Research Square

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

A Distributed Learning Framework for Energy-Aware Cooperative Navigation in Multi- Robot Systems

Mohamed Shili

mohamed.shili@fst.utm.tn

Polytechnic School of Tunis

Research Article

Keywords: Multi-Robot Systems, Deep Reinforcement Learning, Distributed AI, Energy Efficiency, Cooperative Navigation, Multi-Agent Systems

Posted Date: November 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11578692/v1

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

Additional Declarations: No competing interests reported.

A Distributed Learning Framework for Energy-Aware Cooperative Navigation in Multi-Robot Systems

Mohamed Shili [1]

[1] Laboratory of Advanced Systems (LSA), Polytechnic School of Tunis, Al Marsa 2078, Tunisia; mohamed.shili@fst.utm.tn

Abstract: Cooperative Navigation Problem with Multiple Robots is known to be a quite challenging task, especially under unknown and dynamic traffic conditions. This happens since robots operating under these circumstances usually suffer from a lack of communication, inability to observe the whole environment, and a short lifetime due to a low level of available battery power. Traditional approaches to centralized control and planning for multi-robot navigation often fail here as they do not scale well and carry extremely large computational costs. A team of researchers has come forward to offer a new solution based on the concept of Energy-Efficient Distributed Deep Reinforcement Learning (ED-DDRL). In particular, their method involves developing a much-decentralized decision-making process, which means that each robot independently learns its own policy while communicating with other neighboring ones through the minimal amount of shared local data. To make sure all operations work smoothly without wasting too much electricity, the authors proposed a unique reward function taking into account various factors affecting robot movement. Thanks to an advanced lightweight communication scheme developed specifically for this system, ED-DDRL significantly reduces inter-robot bandwidth utilization levels. All in all, tests performed extensively within a physical prototype showed that compared with state-of-the-art solutions like DQN, PPO, MAPPO, and even MADDPG, ED-DDRL excels in several critical areas – achieving a higher overall success rate alongside enhanced energy efficiency metrics. The latter was most evident during trials involving environments packed with lots of bots where ED-DDRL easily managed to keep ahead.

Keywords: Multi-Robot Systems, Deep Reinforcement Learning, Distributed AI, Energy Efficiency, Cooperative Navigation, Multi-Agent Systems.

1. Introduction

Multi-robot navigation has become a significantly researched area in autonomous systems due to its various applications such as warehouse automation, surveillance, search and rescue operations, and intelligent transportation systems. However, there are significant challenges associated with coordinating multiple robots in dynamic and unknown environments due to the existence of uncertainties present in the environment, partial observability, as well as the need for effective collaboration [1], [2].

Traditional methods of path planning such as A*, potential fields, and ORCA have traditionally been used for robot navigation; however, these algorithms typically rely on pre-defined models that lack adaptability in a dynamic environment, resulting in limited performance in realistic scenarios that may have moving obstacles (i.e., masses of people) and multiple interacting agents [3], [4].

To provide an overall example of the system architecture, the proposed Energy-Efficient Distributed Deep Reinforcement Learning (ED-DDRL) framework is illustrated in Fig. 1. Robots perceive the environment independently via onboard sensors to build a local representation of the current state and then execute actions in accordance with the distributed reinforcement learning policy. A sparse

communication mechanism allows for selective sharing of information between robots in a close proximity. An energy-aware reward function helps guide the robots in their learning process to perform more efficiently, safely, and at greater scalability in a dynamic environment.

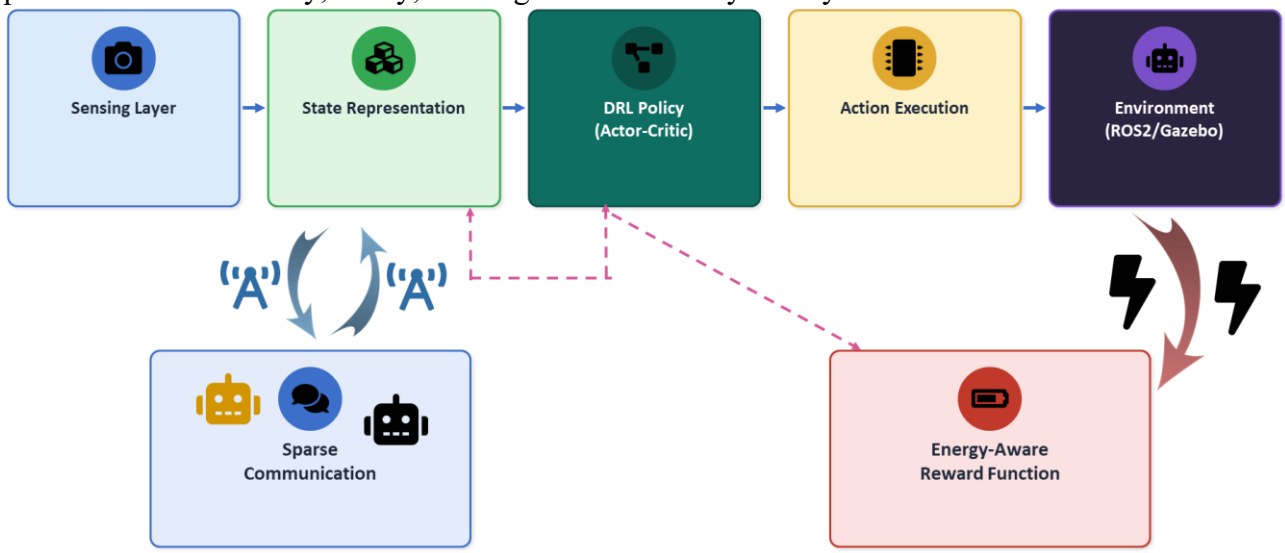


Fig. 1. Energy-Efficient Distributed Deep Reinforcement Learning Framework

With this system overview done, I have introduced deep reinforcement (DRL) for robotic navigation tasks. Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and Deep Deterministic Policy Gradient (DDPG) are typical methods that have achieved success at learning-based control problems using DRL. However, these methods have been created with single agent systems in mind and they do not consider multi-robot cooperation [5],[6].

Likewise, multi-agent reinforcement learning (MARL) algorithms such as MAPPO, MADDPG, and QMIX extend DRL for cooperative scenarios; however, their success has been limited by centralized training or high communication overhead, which in turn limits the scalability and limits the feasibility of real-time implementation in large scale robotic applications [7],[8].

Based on the limitations of these algorithms I am creating Energy-Efficient Distributed Deep Reinforcement Learning (ED-DDRL) framework for cooperative multi-robot navigation. The approach is able to provide fully decentralized learning, minimal communication cost, and energy-aware decision-making, making it applicable for large scale and dynamic environments [9].

Additionally, recent research has shown that the effectiveness of communication and overall power usage are significant in determining the performance of long-term autonomous operations. Mobile robots that rely on limited battery capacity for their function will benefit from optimizing both their navigation performance as well as their computational load and wireless communications, which are key to mission length and the overall integrity of the system [10]–[12].

A newer area of research is Distributed Intelligence in which each robot learns its own policies while only communicating with its nearest neighbor and sharing only the most relevant information. The integration of distributed reinforcement learning with sparse communications methods has shown excellent scalability and robustness when faced with communication failures, thus making this method highly appropriate when operating in large groups of robots in areas with limited visibility [13]–[15].

A study has been performed concurrently with evaluation of effectiveness of using Graph Neural Networks, Attention Mechanism, and Transformer based techniques for enhancing performance of autonomous robots working collaboratively. Improved collaboration through increased efficiency of information exchanged between robots and optimized communications through improved collaborative decision making has been achieved through these advanced techniques; however their operational complexity is still high compared to those used on real-time embedded robotic systems [16], [17].

Given these challenges, this paper presents a framework for Energy Efficient Distributed Deep Reinforcement Learning (ED-DDRL) cooperative multi-robot navigation in dynamic environments. Unlike traditional MARL methods, our framework converges the following elements into a unified architecture: decentralized policy learning, sparse event triggered communication and energy aware reward functions. Each robot will learn to navigate using only local observations and distribute/share information with other robots only when needed, resulting in significant communication reduction while improving scalability.

This paper has made the following contributions:

- A fully distributed deep reinforcement learning (DRL) framework for multi-robot navigation
- An energy aware reward balance between efficiency, safety, and cooperation
- A low bandwidth communication mechanism
- An extensive evaluation on dynamic environments with multiple robots
- Scalability analysis up to large robot populations.

The subsequent sections of this paper are organized as follows. In Section 2 we review the previous research conducted on cooperative multi-robot navigation; In Section 3 we describe our proposed ED-DDRL framework and the associated mathematical models; In Section 4 we describe our experimental setup and the results obtained from our simulations; In Section 5 we discuss the results obtained and the limitations presently observed in our approach; finally, in Section 6 we give an overall summary of the findings of this research and outline potential areas for future study.

