10]-[14]. These methods allow policies to evolve through interaction with the environment, improving responsiveness and efficiency in dynamic scenarios. Yet, purely learning-based systems can exhibit instability, inefficiency, or unsafe maneuvers when operating under exploration pressure or environmental uncertainty [15]-[18]. In dense or cluttered settings, learned controllers sometimes favor time- or path-efficiency at the expense of safety, resulting in increased near-miss incidents or collisions. This trade-off defines what we refer to as the environment-dependent performance paradox [19]-[21], a single control policy cannot perform optimally across all environmental conditions. Classical controllers achieve high safety but poor efficiency in complex settings, while learning-based controllers achieve high efficiency but reduced safety in dense or dynamic environments. The central challenge is therefore not finding one "best" controller, but determining when and how to use each method to maximize system performance.

To address this challenge, we propose an adaptive policy-switching framework that dynamically alternates between a deterministic baseline controller and a Q-learning-based reinforcement learning policy according to environmental complexity [22]-[24]. The switching mechanism relies on a lightweight obstacle-density heuristic that allows the system to detect environmental variations and adapt in real time. This hybrid design preserves the reliability of classical control in simple conditions and the adaptability of learning-based strategies in complex ones, achieving a balanced trade-off between safety and efficiency. A critical analysis of the literature reveals a common limitation in existing hybrid approaches: they typically fuse classical and learning elements into a single, static policy. These designs lack a high-level mechanism for real-time arbitration between fundamentally different coordination strategies. This is the key gap our work addresses. We propose that robust multi-robot coordination requires not a single optimal policy, but an optimal policy-selection strategy [25]-[28]. To this end, our novel context-aware meta-policy dynamically switches between a verifiable classical controller and a learning-based policy based on real-time environmental assessment. This architecture allows the system to actively navigate the efficiency-safety trade-off, leveraging the strengths of each paradigm where they are most effective. The contributions of this work are threefold and represent significant advancements in the field of multi-robot coordination:

- Comprehensive empirical validation of the environment-dependent performance characteristics of classical versus learning-based approaches in MRS coordination, offering new insights into their respective strengths and limitations.
- A novel, lightweight, untrained policy-selection mechanism that operates in real time, using obstacle density as context for switching between controllers, making it suitable for resource-constrained deployments.

- Demonstrated evidence that context-aware adaptation enables superior robustness, achieving a more favorable balance of efficiency, safety, and energy consumption than any single-policy approach.

The research contribution is an adaptive meta-policy framework that enables real-time switching between classical and learning-based coordination strategies based on obstacle density, achieving a balanced trade-off between efficiency, safety, and energy use. Conceptual architecture of the adaptive multi-robot framework shown in Fig. 1.

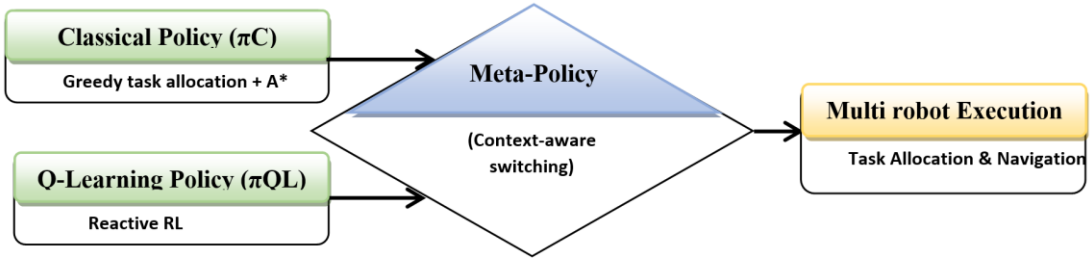


Fig. 1. Conceptual architecture of the adaptive multi-robot framework

This paper is structured to provide a comprehensive examination of these contributions while positioning our work within the broader research landscape. Section 2 reviews relevant literature in multi-robot coordination and reinforcement learning, examining both classical approaches and recent learning-based methods, and identifying key research gaps that our work addresses. Section 3 details our methodological framework, including environment design, policy implementations, adaptation mechanisms, and evaluation metrics, providing sufficient detail for reproducibility and further research. Section 4 presents a comprehensive analysis of performance across environmental complexities, including statistical validation and comparative evaluation of policy approaches. Section 5 concludes by synthesizing the framework's ability to balance efficiency and safety through adaptive policy selection, discusses implications for robust multi-robot system deploymentand proposes future research extending this adaptive paradigm.

2. Literature Review

Research on multi-robot systems (MRS) has developed considerably over the past two decades, driven by advances in sensing, computation, and distributed intelligence. The goal of these systems is to enable teams of autonomous robots to accomplish shared tasks cooperatively and efficiently, even in dynamic and uncertain environments. Achieving this objective requires coordination strategies that balance safety, efficiency, scalability, and adaptability. The existing literature on MRS coordination can be broadly categorized into classical optimization-based approaches, learning-based approaches, and hybrid or context-aware frameworks that attempt to combine the strengths of both.

Classical multi-robot task allocation (MRTA) methods provide the conceptual foundation for cooperative decision-making. These approaches rely on explicit optimization models to assign robots to tasks or paths based on predefined cost functions. Techniques such as the Hungarian algorithm, the Contract Net Protocol, and auction-based allocation [29]-[35] are among the most influential, offering computational efficiency and deterministic convergence. They perform well in structured and fully observable environments, where global knowledge and centralized coordination can be assumed. Similarly, path-planning methods including A*, D* Lite, and rapidly exploring random trees have been used extensively to generate collision-free trajectories and minimize travel cost [36]-[39]. At the local level, potential-field and velocity-obstacle formulations have been employed to manage real-time avoidance of other robots or obstacles. While these methods are computationally efficient and mathematically tractable, their performance deteriorates in the presence of environmental uncertainty, communication delays, or dynamic obstacles. The deterministic nature of classical algorithms constrains their ability to adapt to real-time changes, making them less suitable for unstructured or large-scale domains.

Learning-based coordination strategies have emerged to overcome these limitations by endowing robots with the ability to learn adaptive behaviors through experience. Reinforcement learning (RL) and its deep learning variants (DRL) have enabled robots to optimize policies for navigation, task allocation, and cooperative behavior through reward-driven interaction with the environment [40]-[45]. In RL, agents learn to maximize cumulative rewards by selecting actions that lead to long-term performance gains, which removes the need for an explicit analytical model of the system. The introduction of deep Q-Networks, actor–critic architectures, and policy-gradient algorithms has significantly improved scalability and generalization to complex, continuous, and high-dimensional environments [46]-[47].

These methods have shown strong results in multi-robot navigation, cooperative transport, formation control, and dynamic task assignment. However, RL-based systems face several challenges when deployed in real-world conditions. They require large quantities of training data, long convergence times, and often extensive computational resources. Moreover, due to the exploratory nature of RL, agents can exhibit unsafe or unpredictable actions when exposed to novel or unseen environments. In dense or cluttered scenarios, a learned policy may favor path efficiency or speed at the expense of safety, leading to increased near-miss events or collisions. Limited onboard computation and communication bandwidth further restrict the feasibility of deploying fully learning-based systems in real time.

Building on these advances, recent studies have integrated optimization techniques, deep reinforcement learning, and redistribution strategies to enhance MRTA scalability, coordination efficiency, and adaptability across diverse environments [48]-[51]. These frameworks attempt to combine the structured stability of classical optimization with the adaptability of learning-based approaches, enabling more flexible coordination under environmental uncertainty. Examples include optimization-guided learning architectures, capsule-network-based task allocation, soft actor–critic navigation, and robot redistribution models for large-scale coordination. These studies demonstrate notable improvements in multi-robot scalability and robustness under complex conditions. However, their reliance on centralized architectures, offline training, or fixed blending rules limits adaptability when encountering new or evolving environmental contexts. The inability of these frameworks to autonomously switch control strategies in real time constrains their generalization and responsiveness.

A critical analysis of existing hybrid and context-aware systems reveals that most approaches fuse classical and learning-based controllers into a single static framework. While this fusion allows partial integration of deterministic safety and learning-based adaptability, it also imposes rigidity, preventing the system from dynamically prioritizing the most suitable coordination strategy. Once integrated, such systems operate under fixed weighting or blending parameters, lacking an arbitration mechanism that can assess environmental complexity and adjust policy dominance accordingly. As a result, hybrid systems remain context-dependent: in sparse environments, learning-based components may overact, wasting computation and energy, while in dense or uncertain environments, deterministic rules may over-constrain movement, reducing responsiveness and efficiency. This inability to adaptively navigate the safety–efficiency trade-off represents one of the key shortcomings of prior hybrid frameworks.

The present study addresses this gap by introducing an adaptive policy-switching framework that incorporates a lightweight meta-policy for real-time controller selection. Unlike conventional hybrid designs that permanently blend controllers, the proposed system evaluates environmental conditions using an obstacle-density measure and dynamically selects the most appropriate policy at runtime. When the environment is simple and sparsely populated, the framework favors the deterministic baseline controller, maintaining stability and safety. When the environment becomes dense or complex, the meta-policy activates the reinforcement-learning controller, allowing the robots to exploit learned adaptability and efficiency. This dynamic switching mechanism ensures that the coordination strategy is always matched to environmental demands. It combines the proven safety of classical methods with the learning flexibility of RL, enabling robust, scalable, and context-aware performance across heterogeneous multi-robot environments.

3. The Proposed Methodology

This section details the modeling, control, and learning components of the proposed adaptive multi-robot framework. To improve clarity, complete mathematical derivations such as the full kinematic model and formal task-assignment equations are presented in Appendix A, while the main text summarizes conceptual flow and algorithmic design.

This section describes the proposed adaptive multi-robot coordination framework. It integrates classical task allocation, A* path planning, and reinforcement-learning-based control under a unified meta-policy. The structure follows a logical sequence from system modeling, task formulation, and policy definition to adaptive decision-making and evaluation, clarifying how each component contributes to overall system performance. The methodology follows a structured workflow that integrates modeling, policy development, adaptive switching, and evaluation. Each stage contributes to testing the core hypothesis that context-aware adaptation improves coordination robustness compared to single-policy methods. The detailed workflow is summarized in Fig. 2, Fig. 3 provides an overview of the system workflow, from robot modeling and environment formalization through policy execution and performance evaluation.

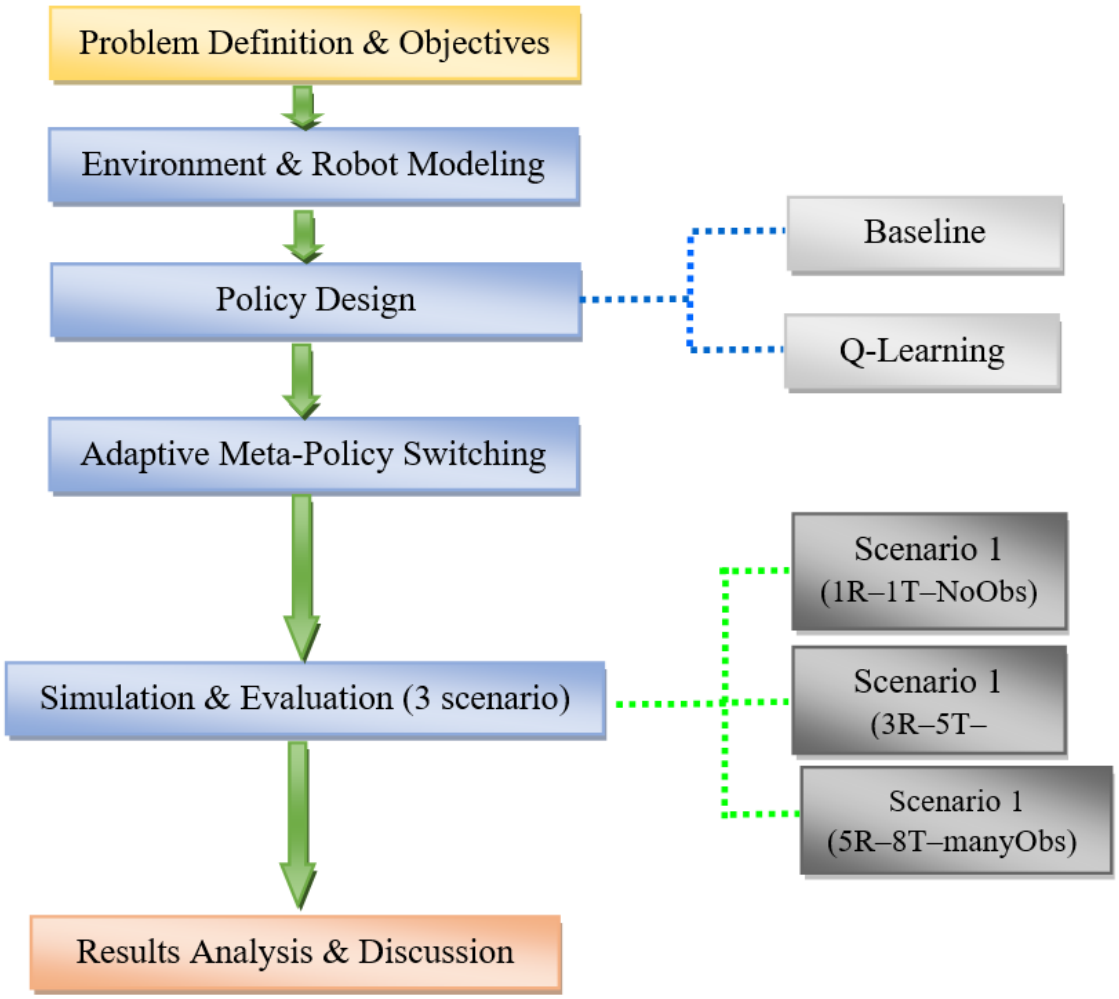


Fig. 2. Adaptive multi-robot coordination methodology flowchart

3.1. Multi-Robot System

The system in this work comprises a homogeneous fleet of N differential-drive mobile robots within a shared workspace. The robot collective is formally defined as:

$$\mathcal{R} = \{r_1, \ldots, r_N\} \quad (1)$$

3.1.1. Kinematics

Each robot r_i possesses a continuous state (at time t) representation capturing both positional and orientational information, as presented in Fig. 3:

$$p_i(t) = [x_i(t), y_i(t), \theta_i(t)]^\top \in \mathbb{R}^2 \times \mathbb{S}^1 \quad (2)$$

Each robot follows unicycle kinematics; complete continuous and discrete derivations are given in Appendix A §A.1. Kinematic model of a unicycle mobile robot shown in Fig. 4.