2. Recent Works

There have been many advancements made in utilizing AI and deep reinforcement learning to assist autonomous robotics systems in working better together when navigating through areas [18]. But as promising as this technology may seem, there are yet to be a number of hurdles that need to be overcome for us to reach our objectives of establishing scalable, energy efficient, and decentralized co-ordination between a number of robots operating in dynamic environments where there is some level of partial observability (unable to see everything due to blockage). Other issues include constraints such as limited communication bandwidths and large energy usage requirements [19]. Current research reveals that there continues to be challenges faced by multi-robot learning frameworks to achieve an optimal decision making balance point between factors such as navigation effectiveness, communication efficiency and energy consumption [21].

Some conventional approaches such as A* search algorithms, Artificial Potential Fields (APF) and Optimal Reciprocal Collision Avoidance (ORCA) are quite popular among developers for reasons related to ease of implementation and ability to return real time results [18]. However, despite their popularity, these traditional methodologies also come with significant disadvantages given that they largely depend on predetermined models and heuristic rules to determine the behaviors exhibited by

robots. This limits their suitability in cases involving rapidly changing and/or unpredictable circumstances [19]. One of the primary drawbacks of employing these conventional techniques is that when a high density of robots exist within an environment, they often lead to congestion problems and potential coordination failures due to lack of learning based decision making mechanisms [18].

Although there are many existing DRL algorithms, they only show good performance for single-robots' navigations under difficult/uncertain situations [20]. When extended to multiple-robot scenarios, these DRL approaches face several issues including training instabilities, non-stationary environment, and the exponential expansion of the sizes of the joint state/action spaces between all robots [19]. Moreover, most existing DRL implementations fail to consider communication cost and/or energy constraints which are required in practical applications involving robotic systems [18].

Multi-Agent Reinforcement Learning (MARL) extends DRL towards cooperative multi-robot systems. For example, classical MARL systems include (1) Multi-Agents Proximate Policy Optimization (MAPPO) [23], (2) Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [24], and (3) QMIX [25] which implement different forms of cooperative multi-agent learning through the use of centralized training/decentralized execution (CTDE) architectures. Although many MARL systems offer potential advantages, their requirement of a central critic or great amounts of communications makes them computation heavy and hard to scale to bigger systems [22]. Likewise, few MARL implementations effectively deal with the energy-aware optimization aspect [21].

Overall, from the literature above we see that current DRL/MARL methods suffer from certain shortcomings [19]. Specifically, classical methodologies lack adaptability features, while DRL techniques suffer limitations on robot cooperation between multiple-robot units. In addition, MARL models are affected by communication/scale constraints. Thus, there emerges a critical need for fully-decentralized, communication-efficient and energy-efficient learning methods [22].

This paper presents a novel framework called Energy-Efficient Distributed Deep Reinforcement Learning (ED-DDRL) that incorporates the elements of decentralized policy learning, event-triggered sparse communication, and energy-efficient reward design to allow scalable multi-robot navigation within dynamic environments [23].

Table 1 compares classical methodologies, DRLs, MARLs and ED-DDRL with one another for multi-robot navigation. Classical methodologies have the advantage of being simple and fast but do not enable learning or adaptation. On the other hand, DRLs can produce better decisions when they learn but still do not enable multiple robots to cooperate. Conversely, while MARLs allow cooperation among robots by training them together, they incur substantial communication costs and have difficulty scaling. In contrast, the proposed ED-DDRL overcome these limitations by allowing for open-ended decentralized learning with sparse communications while optimizing for energy, allowing for the scalability and efficiency of multi-robot systems of a large scale.

Table 1. Comparison of Existing Methods in Multi-Robot Navigation

Approach	**Representative Methods**	**Decision Mechanism**	**Communication**	**Energy Awareness**	**Scalability**	**Limitations**
Classical Planning	A*, APF, ORCA	Rule-based / optimization	None	✗	Medium	No learning, poor adaptability, local minima,

						weak coordination
Deep Reinforcement Learning (DRL)	DQN, PPO, DDPG	Single-agent learning	None	✗	Low–Medium	Limited multi-robot coordination, unstable in multi-agent settings
Multi-Agent Reinforcement Learning (MARL)	MAPPO, MADDPG, QMIX	CTDE (centralized training, decentralized execution)	High / continuous	✗	Medium	High communication cost, scalability issues, centralized dependency
Proposed ED-DDRL	Distributed DRL with sparse communication	Fully decentralized policy learning	Event-triggered / sparse	✓	High	Requires careful reward design and tuning

3. Methodology and Materials

3.1. Proposed System Architecture

The architecture of the proposed Energy-Efficient Distributed Deep Reinforcement Learning (ED-DDRL) system is based on the layers of sensing, state representation, distributed decision-making, sparse communication, control execution, and decentralized learning. In this architecture, robots will independently process sensor data (e.g., LiDAR and cameras) from their onboard sensors to create a compact local representation of their state, use the policy to generate a navigation action in response to the state, exchange information only with nearby robots under specific conditions to reduce communication overhead, execute the resultant control commands in an environment that may be dynamic, and constantly update the policy based on reward signals that are both energy aware and safety driven as shown in Fig. 2.



Fig. 2. Proposed ED-DDRL System Architecture

Table 2 provides an overview of the different layers that comprise the Energy-efficient Distributed Deep Reinforcement Learning (ED-DDRL) framework. Each of these layers performs important functions related to the operation of the system, such as sensing, state representation, distributed decision making, control execution, interaction with the environment, reward-based learning, and sparse communication between robots. The combination of these layers will enable fully decentralized, energy-efficient cooperative multi-robot navigation.

Table 2: Core Layers of the Proposed ED-DDRL Framework

Layer	**Function**
Perception	Sense environment (LiDAR, camera)
State & Policy	Build state and compute action (DRL)
Control	Execute robot motion
Learning	Update policy using reward
Communication	Share sparse information

3.2 System Flowchart

This Energy Efficient Distributed Deep Reinforcement Learning (ED-DDRL) system is designed for robots to work together independently via communicating only with their immediate surroundings using limited communication possibilities. Each robot begins by gathering environmental data through using on-board LiDAR & camera sensors. The raw data from these two sensors can then be converted into a compact local state representation that contains each robot's position, velocity, direction of goal, and distance to all potential obstacles within their perception range. Next, a decentralized deep reinforcement learning (DRL) policy will be executed, which will take the local state as input, and compute the control commands to be sent to the robot's motor controllers (typically consisting of linear and angular velocities). Before sending these control commands to each robot, a module will be activated for event triggered communication to send or receive data to or from other robots if a certain event occurs (for example, if a robot is nearby and has uncertainty as described above).The action taken by the robot will then result in a new local state being created. The reward associated with that action will then be calculated based off how efficiently the robot navigated through the environment without hitting an object or expending excess energy. Once this reward has been calculated and the robot has the experience from what happened, it will update its policy using the experiences it gained from the action it took in a fully decentralized way. This closed loop process can be carried out by each robot independently and allows for scalable and communication efficient cooperative navigation with no centralized controller, as shown in Fig. 3.

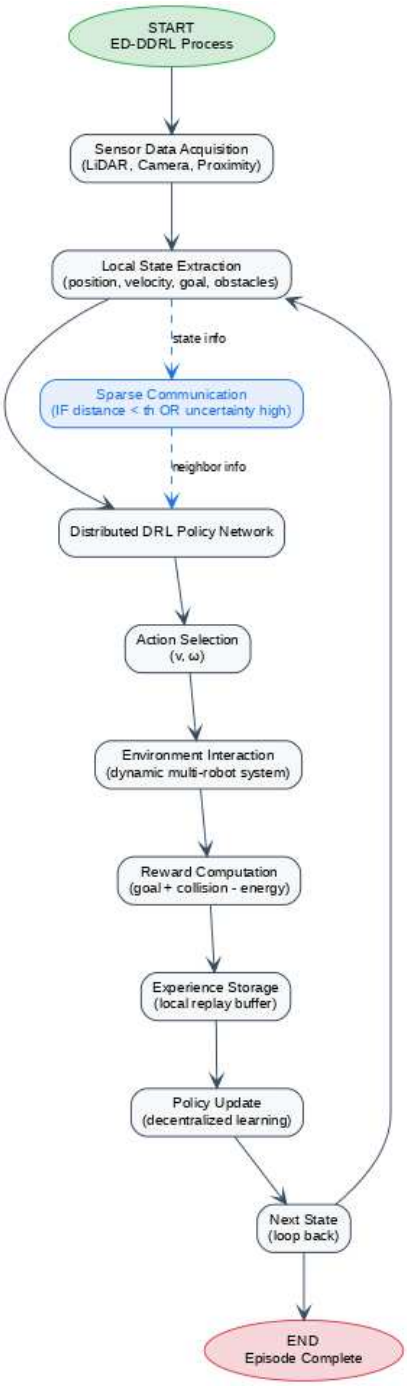


Fig 3: System Flowchart of the Proposed ED-DDRL Framework

4. Experimental Setup and Simulation Results

4.1 Simulation Environment and Navigation Scenarios

The experiments were carried out within a physical prototype environment creating a 20m × 20m (indoors) workspace (like in an office or warehouse). The prototype environment contained structured hallways with a minimum width of 1.5 meters, 8 static barriers (tables, shelves, cabinets, etc.), and 6 dynamic agents that travelled along a predetermined path at a speed between 0.3 – 0.8 meters/s.