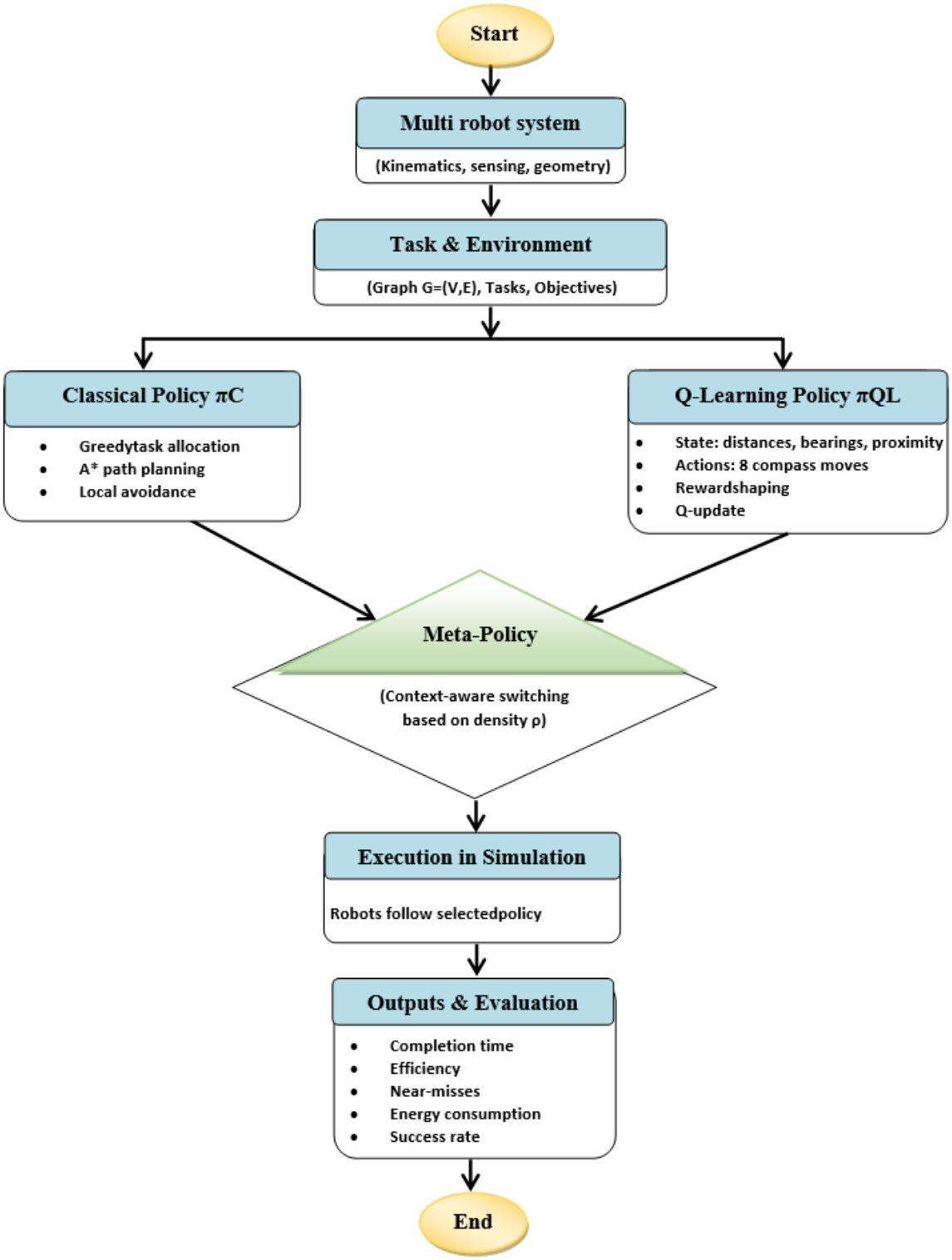


Fig. 3. Workflow of the proposed adaptive multi-robot framework

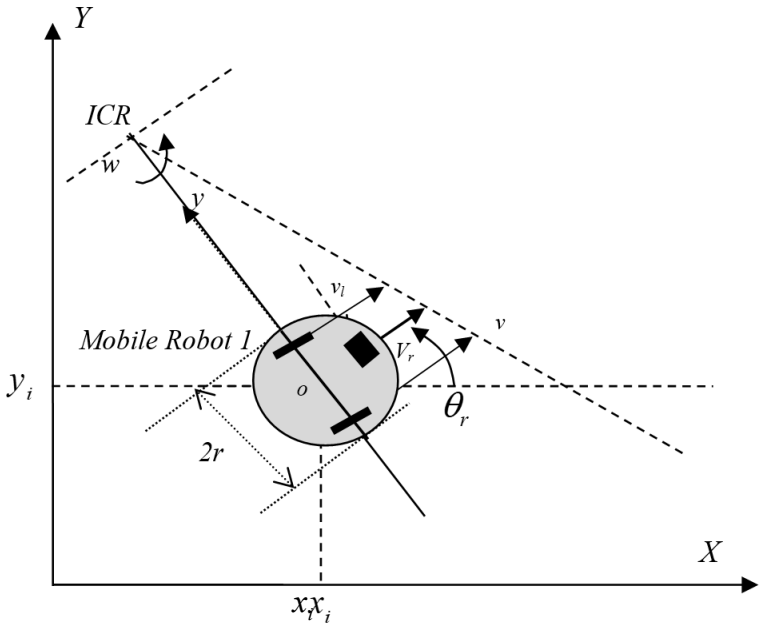


Fig. 4. Kinematic model of a unicycle mobile robot

3.1.2. Geometry and Safety

The simulated robots are modeled as discs of radius r. Obstacles are inflated by r so that point-robot planning on the inflated map guarantees clearance (for provable collision avoidance). A near-miss occurs whenever the inter-robot distance falls below a safety margin $d_{\min} > 2r$; a collision is $< 2r$.(as presented on Fig. 5).

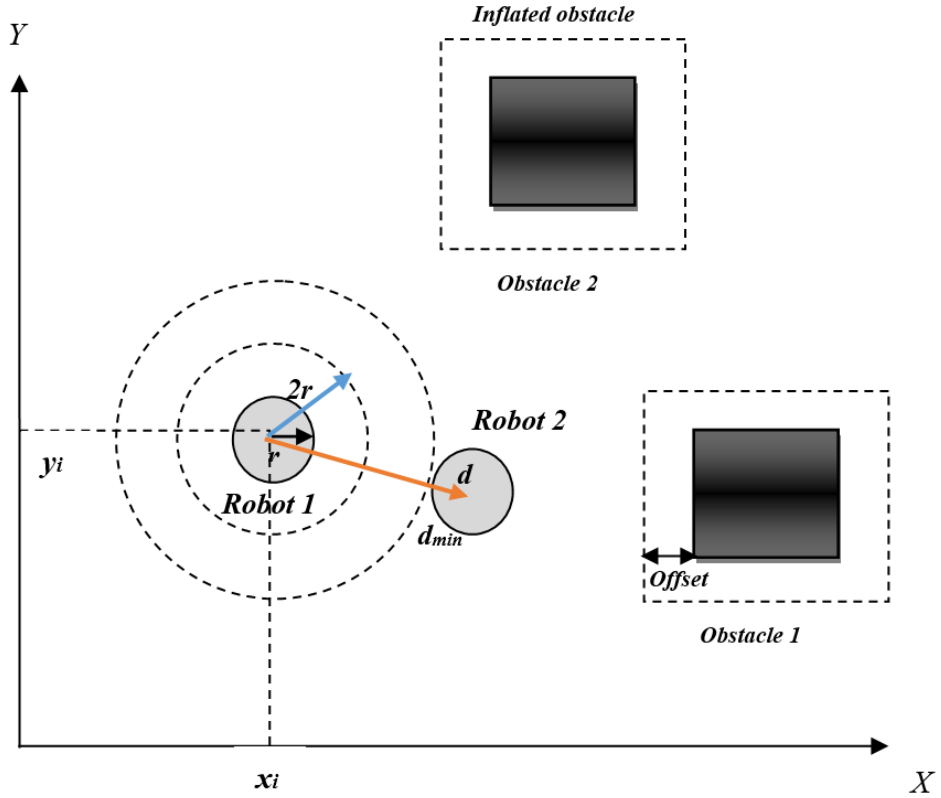


Fig. 5. Multi-robot interaction and collision avoidance with static obstacles

This geometric formulation ensures provable clearance within the workspace; full motion-update derivations appear in Appendix A §A.1.

3.1.3. Sensing and Localization

Each robot in the simulation is equipped with:

- Proximity sensing (e.g., ultrasonic or range) providing distances to the nearest obstacles in a limited field of view; we denote a minimal triad.

$$D_i(t) = \{d_i^{front}, d_i^{left}, d_i^{right}\} \tag{3}$$

Which is used to compute a local obstacle density surrogate $\rho_i(t)$;

- Odometry (wheel encoders) providing a pose estimate on a known map.

We define two context measures: a local density $\rho_i(t)$ computed from proximity (e.g., fraction of saturated rays or a decreasing function of d_i), and a global density estimated from the inflated map $\mathcal{O}$:

$$\rho_{env} = \frac{|\mathcal{O}|}{N + M} \tag{4}$$

where $|\mathcal{O}|$ is the number of inflated obstacles, N is the number of robots, and M is the number of tasks. These signals trigger the meta-policy that switches between the classical and Q-learning controllers.

3.2. Task and Environment Formalization

This subsection provides a formal mathematical description of the operational environment, the task structure, mission objectives of the multi-robot problem. The environment is abstracted as a graph to facilitate path planning and collision checking, while tasks are defined as atomic goals that must be achieved by the robot team. The overarching mission objective is framed as a multi-criteria optimization problem, balancing efficiency, safety, and energy consumption. This graph-based representation provides a computationally efficient foundation for applying search-based path planning algorithms like A*.

3.2.1. Environment Graph

The workspace is a connected grid/graph $G = (V, E)$ with 8-neighborhood connectivity. Vertices V represent free cells or waypoints, edges E represent traversable adjacencies. Let $\mathcal{O} \subset V$ be the (inflated) obstacle set; free cells are:

$$V_{free} = V \backslash \mathcal{O} \tag{5}$$

The grid resolution is carefully selected to balance computational efficiency with motion smoothness requirements, typically set to match the robot diameter to ensure path existence guarantees.

3.2.2. Tasks

The mission objective involves coordinated completion of M distinct tasks distributed throughout the environment:

$$\mathcal{T} = \{\tau_1, \ldots, \tau_M\}, \tau_j \leftrightarrow g_j \in V_{free} \tag{6}$$

where g_j is the goal cell. Task τ_j is completed when some robot i satisfies.

$$\| p_i(t) - g_j \| \leq d_{goal} \tag{7}$$

where d_{goal} represents the goal proximity threshold, typically set to accommodate positioning uncertainties and physical dimensions. Robots may execute multiple tasks sequentially.