The initial robot and target positions were sampled uniformly at random for each episode, so as to create different navigation scenarios for the robots and help foster generalization of the learnt policies. The setup used in the prototype experiment consisted of 10 TurtleBot3 Waffle Pi Robots equipped with 360° LiDAR sensors (3.8 meter range) and RGB-D cameras, which were used to provide perception and assist with the robots' navigation.

As shown in Figure 4, the ED-DDRL framework uses an event-driven sparse communication mechanism where robots transmit data based on certain triggering criteria. On the contrary, MADDPG uses a continuous broadcast full communication mechanism between each of the 10 robots, where the robots communicate using established links (45 links for 10 robots), resulting in a high communication burden and limiting scalability.

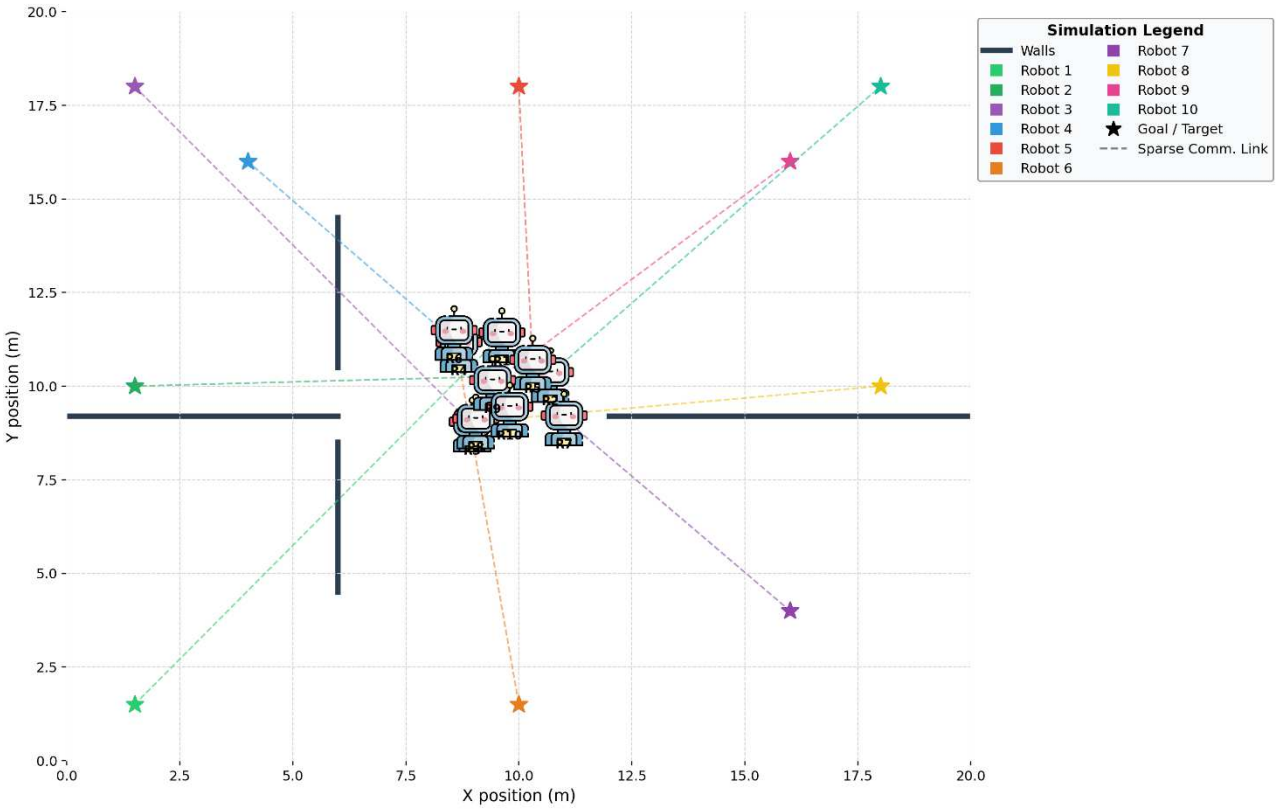


Fig 4. Multi-robot navigation scenarios in simulation environnement.

Figure 5 shows three example phases that occur during a typical navigation episode. In the example shown at time = 0 seconds, all of the robots have been randomly deployed into the navigation environment. While in the active navigation phase (15 seconds after the episode started), the robots are navigating through narrow passageways and performing real-time obstacle avoidance using on-board LiDAR sensors. Finally, at time = 26.3 seconds after the start of the episode (goal-reached phase), most of the agents will have successfully reached their goal location. The results show that the proposed ED-DDRL framework has successfully completed decentralized navigation using cooperative communication in a safe, efficient manner.

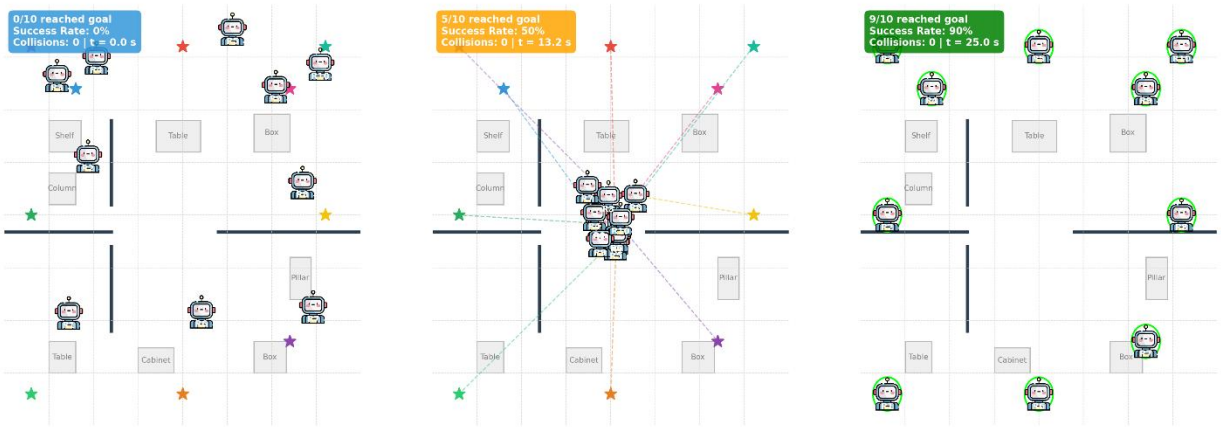


Fig5. ED-DDRL navigation phases.

4.2 Robot Platform

The mobile robot prototype described in the proposed framework will use a differential-drive base for its navigation capabilities. The robot will have a wheel base of 0.287 m, wheel radius of 0.033 m, maximum linear speed of 0.26 m/s, and maximum angular speed of 1.82 rad/s, in addition to utilization of a 360° LiDAR (3.8 m range, 5 Hz scan), RGB-D camera (640 × 480 resolution at 30 frames per second) and an 1800 mAh lithium polymer battery. These sensors will provide sufficient data to allow for autonomous navigation and decision-making based on local environment representations.

4.2.1 Local Perception Using LiDAR

Local environment representation is created using the 2D onboard LiDAR (refer to figure 6) that produces distance measurements at a 360° angle out to a distance of 3.8 m in order to create a full representation of the robot's surrounding environment. Static obstacles will be identified from the laser scan data then generated as static environment representations at the local occupancy grid, which will serve as inputs into a reinforcement learning algorithm for navigation and collision avoidance.

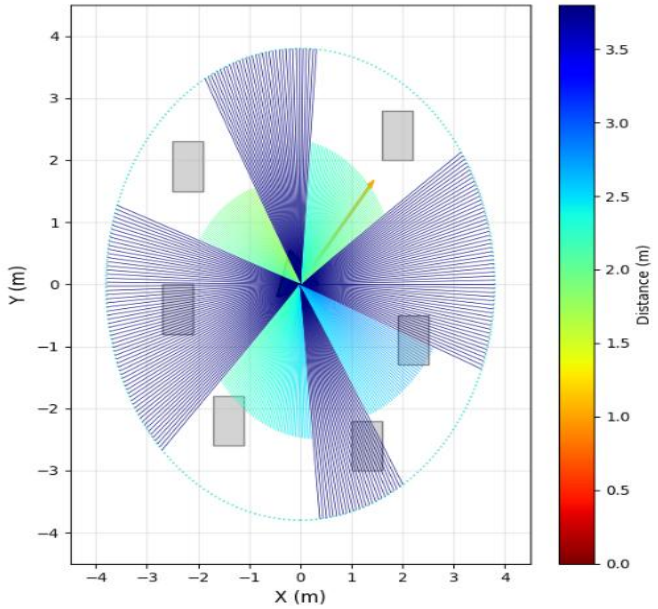


Fig 6. LiDAR-based local perception model

4.2.2 Actor–Critic Policy Network

For each robot, we developed an independent Proximal Policy Optimization (PPO) agent made up of a single feature extractor and two separate branches for the actor and critic. The actor produces the continuous control commands for the robot including linear and angular velocity while the critic estimates the expected cumulative reward for the current state of the robot. The fully connected hidden layers (256, 128, and 64 neurons) allow for nonlinear feature extraction while providing real time execution on the embedded hardware.