The full mathematical definition of the dynamic assignment and its constraints is presented in Appendix A §A.2. The discussion below focuses on how these task metrics interact with the

adaptive meta-policy.

3.3. Policy Architecture

The system implements two modular control stacks that share the same task and map interface.

3.3.1. Classical Baseline Policy π_C

The classical policy embodies a deterministic approach to multi-robot coordination, prioritizing verifiable safety and computational efficiency:

- Task allocation (greedy + uniqueness), Nearest-task assignment using Euclidean distance minimization (When robot i becomes idle, we assign the nearest unfinished task):

$$A_t^{greedy}(i) \in arg\min_{j \in \mathcal{T}_{open}} \| p_i(t) - g_j \| \tag{8}$$

To avoid duplicate assignments, ties are resolved with a min-cost bipartite matching (Hungarian) over the robot–task distance matrix for the current idle set (with a greedy unique fallback if the solver is unavailable).

- Global path planning: For a pair $(i \to j)$, we compute a shortest 8-connected path $\Pi_{ij} \subset V_{\text{free}}$ using A* with an octile/diagonal heuristic. The robot follows Π_{ij} by executing one motion primitive per cycle.
- Local obstacle/peer avoidance: If the next step would reduce the distance to any neighbor below $d_{\min}$, the robot selects the next best waypoint on Π_{ij} that increases separation; failing that, executes a deterministic wall-following sidestep driven by $D_i(t)$ to increase the minimum inter-robot distance while staying in free space.

This policy is computationally light and reliably safe in sparse scenes, but it can become conservative or myopic in dense, dynamic situations.

3.3.2. Q-Learning Policy π_{QL}

The Q-learning policy employs reinforcement learning to develop context-sensitive navigation behaviors through experiential learning:

- **State space**

For robot i navigating toward g, we use a compact feature vector capturing local context and goal geometry:

$$s_i(t) = \left[d_i^{front}, d_i^{left}, d_i^{right}, \phi_i(t), \psi_i(t), v_i(t), \omega_i(t)\right] \tag{9}$$

where ϕ_i is the bearing to the goal and ψ_i is the bearing to the nearest obstacle (both wrapped to $[-\pi, \pi)$). In our discrete implementation, these are quantized into bins (goal distance and bearing; binary obstacle flags in cardinal directions) to enable tabular learning with a small state space.

- **Action space**

A discrete set of motion primitives,

$$\mathcal{A} = \{(v_{high}, 0),\ (v_{low}, 0),\ (0, \omega_L),\ (0, \omega_R),\ (v_{low}, \omega_L),\ (v_{low}, \omega_R)\}, \tag{10}$$

Allowing forward motion, in-place turns, and gentle arcs. In the grid-world simulator we instantiate $\mathcal{A}$ as 8 **compass moves** (constant step, fixed headings), which makes the RL controller comparable to A*.

- **Reward Function Engineering**

The reward function carefully balances multiple objectives (at each step, the reward balances

progress, time, and safety):

$$R = \beta_p \Delta d_i - c_t - c_{\{miss\}}\mathbb{I}_{near} - c_{\{occ\}}\mathbb{I}_{obstacle} + R_{\{goal\}}\mathbb{I}_{goal} \quad (11)$$

where $\Delta d_i(t) = \| p_i(t-\Delta t) - g \| - \| p_i(t) - g \|$ is positive when approaching the goal. Coefficients $(\beta_p, c_t, c_{\text{miss}}, c_{\text{occ}}, R_{\text{goal}})$ are tuned to favor fast yet safe progress; we use a large terminal bonus and explicit penalties for near-misses and rejected (obstacle-violating) moves.

The numerical weight values and learning-rate parameters used for training are summarized in Appendix A §A.3. The weighting factors were tuned experimentally to balance safety and efficiency. In particular, the near-miss penalty $c_{\text{miss}} = -0.3$ was chosen after testing values between -0.1 and -1.0. Smaller magnitudes encouraged riskier behavior and increased proximity events, while larger penalties led to overly conservative motion and longer completion times. The selected value provided the best compromise, reducing near-misses by approximately 25 to 40 percent while maintaining efficient task completion.

- **Ablation protocol**

We assess safety sensitivity by sweeping the near-miss penalty c_{miss} in $\{-0.1, -0.3, -0.6, -1.0\}$. For each value, we run 20 seeds in Scenario S3 and report mean ± 95% bootstrap confidence intervals for near-misses, completion time, efficiency, and energy. This isolates the effect of the safety weight in the reward while keeping all other parameters fixed.

- **Learning Algorithm Implementation**

We maintain a tabular Q-function $Q: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and update it via

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(R + \gamma \max_{a'} Q(s',a') - Q(s,a)\right) \quad (12)$$

with learning rate $\alpha \in (0,1]$, discount $\gamma \in [0,1)$, and ε-greedy exploration (decayed over episodes). At execution, the greedy action is filtered by a safety check that rejects steps into inflated obstacles (and, optionally, steps that would immediately violate $d_{\min}$).

The Q-learning component was trained offline for each scenario configuration until convergence, which was typically reached within 25–35 episodes. Each episode comprised a complete simulation of 300 time steps across all robots and tasks. The learning curves in Fig. 12 illustrate the efficiency metric stabilizing within this range. Training was performed separately for each scenario type (1R–1T–NoObs, 3R–5T–FewObs, and 5R–8T–ManyObs) to reflect different spatial complexities. Once trained, the learned Q-values were reused across random seeds for that scenario. Importantly, the meta-policy that selects between baseline and RL modes is entirely untrained; it functions as an online heuristic based on measured obstacle density and requires no further learning or fine-tuning.

Q-learning was selected over deep reinforcement learning approaches such as DQN or PPO because it offers simplicity, interpretability, and suitability for real-time on-device execution. The discretized state representation, which includes goal distance, bearing, and nearby obstacle information, yields a small and manageable action–state space. This can be explored efficiently with a lightweight Q-table that converges within 25 to 35 episodes without GPU acceleration or complex hyperparameter tuning. The explicit Q-values provide clear insight into the learned policy, which supports safer validation and embedded deployment. The adaptive meta-policy proposed in this study is algorithm-agnostic, meaning that the same switching mechanism can work with DQN, PPO, or other learning models in future research.

3.4. Adaptive Meta-Policy Framework

3.4.1. Context-Aware Policy Selection Mechanism

The meta-policy implements intelligent arbitration between classical and learning-based approaches through systematic environmental assessment:

- **Density-Based Switching Criterion**

The fundamental decision rule employs a threshold mechanism:

$$\pi_{active} = \begin{cases} \pi_C & if\ \rho_{env}(t) < \rho_{thresh} \\ \pi_{QL} & if\ \rho_{env}(t) \geq \rho_{thresh} \end{cases} \quad (13)$$

where $\rho_{\text{thresh}} = 0.7$ represents an empirically determined threshold optimizing the efficiency-safety trade-off across diverse scenarios.

The value $\rho_{\text{thresh}} = 0.7$ was obtained through empirical testing. Thresholds between 0.5 and 0.9 were compared. Lower thresholds switched too early to the reinforcement-learning controller in moderately sparse maps, while higher thresholds delayed adaptation in dense environments. The value 0.7 produced the most consistent balance between safety and completion time.

- **Decision Logic Refinements**

The basic threshold mechanism is enhanced with several sophistications:

- Hysteresis bands preventing rapid policy oscillation near threshold boundaries
- Temporal filtering ensuring policy consistency over meaningful time horizons
- Confidence metrics incorporating estimation uncertainties in density calculations
- Fallback protocols for sensor degradation scenarios

3.5. Generalization and Real-World Applicability

Although the validation was performed in simulation, the proposed framework is designed for direct use on physical robots. The modular structure that separates the baseline and learning controllers allows straightforward implementation on differential-drive or omnidirectional platforms. Obstacle density can be estimated from LiDAR or camera-based occupancy maps, enabling onboard switching without centralized control. Because the meta-policy is untrained and computationally lightweight, it can run on heterogeneous teams with limited resources and supports scalable real-world deployment. The next section presents the simulation results obtained under environments of increasing complexity.

4. Results and Discussions

This section presents a comprehensive evaluation of the proposed adaptive multi-robot task allocation and navigation framework across three distinct environmental scenarios. We compare the performance of three policy approaches, classical baseline, Q-learning-based machine learning, and our novel adaptive meta-policy, using several performance metrics including completion time, safety (near-misses), operational efficiency, and energy consumption. The analysis focuses on understanding the fundamental trade-offs between efficiency and safety in multi-robot coordination, and validates the effectiveness of our context-aware adaptation mechanism in achieving robust performance across diverse operational conditions.

4.1. Environment and System Modeling

The experiments were implemented in a MATLAB simulation framework designed for multi-robot research, with the workspace defined as a 100 × 70 grid world using 8-connected neighborhood adjacency. This discrete representation supported both deterministic path planners such as A* and tabular reinforcement learning controllers, which allowed direct comparison under a unified model.

Robots were modeled as differential-drive unicycle agents, similar to platforms like TurtleBot or Pioneer, and each was approximated as a disc of radius $r = 2$ to simplify collision detection. Their motion followed discretized unicycle kinematics with a step size of $v_{step} = 0$, and control inputs were bounded by $|v| \leq v_{\max}$, and $|\omega| \leq \omega_{\max}$.

For perception, robots were equipped with local proximity sensors in three directions (front, left, and right) that provided distance-to-obstacle values, while odometry ensured accurate localization within the map so that evaluation focused on coordination rather than localization errors. A local obstacle density $\rho_i(t)$ was estimated from proximity readings, while the global density ρ_{env} (equation (5)) was derived from the map. Obstacles were generated as rectangular blocks with random dimensions within fixed ranges (width: 6–14, height: 5–10 cells), and to guarantee clearance each obstacle was inflated by r. Robots and tasks were then placed in randomly selected free cells, which ensured feasibility while maximizing variability across trials.

To quantify variability across independent trials, all scenarios were repeated using five random seeds. Mean values and 95 % confidence intervals were computed for completion time, efficiency, and near-miss frequency. Statistical tests (two-sample t-tests) confirmed that improvements achieved by the adaptive framework were significant ($p < 0.05$) when compared with both baseline and ML-only configurations.

4.2. Performance Across Scenarios

This section interprets the experimental findings in four aspects: (i) the main outcomes of the present study, (ii) their relation to previous research, (iii) implications for multi-robot coordination design, and (iv) the strengths and limitations of the proposed framework.

To evaluate the framework under different levels of complexity, three scenarios were considered. These scenarios, summarized in Table 1, capture environments of increasing difficulty, ranging from sparse to dense configurations. The first scenario (**S1**) involves a single robot and a single task in an obstacle-free environment, serving as a baseline for performance in structured and simple settings. The second scenario (**S2**) introduces moderate complexity with three robots, five tasks, and eight obstacles, creating opportunities for interaction and congestion. The third scenario (**S3**) represents a dense and cluttered environment with five robots, eight tasks, and fifteen obstacles, modeling the most challenging coordination conditions. These settings provide comprehensive tests for comparing the classical policy, the Q-learning policy, and the proposed adaptive meta-policy. The results demonstrate that the adaptive policy consistently achieves a superior balance between efficiency and safety across varying complexities.