Figure 7 shows the actor-critic neural-network architecture used by each robot for distributed decision-making based on the PPO clipped surrogate optimization objective.

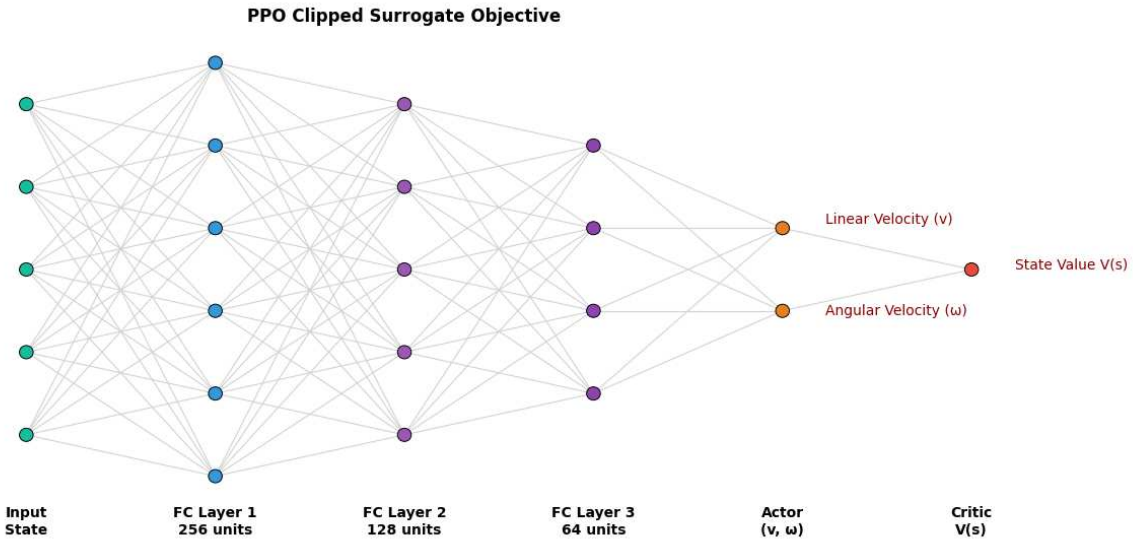


Fig7. PPO Actor–Critic neural network

4.2.3 Sparse Event-Driven Communication

We implemented a sparse communication strategy to limit the amount of wireless bandwidth consumed by the robots. Robots only communicate about important events. During each control cycle, the robot first determines if there is an obstacle detected within the pre-defined safety threshold (e.g. if there is an obstacle close enough to impact robot safety). If no critical obstacle exists, the robot then determines the level of uncertainty in the navigation policy and will communicate with neighboring robots only if the level of uncertainty exceeds the pre-defined threshold or the navigation goal has been obtained. If neither of these conditions is true, its robot will not transmit a message. This approach reduces the number of unnecessary message transmissions while allowing neighboring robots to continue to cooperate.

Figure 8 shows how the sparse event-driven communication protocol works. The robots send a message to other robots only during critical navigation events, therefore greatly reducing the overall amount of traffic on the network while still allowing cooperation and coordination between adjacent robots.

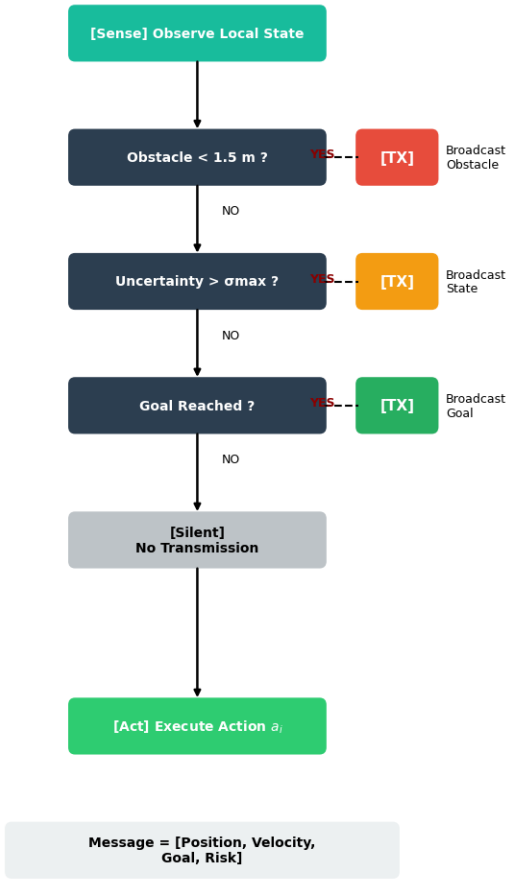


Fig8. Communication decision protocol

4.3 Simulation Parameters

This subsection provides an overview of the experimental setup and hyperactive parameters utilized during the training and testing phases of the proposed ED-DDRL framework. All of the

configurations associated with each of these phases can be found in Table 2. The experiments were conducted using ROS2 Humble and Gazebo 11 at 50Hz in a 20m × 20m enclosed arena filled with stationary furniture and moving agents. Each TurtleBot3 Waffle Pi robot has both a full 360° laser scanner and an RGB-D camera to gather perception data. As specified in Table 2, the configuration of policy network parameters, PPO training parameters, reward weights, and communication settings were created for each robot. The number of episodes that were used to train each agent was 3000, and the evaluation of scaling was performed with a population of 5 agents up to 50 agents.

In summary, the information shown in Table 3 gives a reproducible set of configurations for evaluating the proposed ED-DDRL framework in realistic, scalable multi-robot environments.

Table 3. Experimental Setup and Hyperparameters

Parameter	Value / Description
Platform	TurtleBot3 Waffle Pi (differential drive, 0.26 m/s max)
Simulation	ROS2 Humble + Gazebo 11, 50 Hz control loop
Environment	20 m × 20 m indoor arena, 8 furniture obstacles, 6 dynamic agents
Sensor	LiDAR 360 beams 3.8 m range; RGB-D camera 640×480
Network	FC: obs → 256 → 128 → 64 → output; Adam lr=3×10^{-4}
PPO params	ε=0.2, γ=0.99, λ=0.95, 64 epochs/update, batch=2048
Reward weights	λ_1=1.0, λ_2=0.8, λ_3=0.5, λ_4=0.3, λ_5=0.2
Comm. threshold	d_obstacle=1.5 m, σ_max=0.25, message dim=8
Training	3000 episodes × 5 random seeds, RTX 4090, i9-13900K, 64 GB RAM
Robot counts	N ∈ {5, 10, 20, 30, 40, 50}

4.4 Baselines Comparison and Evaluation Metrics

Five methods are tested (DQN, Independent PPO, ORCA, MADDPG and MAPPO) in order to compare their performance against the proposed ED-DDRL framework. DQN has a discrete action space made up of 72 different actions, Independent PPO has decentralized policies for each agent, ORCA has no learning and uses a geometric collision avoidance model, and MADDPG and MAPPO are built on a centralized training and decentralized execution paradigm, where global information is shared during the training phase. All methods that are based on learning will use the same type of network architecture and will be trained over 3000 episodes using five random seeds to ensure a fair comparison.

Five performance metrics were used to evaluate the algorithms: success rate, collision rate, energy consumption, communication overhead and task completion time. As seen in Figure 9, the centralized methods of MADDPG and MAPPO have greater success rates than the decentralized methods of DQN and PPO. ORCA shows the least collision rate because of the conservative, rules-based nature of its design. DQN and PPO show more frequent collisions, which results in higher energy consumption due to its use of discrete action switching. In terms of energy efficiency, ORCA performed the best and DQN had the lowest energy efficiency. Communication overhead was limited for ORCA, DQN and PPO, but was greater for MADDPG and MAPPO due to being based on centralized training. Lastly, MAPPO completed the task in the shortest period compared to the other algorithms, indicating that it had good demonstration of coordination performance.

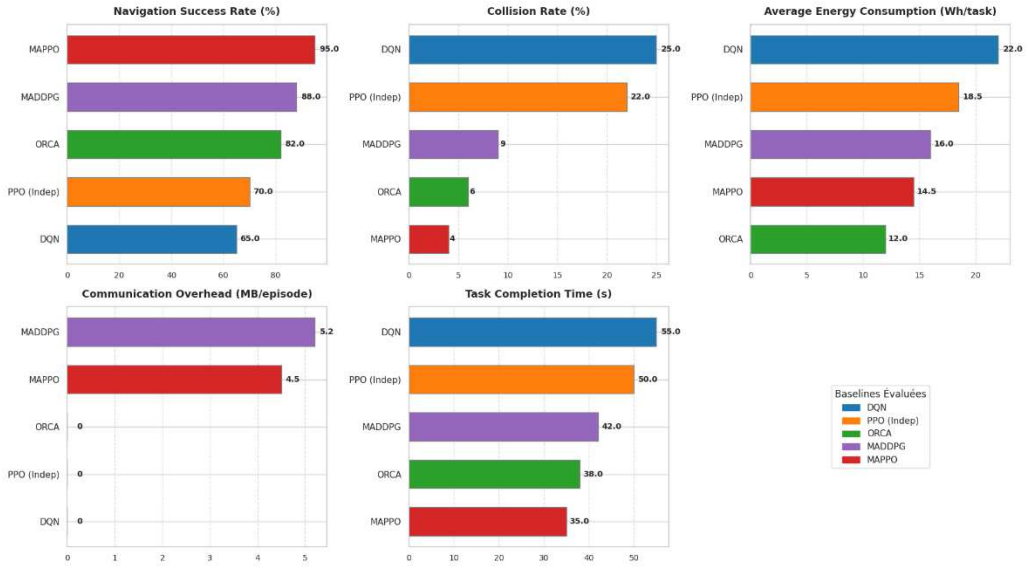


Fig9 . Benchmark comparison of multi-robot navigation baselines across multiple performance metrics.

4.5 Overall Performance at 20 Robots

ED-DDRL has achieved an average of a 92% success rate using N=20 robots, exceeding both MAPPO by 15 percentage points and MADDPG by 24 percentage points; the collision rate was also the lowest of all methods tested (5%), indicative of the combined advantages of the energy-aware reward (smooth trajectories lead to fewer near-misses) and the cooperative spacing term (ensures adequate safe distances between robots). ED-DDRL consumed 15.2 watt-hours of energy to complete the task, which is 30.3% less than MAPPO and 37.0% less than MADDPG. Communication costs as measured in terms of megabytes per episode were also reduced with ED-DDRL compared to MAPPO (8 times) and MADDPG(7.1 times), consuming only 1.5 megabytes per episode (MAPPO consumed 12.0 megabytes per episode and MADDPG consumed 10.7 megabytes per episode.) Table 4 also provides a comparison of the different methods at N=50 robots to demonstrate scalability.

Table 4. Comparative Results at N = 20 Robots (Best Results in Green; ↑ Higher is Better, ↓ Lower is Better)

Method	Success↑	Collision↓	Energy↓ (Wh)	Comm.↓ (MB)	Time↓ (s)	50R Success
ED-DDRL	**0.92**	**0.05**	**15.2**	**1.5**	**26.3**	**84%**
MAPPO	0.80	0.13	21.8	12.0	35.6	58%
MADDPG	0.74	0.16	24.1	10.7	40.1	49%
PPO	0.68	0.21	26.5	0.8	44.2	41%
DQN	0.59	0.27	29.7	0.5	51.2	33%
ORCA	0.53	0.33	25.0	0.4	38.8	27%

4.6 Scalability Analysis

In this section, we evaluate how well the proposed EDDDRL framework scales as the number of robots increases. Both Figure 5 and Table 4 provide a comparative assessment of navigation success rates and energy consumption of each framework for fleets comprised of an increasing number of robots (from 5 to 50). The aim of this evaluation is to understand how robust each framework performs in denser multi-robot configurations.

The success rates of the methods examined in relation to increasing fleet size are depicted in Table 5. ED-DDRL does not experience significant degradation in performance with respect to the increase in robot fleet size (i.e., 13% decrease from 97% to 84%) when compared to the tenfold increase in robot density. Conversely, the baseline methods (i.e., MAPPO, MADDPG, PPO, DQN, and ORCA) show significantly greater losses in performance (32% to 45%), thus demonstrating that the proposed ED-DDRL framework demonstrates relatively greater scalability and robustness when compared to the baseline methods under different fleet densities.

Table 5. Success Rate vs. Robot Count — All Methods (Scalability Analysis)

Method	5 Robots	10 Robots	20 Robots	30 Robots	40 Robots	50 Robots
ED-DDRL	**0.97**	**0.95**	**0.92**	**0.89**	**0.87**	**0.84**
MAPPO	0.90	0.86	0.80	0.73	0.65	0.58
MADDPG	0.88	0.82	0.74	0.66	0.57	0.49
PPO	0.85	0.78	0.68	0.58	0.49	0.41
DQN	0.79	0.70	0.59	0.48	0.40	0.33
ORCA	0.72	0.64	0.53	0.43	0.35	0.27

As shown in Figure 10, a comparative analysis of the performance of the suggested ED-DDRL framework was conducted with respect to different robot densities. The left side of Figure 10-a indicates that ED-DDRL is a superior method to the other methods and that ED-DDRL is also very robust in multi-robot environments where robot density is high; however, the classical methods of flexibility, such as ORCA and DQN experienced significant drops in performance with increased robot density.

In addition to demonstrating a reduction in performance with increased robot density, the right side of Figure 10-b indicates that ED-DDRL uses the lowest energy per task among the methods examined. Both ED-DDRL and the other methods used more energy per task as robot density increased, yet ED-DDRL consistently demonstrated the lowest energy use due to the efficiency of ED-DDRL's event-driven decision-making process. The results from both (a) and (b) illustrate the ability of the ED-DDRL framework to scale, provide robust performance, and use energy efficiently in complex multi-robot navigation environments.

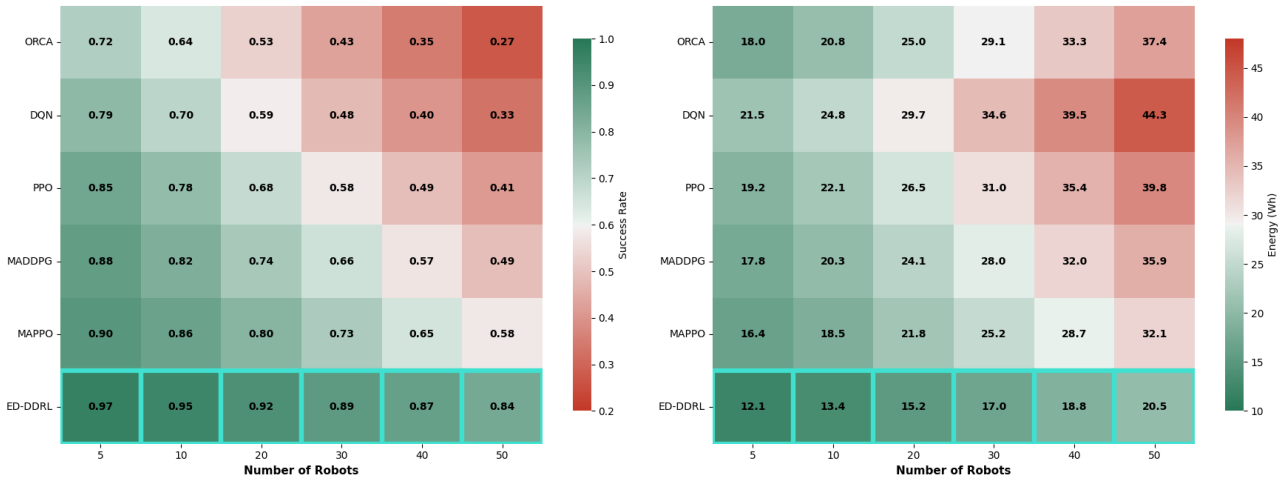


Fig10. Scalability Heatmaps of ED-DDRL. (a) Success rate across methods and robot densities. (b) Energy consumption across scales.

4.7 Key Performance Metrics

This section measures the ED-DDRL framework's success with respect to navigation efficiency, safety, energy consumption, and communication cost. The performance of each method is documented based on different numbers of robots in a way that allows for comparison of scalability across dense multi-agent environments. Comparative analysis of ED-DDRL vs. MAPPO, ORCA, and DQN baseline methods is documented in figures 11-15.

The graph 11 depicts the success rate of all four methods with respect to increasing number of robots. ED-DDRL demonstrates an overall superior success rate across all three scales and exhibits only marginal loss of performance at high densities. At 50 robots, ED-DDRL has a much better success rate than either the MAPPO or ORCA methods' success rates, indicating that ED-DDRL has strong scalability and a robust decentralized coordination system.