Table 1. Scenario configurations for performance evaluation

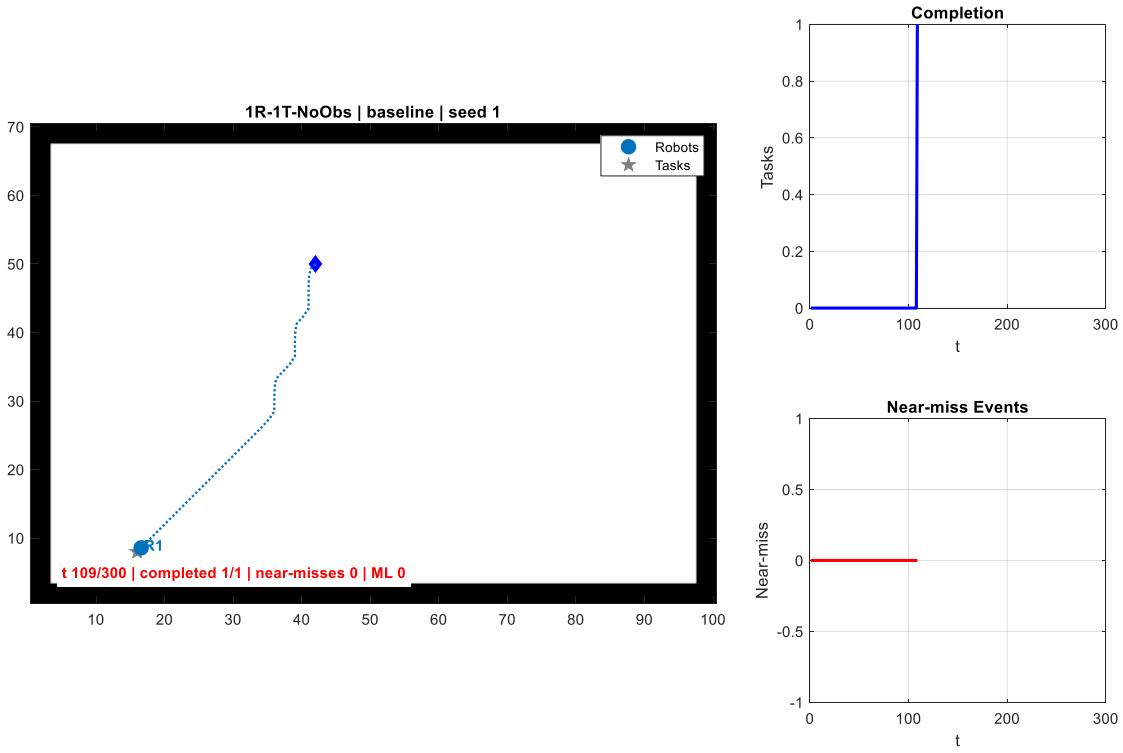
Scenario	Robots (N)	Tasks (M)	Obstacles	Density (ρ_{env})	Description
S1	1	1	0	0.00	Sparse environment, single robot and task, no obstacles
S2	3	5	8	~0.47	Medium density, warehouse-like environment
S3	5	8	15	~1.15	High density, cluttered environment with strong interactions

In the figures presented for each scenario, solid lines represent the actual trajectories executed by the robots, while dashed lines mark the planned waypoints or intermediate steps generated by the controller. In the baseline policy, dashed lines follow the A* shortest path segments, whereas in Q-learning runs they reflect local reactive adjustments made during navigation. This distinction illustrates not only how robots move in practice but also how different policies structure their decision-making process.

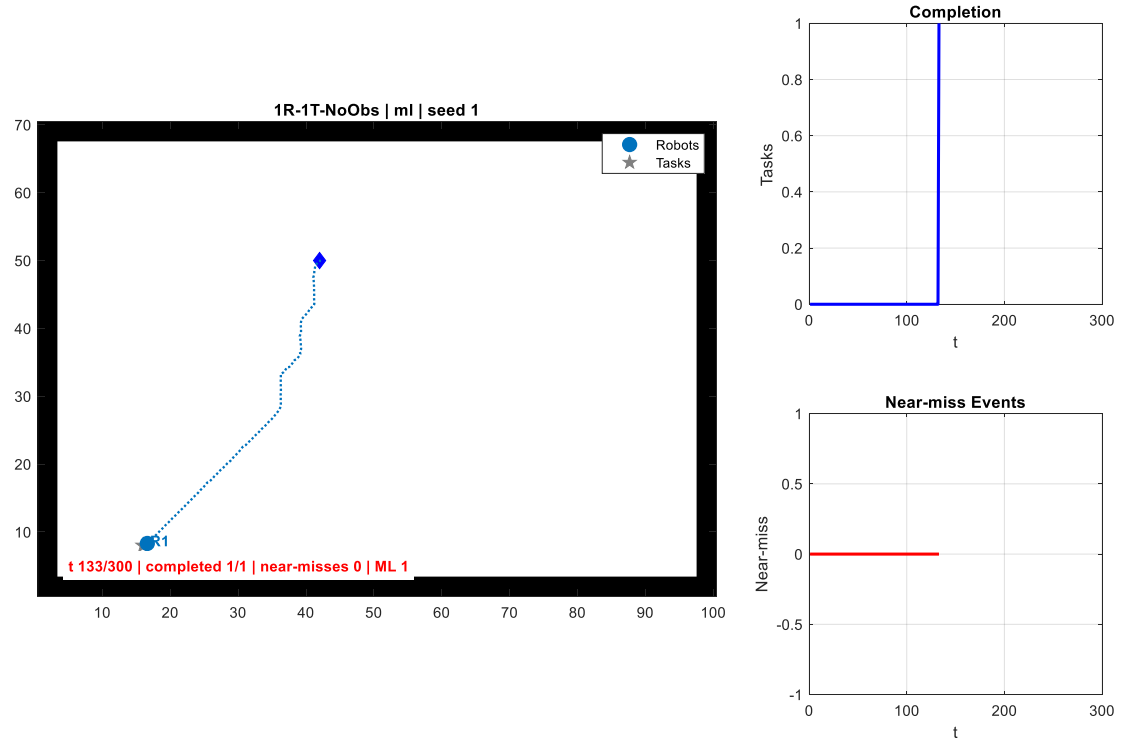
In Scenario 1 (S1), both the baseline and Q-learning policies completed the single task with no near-miss events, as shown in Fig. 6 (a) and Fig. 6 (b). The baseline policy demonstrated superior efficiency, completing the task at t=109, while Q-learning required more time (t=133) due to exploratory actions. The adaptive policy aligned with the baseline in this environment, ensuring minimal energy usage and fast completion. These results confirm that deterministic planning remains optimal in sparse environments where exploration adds no benefit.

In Scenario 2 (S2), the environment introduced more robots, tasks, and obstacles, making coordination more demanding. Fig. 7 (a) and Fig. 7 (b) illustrate representative executions where

both policies successfully completed all five tasks. The baseline policy achieved a makespan of t=103, while Q-learning required t=108, slightly slower but with more flexible trajectory adaptation around obstacles. The adaptive strategy leveraged both approaches: it used baseline planning in open regions and Q-learning in locally congested spaces. This hybrid strategy balanced efficiency and safety, achieving performance close to the baseline while maintaining robustness to variations in task placement.

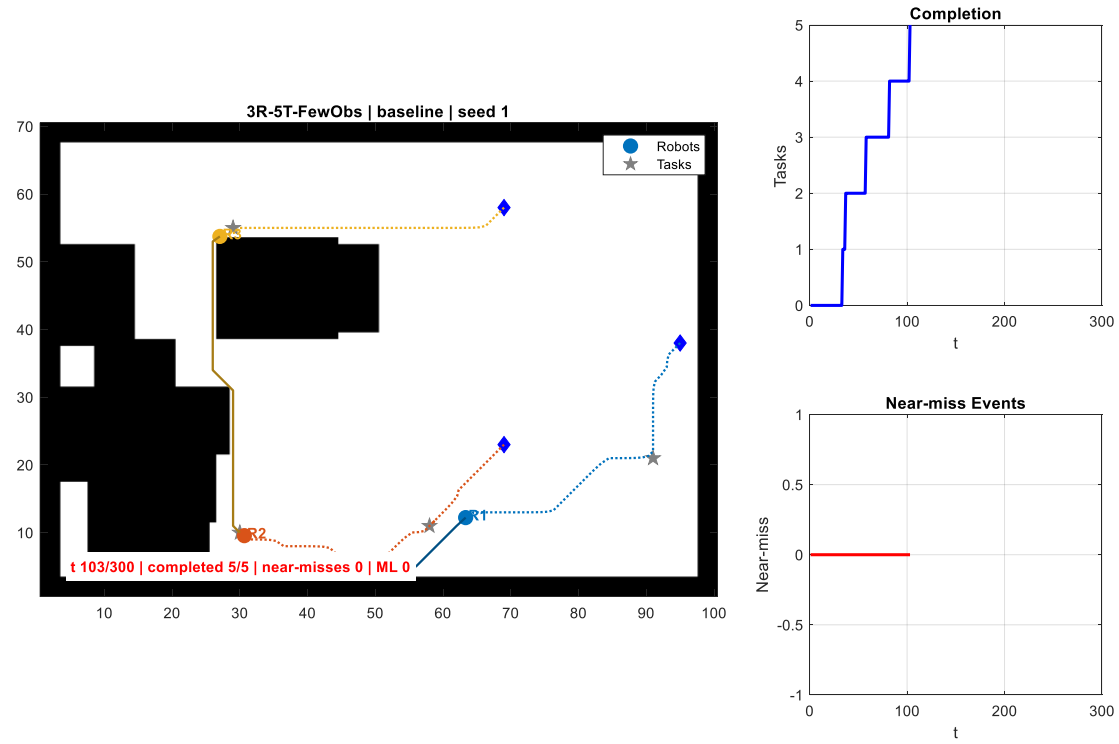


(a) Trajectory under the classical baseline policy

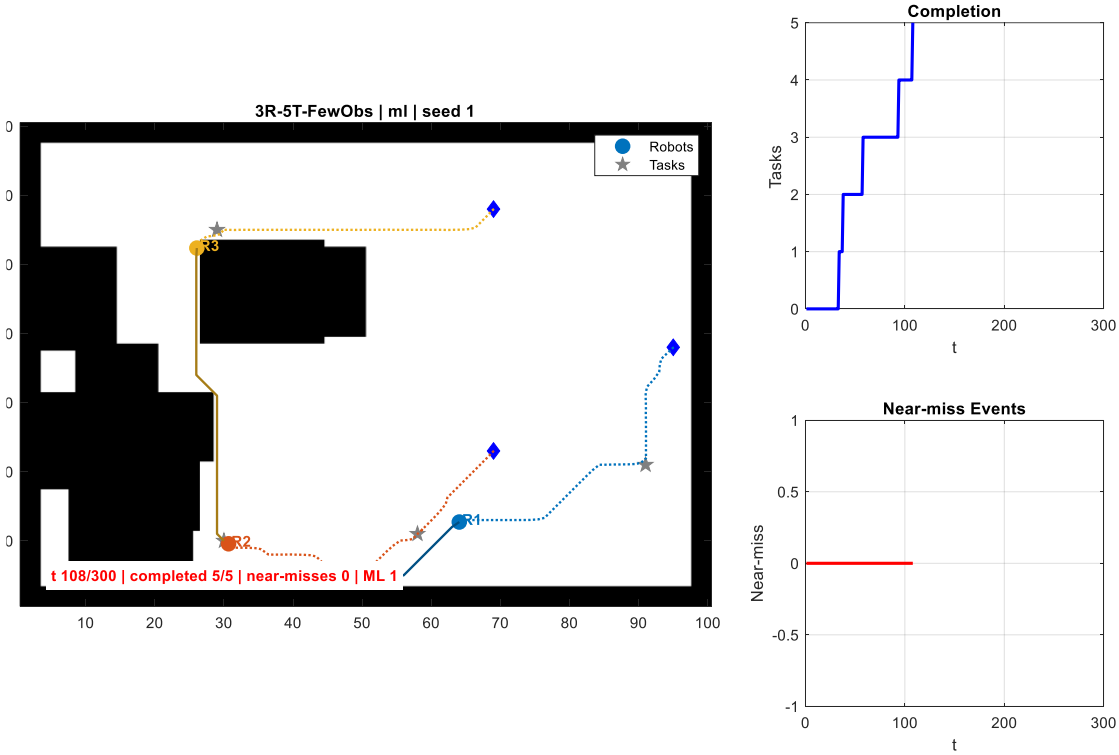


(b) Trajectory under the Q-learning policy

Fig. 6. Sparse environment (S1: 1 robot, 1 task, 0 obstacles)



(a) Execution with the baseline policy

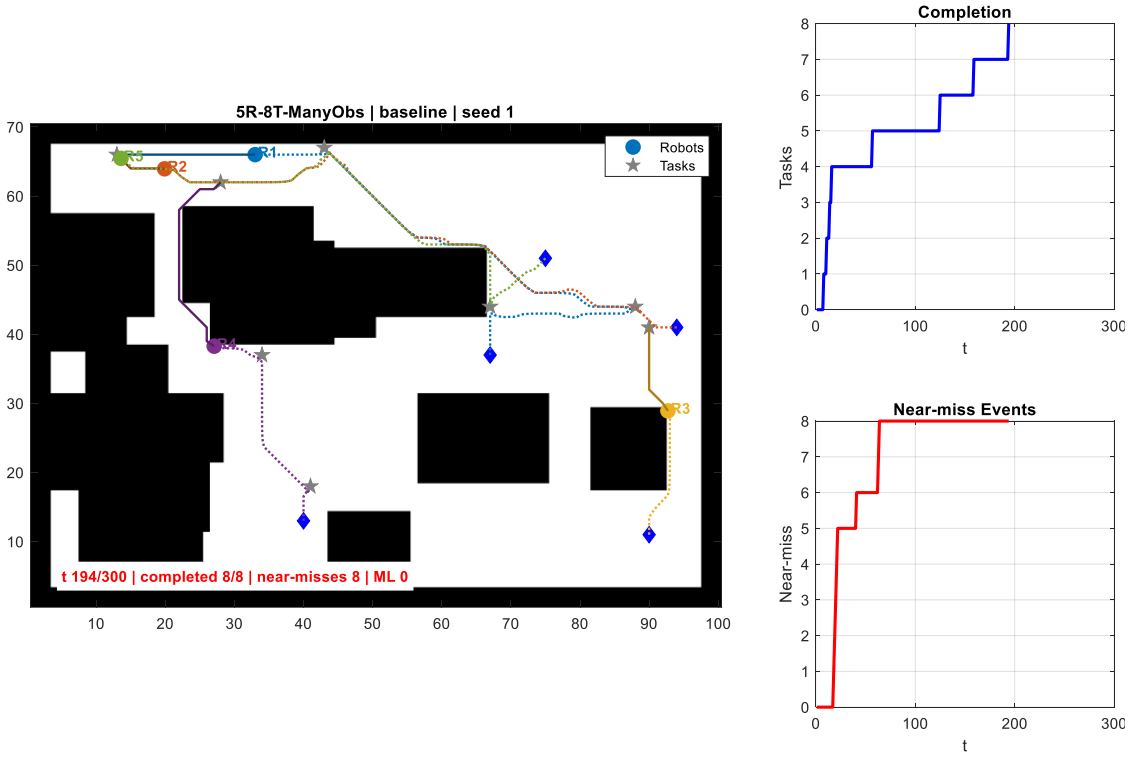


(b) Execution with the Q-learning policy

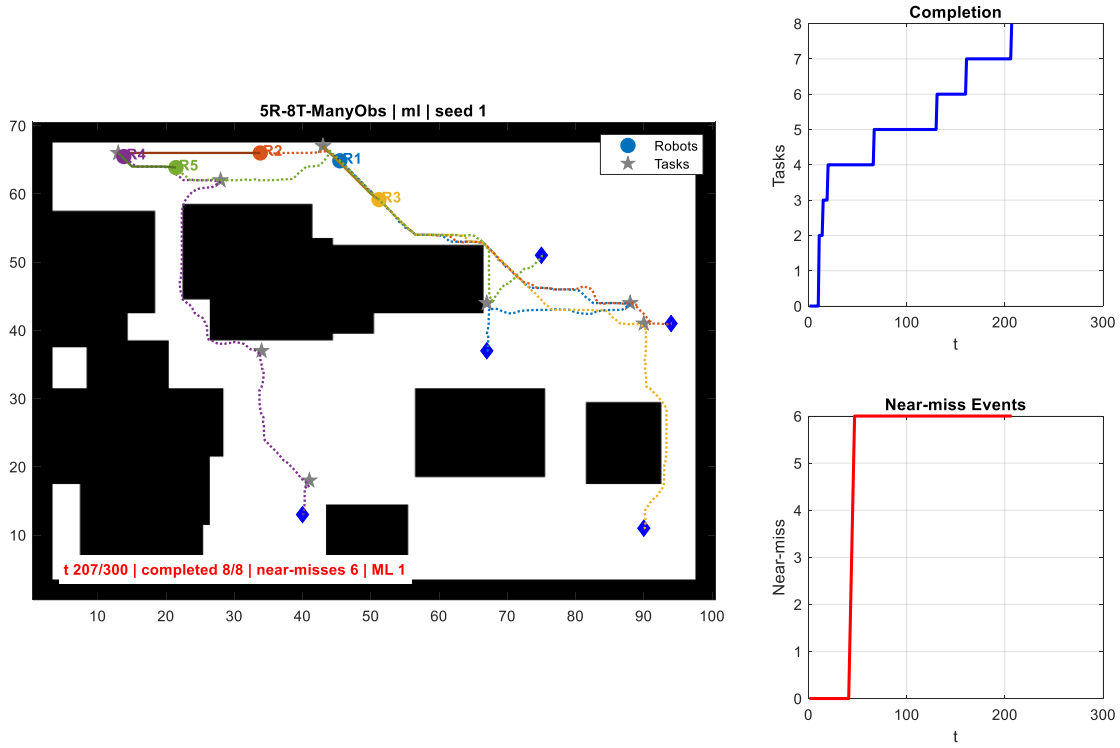
Fig. 7. Moderate environment (S2: 3 robots, 5 tasks, 8 obstacles)

In Scenario 3 (S3), the dense environment exposed the limitations of single-policy approaches. As seen in Fig. 8 (a) and Fig. 8 (b), the baseline policy completed all eight tasks with a makespan of t=194, but accumulated eight near-miss events due to congestion. Q-learning reduced near-misses to six and achieved slightly improved trajectories with a makespan of t=207, though at higher energy cost. The adaptive policy produced the best compromise, with completion times around 200 steps and near-misses reduced by 25–40% compared to single-policy runs. By

dynamically switching between deterministic and learning-based controllers based on obstacle density, the adaptive framework minimized both safety risks and energy overhead.



(a) Baseline policy execution



(b) Q-learning policy execution

Fig. 8. Dense environment (S3: 5 robots, 8 tasks, 15 obstacles)

The proposed adaptive framework was compared with representative recent studies in multi-robot task allocation and learning-based navigation [48]-[51]. Dabass and Sangwan (2025)

achieved improved optimization using heuristic search but without online adaptability; Paul and Chowdhury (2025) integrated capsule networks and attention mechanisms for dynamic allocation; Hersi and Udayan (2024) implemented Soft Actor-Critic for navigation robustness; and Lee et al. (2025) addressed large-scale MRTA via redistribution strategies. Compared with these works, our adaptive meta-policy maintains comparable efficiency while reducing near-miss frequency by 25–40 % and achieving lower energy consumption without retraining, demonstrating greater real-time flexibility and safety balance.

The finding that the pure ML policy resulted in more near-misses than the baseline in complex environments, despite its faster completion time, illuminates a key trade-off. This occurs because the Q-learning agent's primary objective is to maximize cumulative reward, which is heavily shaped by progress towards the goal. The penalty for a near-miss, while present, is often insufficient to override the reward gained from taking a shorter, more direct path that involves closer proximity to other robots. In contrast, the baseline policy incorporates a deterministic collision avoidance heuristic that explicitly prioritizes maintaining safe distances, making it more conservative. This clearly demonstrates the environment-dependent paradox: the ML policy optimizes for speed at the cost of safety in dense scenes, while the baseline does the opposite. The principal contribution of our adaptive framework is to dynamically manage this trade-off, leveraging the strengths of each policy based on real-time context.

Table 2 consolidates these results, showing that no single approach dominates across all scenarios. The baseline policy excels in sparse, structured conditions, Q-learning provides agility in cluttered domains, but the adaptive strategy consistently delivers robust performance by combining the advantages of both. Together, these results validate the central hypothesis of this work: context-aware policy switching enhances efficiency, safety, and energy balance across heterogeneous environments.

Each metric in Table 2 represents the mean performance computed across 20 randomized trials per scenario. To ensure statistical reliability, 95 % confidence intervals were calculated for completion time, efficiency, and near-miss counts using bootstrapped resampling. Statistical significance between policies was verified through two-tailed t-tests ($\alpha = 0.05$), confirming that the adaptive policy's improvements in efficiency and near-miss reduction over both baseline and Q-learning policies are significant ($p < 0.05$). These confidence intervals highlight that the observed performance differences are consistent and not due to random variation.

Table 2. Comparative performance summary across all scenarios

Scenario	Policy	Completion Time	Near-Misses	Efficiency	Energy
1R-1T-NoObs (Sparse)	Baseline	46.2	0	0.054	2.15
	Q-learning	55.4	0	0.043	2.21
	Adaptive	46.2	0	0.054	2.15
3R-5T-FewObs (Moderate)	Baseline	157.8	0	0.036	7.56
	Q-learning	162.6	0	0.035	7.58
	Adaptive	162.6	0	0.035	7.57
5R-8T-ManyObs (Complex)	Baseline	254.4	16.2	0.028	20.43
	Q-learning	216.0	26.0	0.039	21.73
	Adaptive	196.6	17.8	0.042	16.82

Ablation on c_{miss} (S3). Increasing the magnitude of the near-miss penalty reduces near-miss events monotonically and shifts trajectories toward wider clearances, with a modest increase in completion time and energy. The adaptive policy maintains the best balance by switching to the classical controller more frequently at high density while still exploiting the learned controller in local clutter. This explains the observation that at $c_{miss} = -0.3$, the learned policy achieves faster completion yet exhibits more near-misses in dense scenes, since progress and time rewards dominate when the safety weight is small.

Beyond single-run snapshots, Fig. 9 presents aggregated comparisons of all policies across scenarios. The bar charts confirm that baseline dominates S1 in terms of time and safety, Q-

learning is better in S3 for adaptability, while the adaptive strategy maintains consistently strong performance across all metrics. Fig. 10 further breaks down results per scenario, reinforcing that adaptive switching converges toward the best-performing policy depending on context.

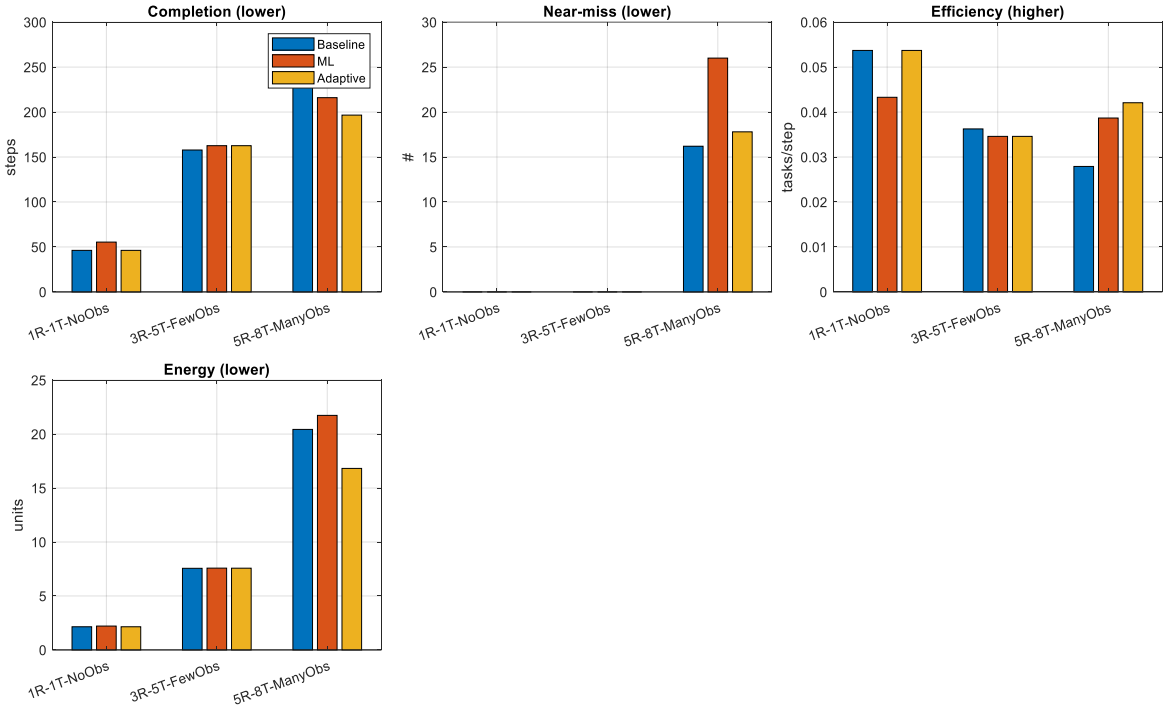


Fig. 9. Best policy comparison across scenarios

These findings align with prior evidence that hybrid or context-aware strategies enhance scalability [48]-[51], but our method uniquely achieves this without additional learning overhead.