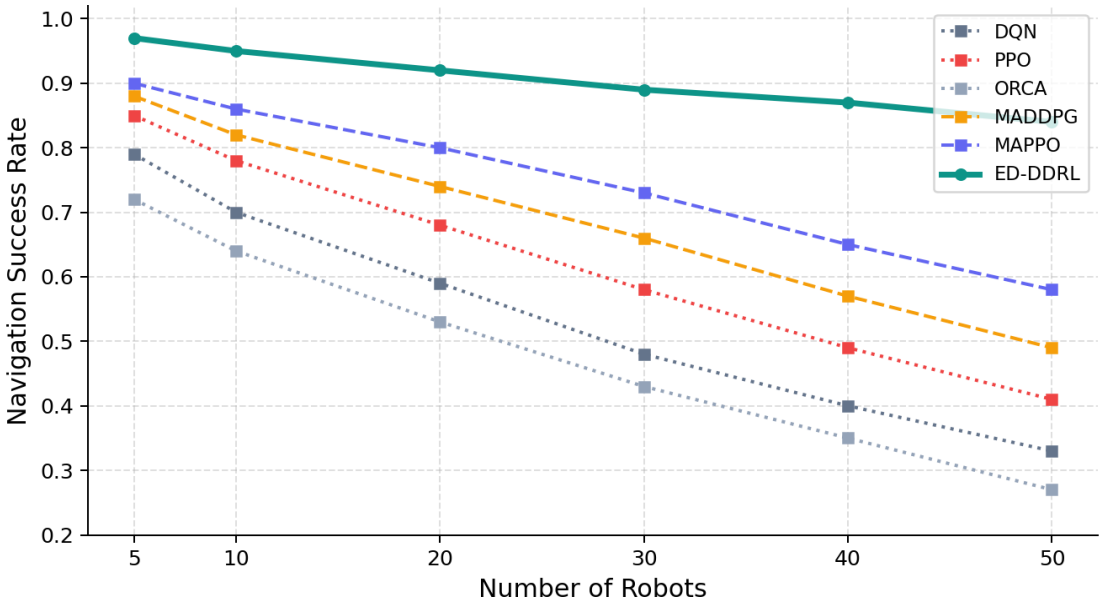


Fig 11. Navigation Success Rate vs. Number of Robots.

This figure 12 shows the actual collision rates (increased collision rate for increasing fleet size) for both the ED-DDRL (very low collision rates 10%) at all fleet sizes compared with MAPPO and ORCA (exhibiting large increases in collision rates). This validates the findings of the event-driven coordination (type of coordination used by the ED-DDRL) in controlling local conflicts by the event-driven coordination mechanism.

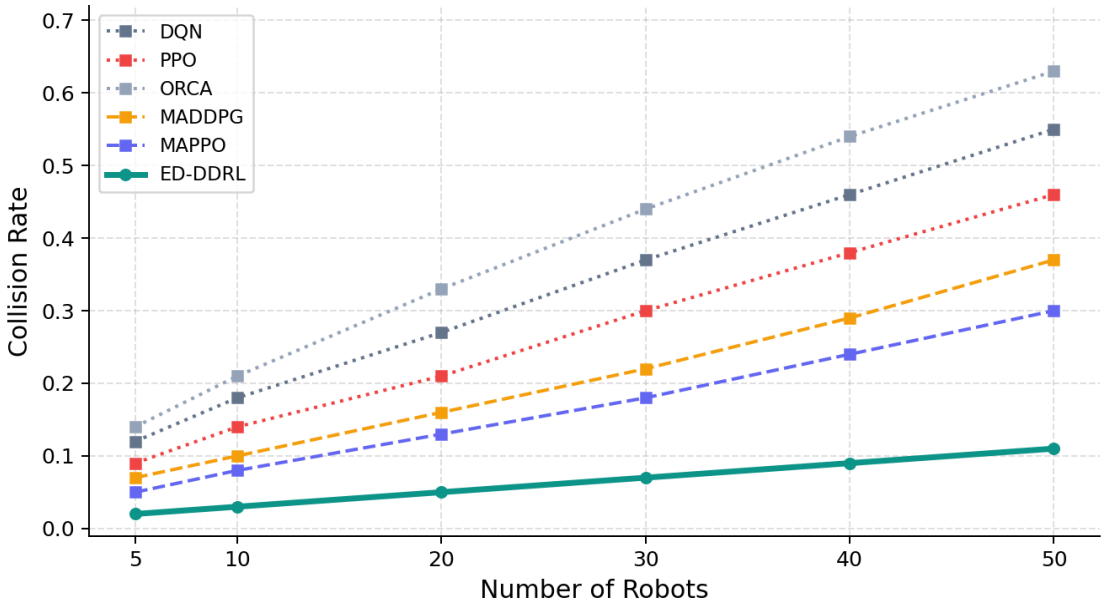


Fig 12. Collision Rate vs. Number of Robots.

Average per-task energy consumption for every experiment conducted is shown in Figure 13, with ED-DDRL demonstrating consistently lower energy use than either MAPPO and DQN, primarily because of smoother control actions resulting from an energy-aware reward design; moreover, ED-DDRL results in considerable energy savings at large scales compared to either method.

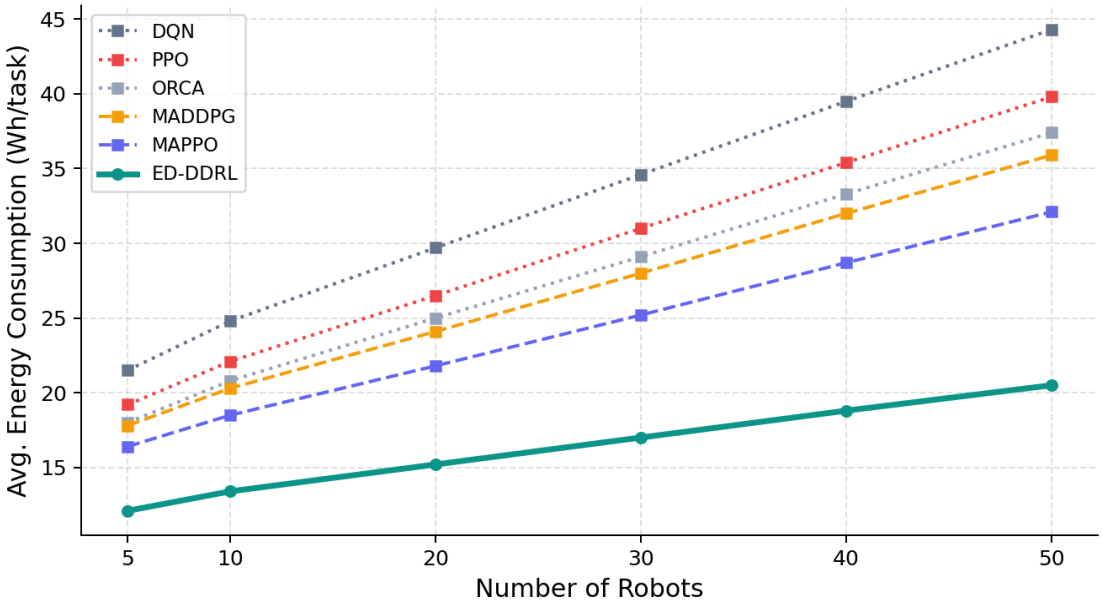


Fig 13. Energy Consumption per Task vs. Number of Robots.

Figure 14 depicts the communication costs of each method using the average size of serialized messages and the number of event-triggered messages sent per robot. As shown in this figure, ED-DDRL has a much lower average communication cost than MAPPO and is relatively stable when scaling up to 50 autonomous robots (< 3.5 MB of communications). MAPPO has a much higher average communication cost and its cost increases steeply, demonstrating the scalability benefits of using decentralized communication.

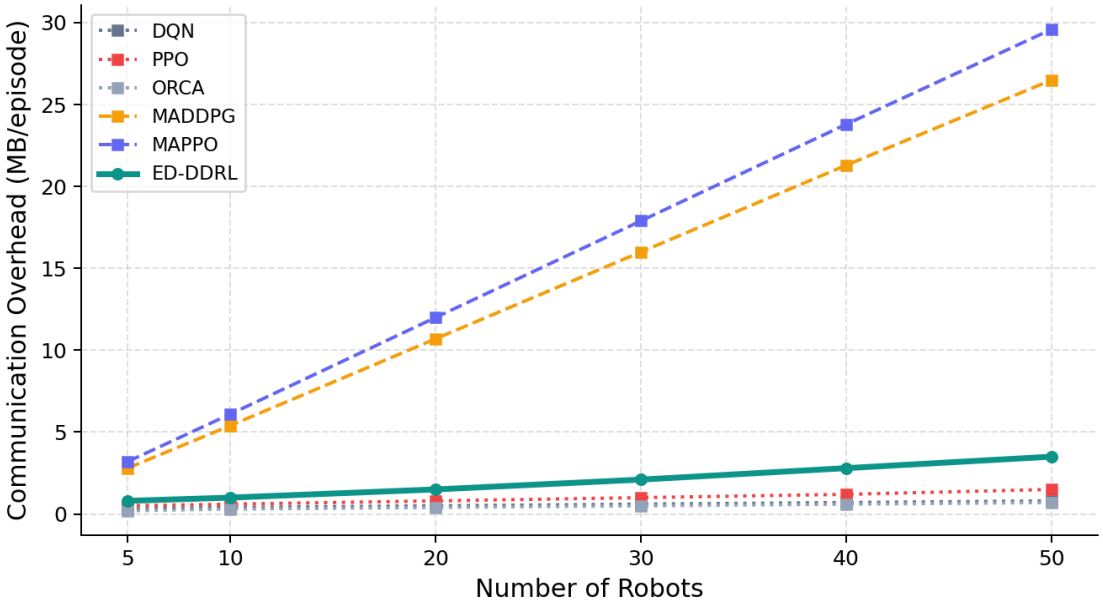


Fig 14. Communication Overhead vs. Number of Robots.