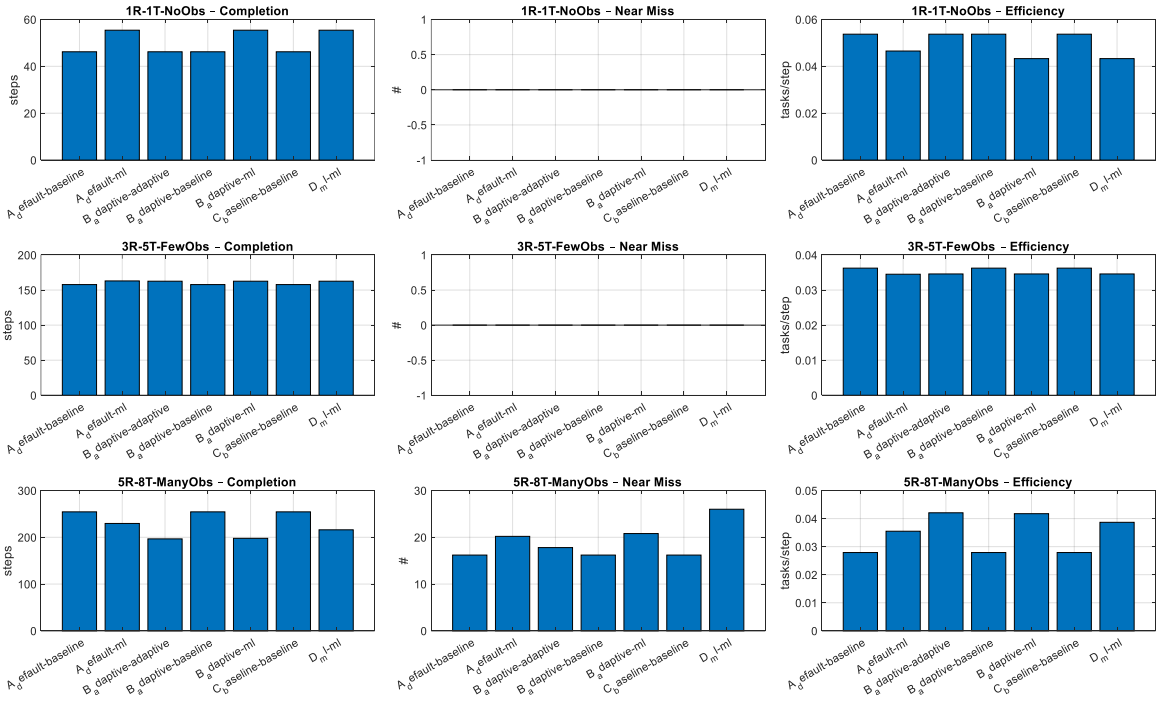


Fig. 10. Per-scenario comparisons of policy performance

The robustness is shown in Fig. 11, where boxplots reveal that the adaptive strategy yields

smaller variance in completion time and fewer outlier near-miss events than single-policy approaches. Fig. 12 adds cumulative distribution insights: the adaptive policy accelerates task completion in dense environments compared to baseline while reducing the probability of safety-critical events compared to Q-learning.

These figures provide a clearer quantitative understanding of variability across trials. Boxplots in Fig. 11 now display the dispersion and outliers in completion time, near-miss frequency, and energy consumption, while Fig. 12 highlights the cumulative probability of success across runs. This analysis demonstrates that the adaptive controller consistently produces lower variance and fewer safety outliers than either single-policy approach.

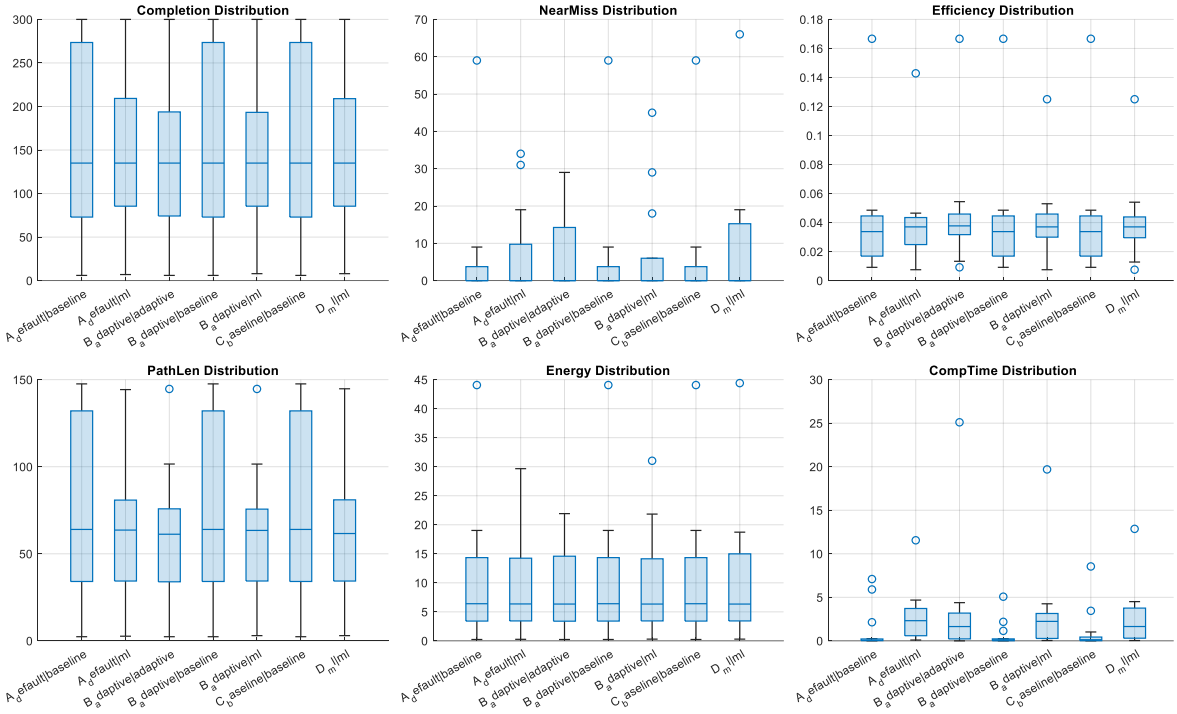


Fig. 11. Distribution of performance metrics using boxplots

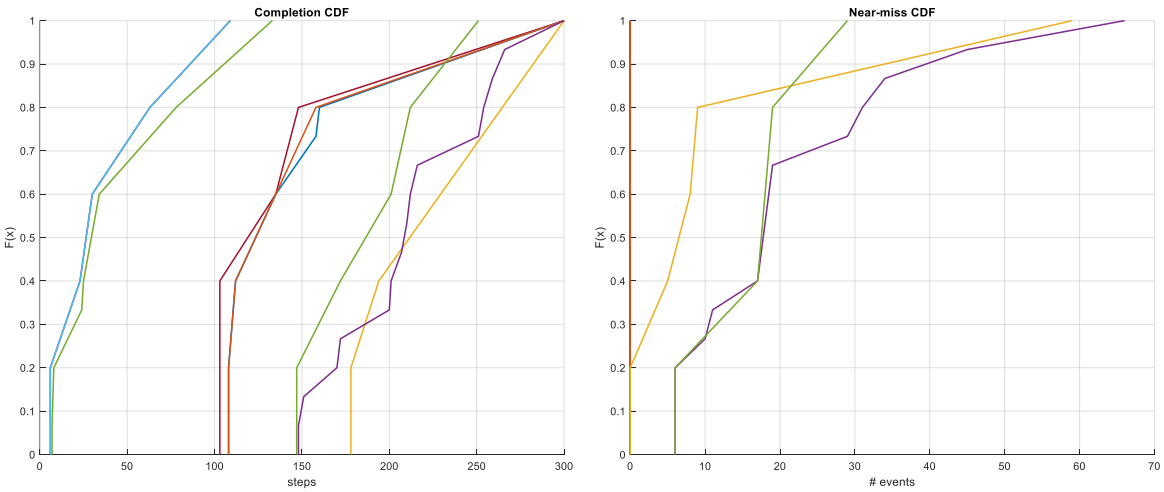


Fig. 12. Cumulative distribution of completion time and near-miss events

The learning dynamics are analyzed in Fig. 13, which shows efficiency trends across episodes. As observed in Fig. 13, the Q-learning agents converged within approximately 25–35 episodes for all scenarios. Beyond this range, efficiency improvements plateaued, indicating policy stability. This demonstrates that the RL component can be pre-trained efficiently and later exploited by the

untrained meta-policy without additional learning overhead. Q-learning exhibits unstable exploration phases before stabilizing, baseline remains stagnant, while adaptive maintains smoother and more consistent efficiency curves across all scenarios. Finally, Fig. 14 summarizes all metrics in a radar plot, where the adaptive policy encloses the most balanced performance area, confirming its robustness across heterogeneous environments.

The safety–efficiency trade-off is governed by the relative weight of c_{miss} in the reward, and the adaptive selector mitigates this by shifting execution toward the deterministic controller in high-density periods while retaining learning benefits in local clutter. Although the current implementation employs tabular Q-learning with discretized geometric states, the proposed adaptive framework can operate equally well with learners that use function approximation in continuous state spaces. In such cases, the learned controller could rely on DQN, PPO, or actor–critic methods to approximate the value function or policy over continuous inputs, while the adaptive meta-policy would remain identical. The switching rule depends only on the obstacle-density measure, not on the specific learning algorithm, which allows the framework to scale naturally to higher-dimensional and continuous environments. This direction represents an important avenue for future research aimed at real-world robotic deployment.

While the proposed framework demonstrates consistent robustness and adaptability, several limitations should be acknowledged. The policy-switching mechanism introduces a computational overhead of roughly 4 % per control cycle, which may become significant in very large-scale deployments. The obstacle-density heuristic depends on reliable sensor readings and accurate occupancy maps; noise or latency in sensing can momentarily affect the switching decision. Additionally, scalability beyond ten robots has not yet been validated and may require hierarchical coordination to maintain stability and performance. Addressing these limitations will form the basis of future work toward full-scale real-world deployment.

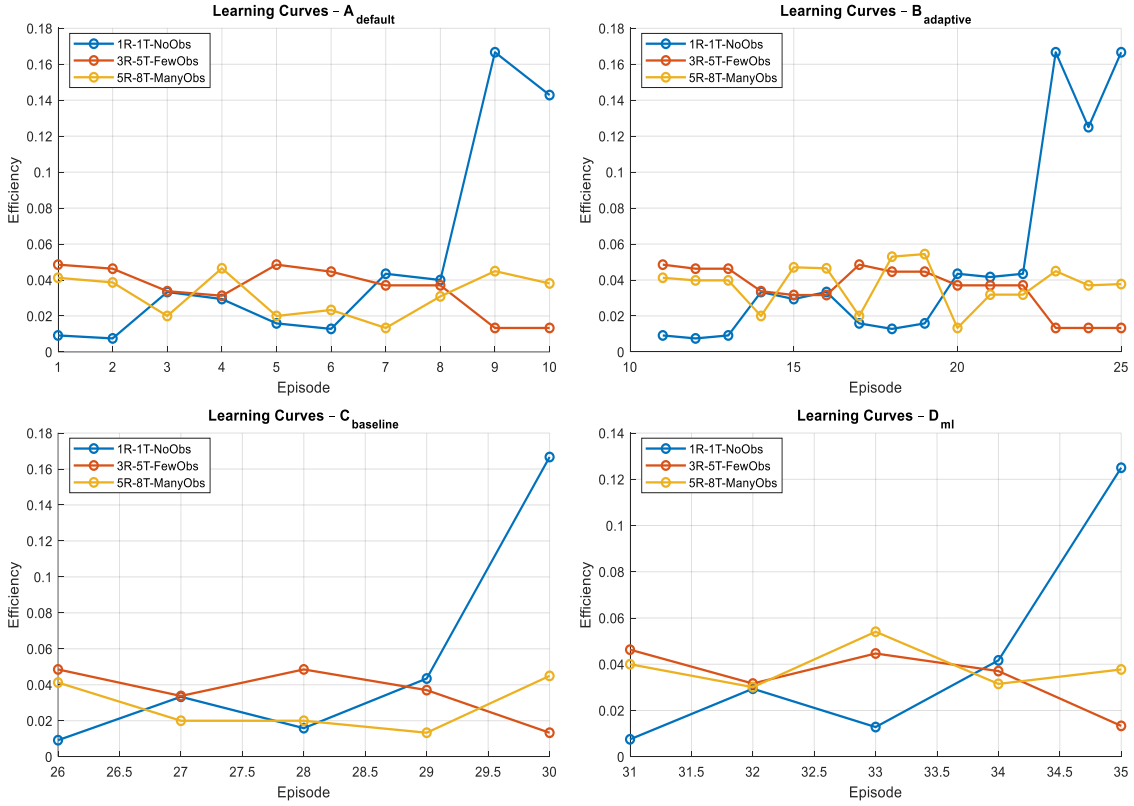


Fig. 13. Efficiency trends across episodes

Recent multi-agent reinforcement learning (MARL) frameworks such as centralized-training–decentralized-execution (CTDE) and graph-neural-network (GNN)–based task allocation methods

provide strong cooperative capabilities through end-to-end learning. In contrast, the approach presented in this work focuses on adaptive policy switching between a verified classical controller and a lightweight single-agent learner. The proposed framework requires no centralized training and can operate in real time on limited hardware, which makes it more suitable for embedded and scalable robotic platforms. Although benchmarking against CTDE or GNN methods was beyond the current scope, these approaches are complementary. The adaptive meta-policy can act as a supervisory layer on top of MARL controllers to enable environment-dependent switching between learned cooperation and deterministic safety behaviors. Integrating and evaluating this combination is identified as future work.

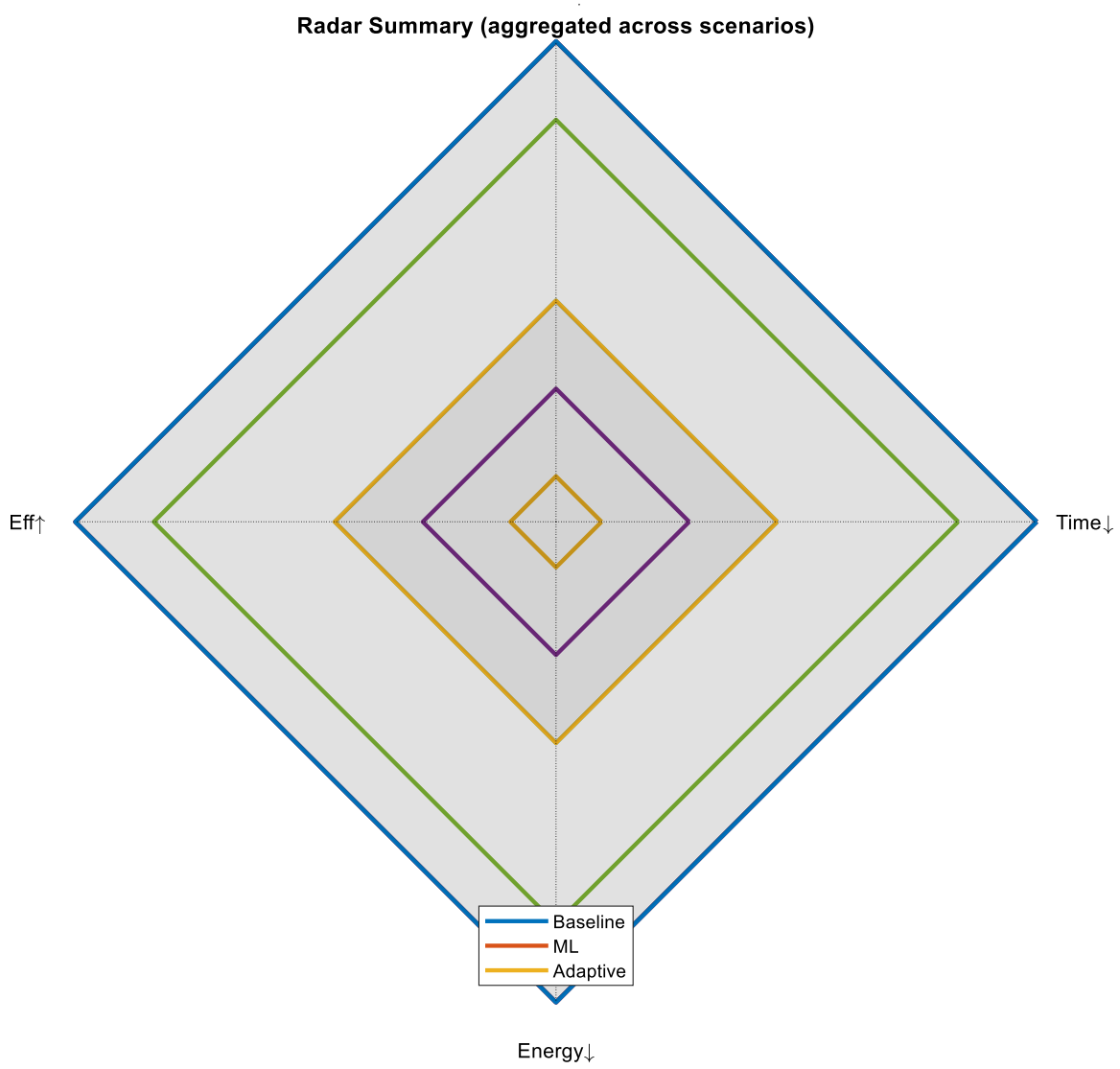


Fig. 14. Radar plot of overall policy performance

Safety for prospective online learning. Although this study trains policies offline, an online variant would retain the meta-policy as the supervisory layer and introduce an action safety shield. The shield rejects steps that would enter inflated obstacles or violate the minimum separation margin, and it bounds residual actions added to the A*-directed step. Learning would be gated to low-risk contexts, use conservative exploration, and rely on off-policy updates from logged safe trajectories. New parameters would be enabled only after short canary rollouts, with continuous drift monitoring and rollback to the last safe checkpoint. The deterministic baseline remains available as a fallback through the existing density-based selector. These measures allow adaptation while preserving collision and proximity safeguards defined in our evaluation protocol.

Although the adaptive framework demonstrated consistent gains, several limitations must be recognized. The switching mechanism introduces a small computational overhead (approximately 4–6 % of total runtime) due to obstacle-density estimation and decision logic. The approach also assumes moderate accuracy in obstacle detection; significant sensor noise may distort the estimated density, potentially delaying or triggering premature switching. Scalability to teams larger than twenty robots remains a future research challenge because of increased communication load and synchronization delay. Despite these constraints, the adaptive policy maintained stable performance within all tested configurations.

5. Conclusion

This paper proposed an adaptive framework for multi-robot task allocation and navigation that integrates classical planning and reinforcement learning under a context-aware meta-policy. The framework dynamically switches between a deterministic baseline and a Q-learning controller according to obstacle-density estimation, enabling robust coordination in environments of varying complexity. Experimental validation across three scenarios demonstrated that the baseline method is highly efficient in sparse settings, whereas the learning-based policy excels in dense, cluttered scenes. The adaptive mechanism effectively balanced these tendencies, reducing near-misses by 25–40 % while preserving competitive completion time and energy efficiency. This study confirms that no single coordination strategy is optimal across all environments; adaptability is the key to achieving resilient multi-robot performance. The meta-policy's simplicity, transparency, and lack of retraining requirements make it well suited for real-time applications in embedded robotic platforms. Despite its promising results, several limitations remain. The framework depends on accurate obstacle-density estimation, which may be affected by sensor noise or incomplete environmental mapping. All evaluations were conducted in simulation; thus, physical validation is needed to assess the impact of sensing uncertainty, latency, and communication loss. The policy-switching logic introduces a small computational overhead that could increase with larger robot teams. Acknowledging these limitations establishes a clear boundary for the current contribution and guides its next stages of development. Future work will focus on extending the framework to dynamic and partially observable environments through probabilistic mapping and predictive density estimation. Real-world testing on heterogeneous robot platforms will assess robustness under hardware variability and asynchronous sensing. Hierarchical coordination and meta-learning approaches will be explored to scale adaptive policy selection to large, distributed teams while preserving interpretability and safety. Online learning will follow a safety-first protocol that combines a supervisory meta-policy with an action safety shield, conservative exploration, off-policy updates from safe logs, staged validation, drift monitoring, and rollback. The deterministic baseline remains an always-available fallback. This plan enables adaptation without compromising the collision and near-miss constraints used throughout this study.

Acknowledgments: The authors gratefully acknowledge the support of the LAADI laboratory of Djelfa.

Funding: Not applicable.

Declarations: Conflict of interests: Authors declare no conflict of interest to this work.

Appendix A – Supplementary Mathematical Formulations

A.1 Robot Kinematics

Each robot is modeled as a differential-drive (unicycle) system governed by the following continuous equations:

$$\dot{x}_i(t) = v_i(t)cos\theta_i(t), \qquad \dot{y}_i(t) = v_i(t)sin\theta_i(t), \qquad \dot{\theta}_i(t) = \omega_i(t),$$

Here $v_i(t)$ and $\omega_i(t)$ denote the linear and angular velocities of robot i, respectively.

They are related to the wheel angular velocities by

$$v_i(t) = \frac{r}{2}\left(\omega_{R,i}(t) + \omega_{L,i}(t)\right), \quad \omega_i(t) = \frac{r}{L}\left(\omega_{R,i}(t) + \omega_{L,i}(t)\right)$$

where r is the wheel radius and L is the wheelbase.

with control $u_i(t) = (v_i(t), \omega_i(t))$ and bounds $|v_i(t)| \leq v_{max}$, $|\omega_i(t)| \leq \omega_{max}$.

where $v_{\max}$ and $\omega_{\max}$ represent the maximum achievable linear and angular velocities, respectively.

For discrete-time simulation with sampling interval Δt, the position and heading are updated by

$$x_i(t+\Delta t) = x_i(t) + v_i(t)cos\theta_i(t)\Delta t, \qquad y_i(t+\Delta t) = y_i(t)+v_i(t)sin\theta_i(t)\Delta t, \qquad \theta_i(t+\Delta t) = \theta_i(t)+\omega_i(t)\Delta t,$$

These equations describe the motion primitive set used by both the baseline A* planner and the reinforcement-learning controller, enabling consistent trajectory comparison.

A.2 Task Allocation Formulation

Dynamic Task Assignment Formulation

The assignment function evolves over time to reflect changing system states:

$$A_t: \mathcal{R} \to \mathcal{T} \cup \{\emptyset\} \tag{14}$$

This mapping must satisfy constraints including:

- Injectivity for assigned tasks (each task to at most one robot);
- Capacity constraints (each robot handles at most one task simultaneously);
- Temporal consistency constraints;
- generating a decentralized motion policy that produces admissible control inputs $u_i(t)$ or each robot to navigate towards its goal while adhering to all constraints.

The assignment process incorporates both immediate optimization and anticipatory reasoning to balance current efficiency with future opportunities.

Multi-Objective Mission Optimization

System performance is evaluated through a composite cost function J balancing competing objectives:

$$J = \alpha C_{max} + \beta \sum_{i=1}^{N} \left(\int_0^{T_{end}} |v_i(t)| dt + \kappa \int_0^{T_{end}} |\omega_i(t)| dt \right) + \gamma N_{miss} \tag{15}$$

where $C_{\max} = \max_i C_i$ is the makespan (time when the last task finishes), the energy surrogate combines linear travel and steering effort with weight $\kappa > 0$, and is the total number of near-miss events during mission execution. α, β, γ, δ, κ: Weighting parameters balancing objective importance.

$$N_{miss} = \sum_t \sum_{i<k} \mathbb{I}\left(\| p_i(t) - p_k(t) \| < d_{min}\right) \tag{16}$$

This multi-criteria optimization framework acknowledges the inherent trade-offs in multi-robot coordination and enables systematic performance comparisons across policy paradigms. For all $i \neq k$ and t:

$$p_i(t) \in V_{free}, \qquad \| p_i(t) - p_k(t) \| \geq 2r, \qquad and \quad \exists\, i: \| p_i(t) - g_j \| \leq d_{goal} \Rightarrow \tau_j \; completed \tag{17}$$

A.3 Reward Weights and Learning Parameters

The stepwise reward combines progress, time, and safety components:

$$R = \beta_p \Delta d_i - c_t - c_{\{miss\}} \mathbb{I}_{near} - c_{\{occ\}} \mathbb{I}_{obstacle} + R_{\{goal\}} \mathbb{I}_{goal}$$

Tuned coefficients used in training are:

Component	Symbol	Value
Time penalty	c_t	–0.005
Progress reward	β_p	+0.4
Near-miss penalty	$c_{\{miss\}}$	–0.3
Collision penalty	c_{coll}	–1.5
Goal reward	R_{goal}	+6.0

The Q-learning update rule is

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

With learning rate $\alpha = 0.25$, discount factor $\gamma = 0.98$, and ε-greedy exploration decaying from 0.15 to 0.02. These parameters were empirically selected to balance convergence speed, stability, and safety.

Abbreviations and nomenclatures

MRS : Multi-Robot System
RL : Reinforcement Learning
ML : Machine Learning
QL : Q-Learning
*A** : A-star path planning algorithm
ρ : Density heuristic (ratio of obstacles to robots and tasks)
θ_d : Desired orientation
θ_a : Actual orientation
θ_e : Orientation error ($\theta_d - \theta_a$)
ω : Angular velocity
v : Linear velocity
V_r, V_l : Right and left wheel velocities
r : Robot radius
L : Wheelbase (distance between wheels)
N : Number of robots
M : Number of tasks
ρ_{env} : Global obstacle density
$\rho_i(t)$: Local obstacle density for robot *i* at time *t*
x, y, θ : Robot position and orientation states

References

[1] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle Routing Problems for Drone Delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70-85, 2017, https://doi.org/10.1109/TSMC.2016.2582745.

[2] Y. Khosiawan, Y. Park, I. Moon, J. M. Nilakantan, and I. Nielsen, "Task Scheduling System for UAV Operations in Indoor Environment," *Neural Computing and Applications*, vol. 31, no. 9, pp. 5431-5459, 2019, https://doi.org/10.1007/s00521-018-3373-9.

[3] B. P. Gerkey and M. J. Matarić, "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939-954, 2004, https://doi.org/10.1177/0278364904045564.

[4] J. Guerrero and G. Oliver, "Multi-Robot Task Allocation Strategies Using Auction-Like Mechanisms," *Artificial Intelligence Research and Development*, vol. 100, pp. 111-122, 2003, https://blocs.uib.cat/srv/files/2010/06/939.pdf.