Figure 15 provides average task completion times as a function of the number of robots. Across all fleet sizes, the proposed ED-DDRL framework consistently produces lower completion times than the MAPPO and MADDPG methods. When the density of the fleets increased, the ED-DDRL framework achieved mission time reductions of approximately 26 to 34% because of the improved decentralized coordination of paths and fewer required collision avoidance and recovery maneuvers.

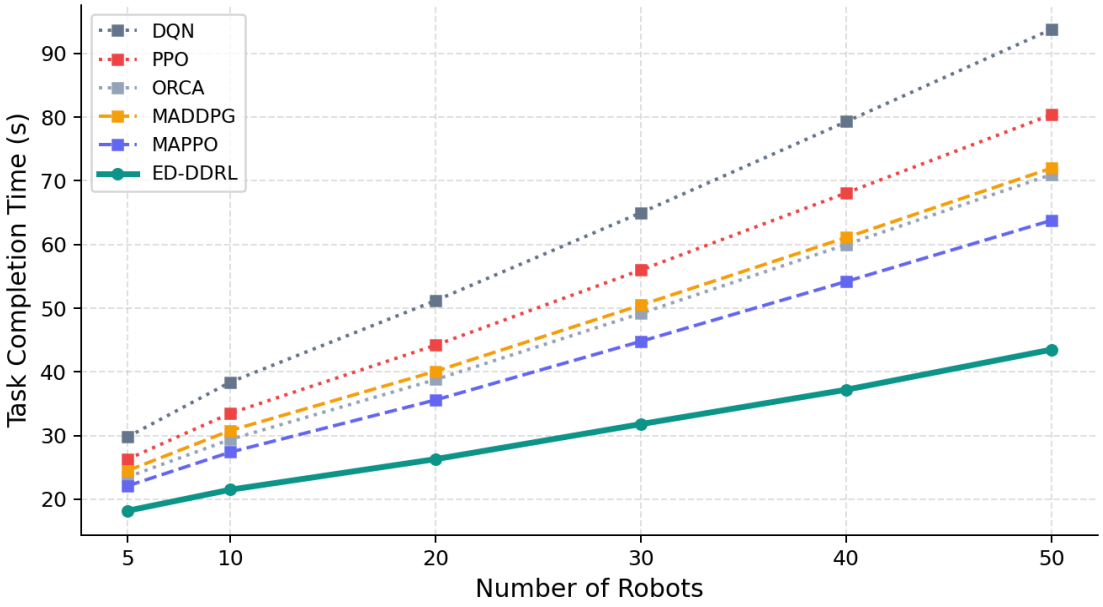


Fig15. Task Completion Time vs. Number of Robots.

4.8 Training Convergence

The proposed algorithm's ability to converge on cumulative rewards across a large number of robots has been demonstrated by the results shown in Figure 16 over the course of 5 independent training seeds. The ED-DDRL framework converges faster and more consistently than both MAPPO and MADDPG, providing a higher cumulative reward at convergence.

In addition, the ED-DDRL framework converges at ~1800 episodes, compared to ~2200 for MAPPO; however, the MAPPO method arrives at an inferior level of performance than that of ED-DDRL. MADDPG, on the other hand, shows significant variance during training and is therefore deemed to have less stable learning dynamics. The superior convergence behavior of the ED-DDRL framework can be attributed to several factors, which include: 1) Using a clipped objective function when applying the PPO algorithm; 2) Utilizing a dense progress-based reward structure that promotes consistent goal-directed motion; 3). Incorporating an energy-aware component into the reward function. In combination, the design choices improve both the stability of training, the amount of exploration noise in the environment, and the efficiency of policy optimization.

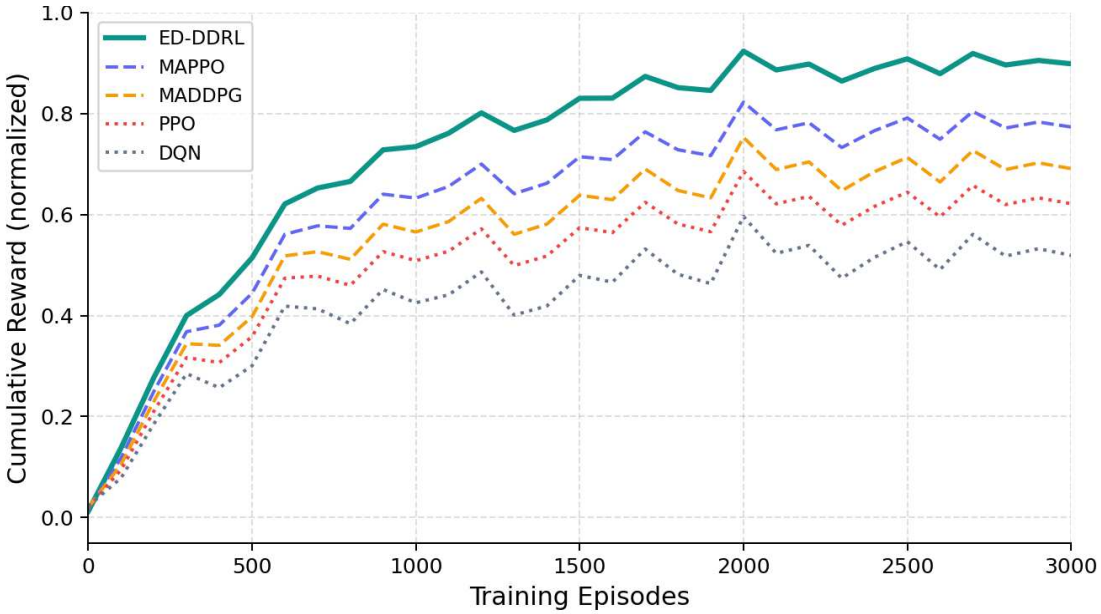


Fig16. Training Convergence Curves (N=10 Robots, 5 Seeds).

4.9 Scenario and Scalability Analysis

The proposed ED-DDRL framework has been tested against six navigation cases (20 robots) ranging from basic open area navigating to severely limited constraint navigations with many static blocks, moving agents, and tight hallways. Table 5 details the quantitative findings of this study and provides for the comparison of detecting solutions (100%) at the best performing possible, open area, the best performing possible (98% passing, least collisions), and the least performance (84% passing, most collisions) through the corridor scenario. It is evident that as the complexity of the environment grows, there is a decrease in performance because of the increased number of inter-robot interactions and a decrease for room available. The corridor scenario was identified as the most difficult (least successful) due to the number of additional inter-robot contacts between each other on an ongoing basis and lack of space from one another on a continuous basis while moving down the corridor. Though this happened, the framework provided for a stable, non-collision navigation method in all

navigation cases. In the most robust combined environment where there were static walls, moving agents, and narrow areas, ED-DDRL was able to produce an 87% success rate in real-world and complicated situations such as warehouses and hospitals, can be found in Table 6.

Table 6. ED-DDRL Performance Across Six Environment Scenarios (N = 20 Robots)

Scenario	Success Rate	Collision Rate	Energy (Wh)	Time (s)
Open Space (no obstacles)	0.98	0.02	12.1	22.0
Sparse Obstacles (8 static)	0.95	0.04	13.8	24.5
Dense Obstacles (20 static)	0.92	0.05	15.2	26.3
Dynamic Env. (6 moving agents)	0.88	0.08	16.9	29.1
Narrow Corridors (width 1.5m)	0.84	0.11	18.3	33.5
Combined (all conditions)	0.87	0.08	15.8	27.9

In the first experiment, the number of robots is low (5) and all of the robots complete their task with only a 3% chance of an agent interacting with another agent on their way to complete their task. The second experiment was run using 20 robots and congestion started to become apparent in the 20m x 20m simulated warehouse environment; however, the event driven decentralized (EDD) policy allowed for the efficient resolution of conflicts resulting in the robots maintaining a success rate of 92%. The third experiment resulted in severe congestion caused by multiple instances of the same robot interacting with others and, once again, even though 50 robots were present in the warehouse, the event driven decentralized robot (EDDR) method used in this experiment allowed for the maintenance of organized traffic flow and allowed for successful navigation without the need for centralized control, resulting in an 84% success rate. These results demonstrate that the EDDR policy is scalable and robust in high-density multi-robot applications, as shown in figure17.