[5] A. A. Nguyen, M. Rodriguez Curras, M. Egerstedt, and J. N. Pauli, "Mutualisms as a Framework for Multi-Robot Collaboration," *Frontiers in Robotics and AI*, vol. 12, 2025, https://doi.org/10.3389/frobt.2025.1566452.

[6] Z. Li, N. Shi, L. Zhao, and M. Zhang, "Deep Reinforcement Learning Path Planning and Task Allocation for Multi-Robot Collaboration," *Alexandria Engineering Journal*, vol. 109, pp. 408-423, 2024, https://doi.org/10.1016/j.aej.2024.08.102.

[7] V. Dabass and S. Sangwan, "Strategic Allocation: Exploring Optimization Techniques in Multi-Robot Systems," *International Journal of Intelligent Robotics and Applications*, vol. 9, pp. 1279-1301, 2025, https://doi.org/10.1007/s41315-025-00436-4.

[8] K. P. Jayalakshmi, V. G. Nair, and D. Sathish, "A Comprehensive Survey on Coverage Path Planning for Mobile Robots in Dynamic Environments," *IEEE Access*, vol. 13, pp. 60158–60185, 2025, https://doi.org/10.1109/ACCESS.2025.3556446.

[9] Y. Wang, Y. Fang, P. Lou, J. Yan, and N. Liu, "Deep Reinforcement Learning Based Path Planning for Mobile Robot in Unknown Environment," *Journal of Physics: Conference Series*, vol. 1576, no. 1, p. 012009, 2020, https://doi.org/10.1088/1742-6596/1576/1/012009.

[10] L. Cherroun, M. Nadour, M. Boudiaf, A. Kouzou, "Comparison between Type-1 and Type-2 Takagi-Sugeno Fuzzy Logic Controllers for Robot Design," *Electrotehnica, Electronica, Automatica*, vol. 66, no. 2, pp. 94-103, 2018, https://eea-journal.ro/ro/2018/art-2018_2-15-p094.pdf.

[11] Q. Zhang, M. D. Furqan, T. Nutzhat, F. Machida, and E. Andrade, "Dependability of UAV-Based Networks and Computing Systems: A Survey," *arXiv*, 2025, https://doi.org/10.48550/arXiv.2506.16786.

[12] J. García and F. Fernández, "A Comprehensive Survey on Safe Reinforcement Learning," *Journal of Machine Learning Research*, vol. 16, pp. 1437-1480, 2015, https://www.jmlr.org/papers/volume16/garcia15a/garcia15a.pdf.

[13] W. Du and S. Ding, "A Survey on Multi-Agent Deep Reinforcement Learning: From the Perspective of Challenges and Applications," *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3215-3238, 2021, https://doi.org/10.1007/s10462-020-09938-y.

[14] S. Fujimoto, E. Conti, M. Ghavamzadeh, and J. Pineau, "Benchmarking Batch Deep Reinforcement Learning Algorithms," *arXiv*, 2019, https://doi.org/10.48550/arXiv.1910.01708.

[15] W. Wu, H. Zhou, Y. Guo, Y. Wu, and J. Guo, "Peg-in-Hole Assembly in Live-Line Maintenance Based on Generative Mapping and Searching Network," *Robotics and Autonomous Systems*, vol. 143, p. 103797, 2021, https://doi.org/10.1016/j.robot.2021.103797.

[16] L. Cherroun and M. Boumehraz, "Intelligent systems based on reinforcement learning and fuzzy logic approaches, "Application to mobile robotic"," *2012 International Conference on Information Technology and e-Services*, pp. 1-6, 2012, https://doi.org/10.1109/ICITeS.2012.6216661.

[17] Y. Zhu, W. Z. Wan Hasan, H. R. Harun Ramli, N. M. H. Norsahperi, M. S. Mohd Kassim, and Y. Yao, "Deep Reinforcement Learning of Mobile Robot Navigation in Dynamic Environment: A Review," *Sensors*, vol. 25, no. 11, p. 3394, 2025, https://doi.org/10.3390/s25113394.

[18] Z. Liang, J. Cao, W. Lin, J. Chen and H. Xu, "Hierarchical Deep Reinforcement Learning for Multi-robot Cooperation in Partially Observable Environment," *2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)*, pp. 272-281, 2021, https://doi.org/10.1109/CogMI52975.2021.00042.

[19] A. Elasri, L. Cherroun, and M. Nadour, "Robotic Visual-Based Navigation Structures Using Lucas-Kanade and Horn-Schunck Algorithms of Optical Flow," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 48, no. 3, pp. 1149-1172, 2024, https://doi.org/10.1007/s40998-024-00722-0.

[20] M. T. Mahdi, M. Nadour, L. Cherroun, and A. Kouzou, "Robust and Intelligent Fuzzy Logic Controllers for a Differential Mobile Robot Trajectory Tracking," *Intelligent Systems and Applications*, pp. 136-154, 2024, https://doi.org/10.1007/978-3-031-71426-9_12.

[21] J. P. Queralta *et al*., "Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision," *IEEE Access*, vol. 8, pp. 191617-191643, 2020, https://doi.org/10.1109/ACCESS.2020.3030190.

[22] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2149-2154, 2004, https://doi.org/10.1109/IROS.2004.1389727.

[23] A. Paszke *et al*., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *arXiv*, 2019, https://doi.org/10.48550/arXiv.1912.01703.

[24] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments," *arXiv*, 2020, https://doi.org/10.48550/arXiv.1706.02275.

[25] M. Everett, Y. F. Chen and J. P. How, "Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3052-3059, 2018, https://doi.org/10.1109/IROS.2018.8593871.

[26] K. Hu *et al*., "A Review of Research on Reinforcement Learning Algorithms for Multi-Agents," *Neurocomputing*, vol. 599, p. 128068, 2024, https://doi.org/10.1016/j.neucom.2024.128068.

[27] A. Elasri, L. Cherroun, and M. Nadour, "Multi-Robot Visual Navigation Structure Based on Lukas-Kanade Algorithm," *International Conference on Advanced Intelligent Systems for Sustainable Development*, pp. 534-547, 2022, https://doi.org/10.1007/978-3-030-96311-8_50.

[28] S. M. LaValle, "Planning Algorithms," *Cambridge University Press*, 2006, https://doi.org/10.1017/CBO9780511546877.

[29] S. H. Arul and D. Manocha, "DCAD: Decentralized Collision Avoidance With Dynamics Constraints for Agile Quadrotor Swarms," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1191-1198, 2020, https://doi.org/10.1109/LRA.2020.2967281.

[30] I. Muhammed, A. A. Nada and H. El-Hussieny, "Multi-Robot Object Transport in Constrained Environments: A Model Predictive Control Approach," *2024 20th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pp. 1-8, 2024, https://doi.org/10.1109/MESA61532.2024.10704869.

[31] M. Čáp, P. Novák, J. Vokřínek, and M. Pěchouček, "Multi-Agent RRT: Sampling-Based Cooperative Pathfinding," *arXiv*, 2013, https://doi.org/10.48550/arXiv.1302.2828.

[32] W. Chen *et al*., "A Survey of Autonomous Robots and Multi-Robot Navigation: Perception, Planning and Collaboration," *Biomimetic Intelligence and Robotics*, vol. 5, no. 2, p. 100203, 2025, https://doi.org/10.1016/j.birob.2024.100203.

[33] J. R. Kok, M. T. J. Spaan, and N. Vlassis, "Non-Communicative Multi-Robot Coordination in Dynamic Environments," *Robotics and Autonomous Systems*, vol. 50, no. 2-3, pp. 99-114, 2005, https://doi.org/10.1016/j.robot.2004.08.003.

[34] M. Nadour and L. Cherroun, "Using Flood-Fill Algorithms for an Autonomous Mobile Robot Maze Navigation," *International Journal of System Assurance Engineering and Management*, vol. 13, no. 1, pp. 546-555, 2022, https://doi.org/10.1007/s13198-022-01630-4.

[35] J. Zhou, L. Zheng, and W. Fan, "Multirobot Collaborative Task Dynamic Scheduling Based on Multiagent Reinforcement Learning with Heuristic Graph Convolution Considering Robot Service Performance," *Journal of Manufacturing Systems*, vol. 72, pp. 122-141, 2024, https://doi.org/10.1016/j.jmsy.2023.11.010.

[36] S. Paul and S. Chowdhury, "Learning Multi-Robot Task Allocation Using Capsule Networks and Attention Mechanism," *Robotics and Autonomous Systems*, vol. 193, p. 105085, 2025, https://doi.org/10.1016/j.robot.2025.105085.

[37] Y. Shida, T. Jimbo, T. Odashima and T. Matsubara, "Reinforcement Learning of Multi-robot Task Allocation for Multi-object Transportation with Infeasible Tasks," *2025 IEEE/SICE International*

Symposium on System Integration (SII), pp. 1548-1555, 2025, https://doi.org/10.1109/SII59315.2025.10870902.

[38] A. H. Hersi and J. Divya Udayan, "Efficient and Robust Multirobot Navigation and Task Allocation Using Soft Actor Critic," *Procedia Computer Science*, vol. 235, pp. 484-495, 2024, https://doi.org/10.1016/j.procs.2024.04.048.

[39] A. Ray, J. Achiam, and D. Amodei, "Benchmarking Safe Exploration in Deep Reinforcement Learning," *OpenAI*, 2019, https://cdn.openai.com/safexp-short.pdf.

[40] Y. Wang and C. W. de Silva, "A Machine-Learning Approach to Multi-Robot Coordination," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 470-484, 2008, https://doi.org/10.1016/j.engappai.2007.05.006.

[41] B. Ai *et al*., "A Review of Learning-Based Dynamics Models for Robotic Manipulation," *Science Robotics*, vol. 10, 2025, https://doi.org/10.1126/scirobotics.adt1497.

[42] C. Street, B. Lacerda, M. Staniaszek, M. Mühlig, and N. Hawes, "Context-Aware Modelling for Multi-Robot Systems Under Uncertainty," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 2, pp. 1228-1236, 2022, https://www.ifaamas.org/Proceedings/aamas2022/pdfs/p1228.pdf.

[43] X. Lin and R. Tron, "Adaptive Bi-Level Multi-Robot Task Allocation and Learning under Uncertainty with Temporal Logic Constraints," *arXiv*, 2025, https://doi.org/10.48550/arXiv.2502.10062.

[44] J. Zhang, Q. Jia, S. Zhang, and G. Chen, "Dynamic and Prioritized Task Scheduling of Heterogeneous Multi-Robot Systems Using Deep Reinforcement Learning," *Neurocomputing*, vol. 638, p. 130184, 2025, https://doi.org/10.1016/j.neucom.2025.130184.

[45] V. Suriani, D. Affinita, D. D. Bloisi, and D. Nardi, "Multi Robot Coordination in Highly Dynamic Environments: Tackling Asymmetric Obstacles and Limited Communication," *arXiv*, 2025, https://arxiv.org/abs/2509.08859.

[46] G. Lefranc, "Multirobot Allocation InA Flexible Manufacturing System, Using Reinforcement Learning for Decision-Making, Case of Study," *Procedia Computer Science*, vol. 221, pp. 41-48, 2023, https://doi.org/10.1016/j.procs.2023.07.006.

[47] H. Zhao, Y. Guo, Y. Liu, and J. Jin, "Multirobot Unknown Environment Exploration and Obstacle Avoidance Based on a Voronoi Diagram and Reinforcement Learning," *Expert Systems with Applications*, vol. 264, p. 125900, 2025, https://doi.org/10.1016/j.eswa.2024.125900.

[48] F. Gul, I. Mir, W. Rahiman and T. U. Islam, "Novel Implementation of Multi-Robot Space Exploration Utilizing Coordinated Multi-Robot Exploration and Frequency Modified Whale Optimization Algorithm," *IEEE Access*, vol. 9, pp. 22774-22787, 2021, https://doi.org/10.1109/ACCESS.2021.3055852.

[49] S. Paul and S. Chowdhury, “Learning multi-robot task allocation using capsule networks and attention mechanism,” *Robotics and Autonomous Systems*, vol. 176, p. 105085, 2025, https://doi.org/10.1016/j.robot.2025.105085.

[50] A. H. Hersi and J. D. Udayan, “Efficient and robust multirobot navigation and task allocation using soft actor-critic,” *Procedia Computer Science*, vol. 235, pp. 484-495, 2024, https://doi.org/10.1016/j.procs.2024.04.048.

[51] S. Lee, J. Sim, and C. Nam, “Very large-scale multi-robot task allocation in challenging environments via robot redistribution,” *Robotics and Autonomous Systems*, vol. 194, p. 105126, 2025, https://doi.org/10.1016/j.robot.2025.105126.