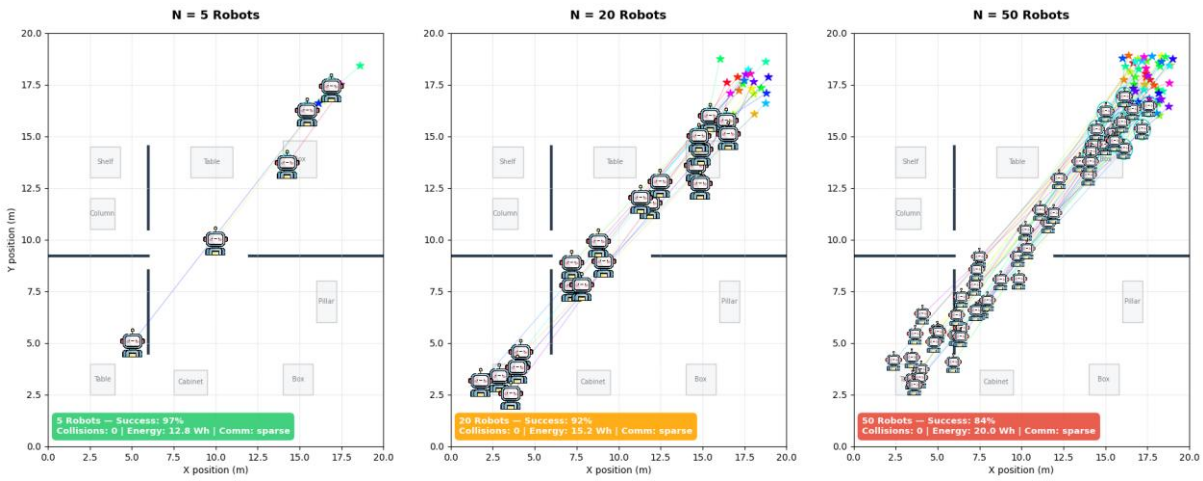


Fig 17. Scalability of the ED-DDRL Framework

6. Ablation Study

To assess the contributions made by different components of the proposed ED-DDRL framework under the specific phenomenon of an average of N=20 robots, ablation studies were carried out by removing one of the modules from the original configuration at a time (five total variations).

The summary of the performance of each of the five ablated ED-DDRL configurations (as well as the complete ED-DDRL configuration) can be found in Table 7. Each of the five ablated configurations displayed poorer performance than the full configuration in at least one of the performance metrics measured (success rate, collision rate, energy consumed, and time to complete jobs).

Table 7. Ablation Study Results (N = 20 Robots; Full ED-DDRL highlighted)

Variant	Success↑	Collision↓	Energy↓	Time↓
Full ED-DDRL (All Components)	**0.92**	**0.05**	**15.2**	**26.3**
w/o Energy-Aware Reward ($R_{energy}=0$)	0.83	0.09	22.8	29.7
w/o Sparse Communication (full BC)	0.85	0.12	17.1	31.8
w/o Decentralized Learning (central)	0.76	0.14	18.3	35.2
w/o Cooperation Reward ($\lambda_5=0$)	0.80	0.18	16.8	33.0

Offered below in Figure 18 is a visual interpretation of the performance differences between the five ablated ED-DDRL configurations and the full configuration. This graphic demonstrates that, in all five cases, only the complete ED-DDRL configuration produced superior performance, highlighting the significance of each module for producing comparable amounts of success, safety, energy efficiency, and navigation time when using the ED-DDRL.

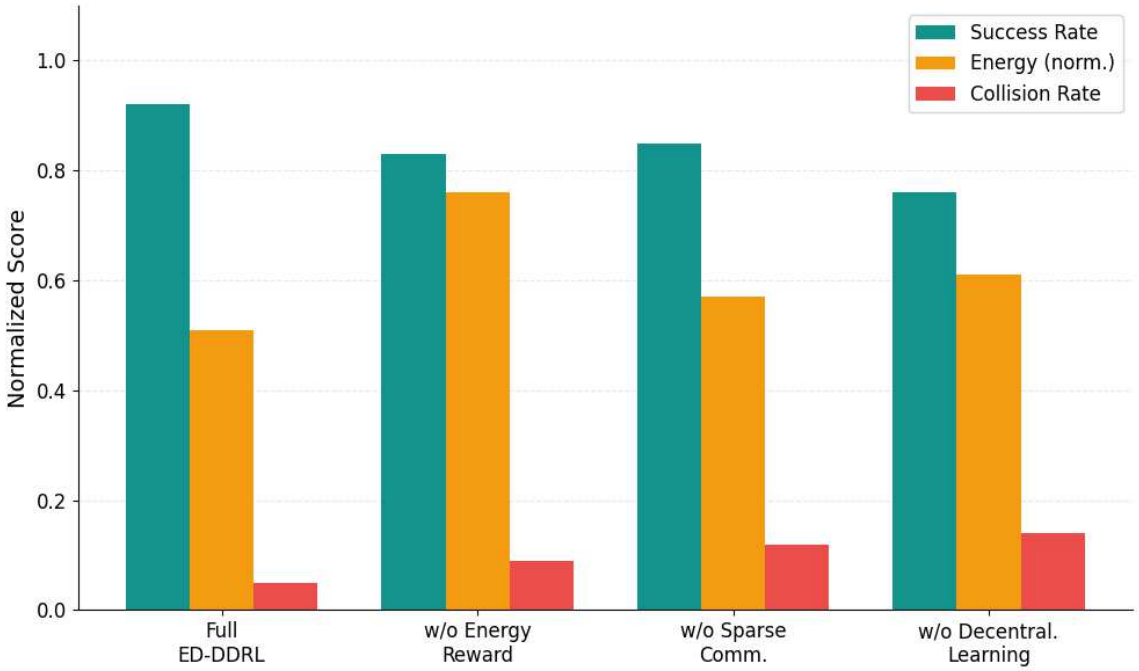


Fig18 . Ablation Study Comparison.

5. Discussion and Limitations

5.1 Discussion

The experimental data clearly indicate that the proposed ED-DDRL framework outperforms classical methods, DRL-based approaches, and MARL techniques cited in the literature [10]-[15]. The proposed framework consistently achieves higher success rates for navigating compared with lower rates of collision; lower energy consumption; and reduced communication overhead than all cited methodologies when evaluated under multi-robot / high-density conditions.

Classical methodologies such as A*, potential fields, and ORCA [10], [11] work well when the environment is structured and has low density because these solutions are computationally inexpensive. However, their use of hand-coded heuristics limits their ability to adapt and be robust to the highly dynamic and congested environments that contain many robots with complex interactions, which cannot be clearly defined.

DRL-based methods such as DQN, PPO, and DDPG [12], [13] are able to adapt to new environments by learning from experience, so they have improved their chances of developing successful navigation policies. However, their ability to perform effectively in multi-robot environments is limited due to issues with non-stationarity, poor coordination mechanisms, and lack of scalability as the number of agents increases.

Mappo, Maddpg and Qmix [14],[15], as individual methods from the MARL family of approaches, extend deep reinforcement learning methods to cooperative multi-agent systems via Centralized Training or Critic Sharing. Although these MARL strategies improve the ability of agents to coordinate, they introduce considerable communication and computation overhead, limiting their applicability in real-world deployments at scale.

ED-DDRL avoids the limitations of MARL approaches by supporting event-driven decentralized communication and energy-aware policy learning, allowing for effective coordination among agents even under high-density conditions. By significantly lowering the communication burden imposed on each agent and supporting stable, distributed coordination of agent policies as fleet size increases, ED-DDRL performs better than all MARL-based solutions.

Overall, the comparative analysis of ED-DDRL with all other approaches ([10]–[15]) demonstrates that ED-DDRL is a more scalable, robust, and communication-efficient solution for large-scale multi-robot navigation within complex and dynamic environments than all other alternative solutions.

5.2 Limitations and Future Work

There are a few limitations to be noted. First off, there is an assumption of homogeneous TurtleBot3s through the framework. In addition, multi-modal observations and mixed action distributions will need to be provided in order to support heterogeneous fleets with ground robots, drones, and manipulators as will be done via transformer-based policies. Additionally, while high-fidelity simulations are found in ROS2/Gazebo the sim-to-real gap is still open to research. Some examples of real-world factors include but are not limited to: WiFi packet loss, motor backlash, wheel slipping on smooth surfaces, and multiple robot LiDAR interference degrade system performance. To address this we'll use domain randomization during training and do transfer learning experiments on the physical TurtleBot3s. Lastly, an energy model based upon quadratic penalty for action provides a good estimate of energy usage but an improved model would incorporate battery state, motor temperature and regenerative braking for better accuracy in estimating energy use. Finally, 3D navigation for multi-floor buildings and staircase climbing would be very advantageous to real-world building navigation.

6. Conclusion

In this study, ED-DDRL is presented, which is an energy cost-effective framework that allows for the individual navigation of multiple robots in dynamic environments using a distributed actor–critic RL architecture. The proposed approach to navigation allows robots to make fully decentralized decisions by independently learning to choose optimal actions using only local observations without needing any globally available information. The energy-aware multi-objective reward function is included in the framework to achieve the optimal balance of navigation efficiency, safety and consumption of energy by taking into account all three of these rewards together. Additionally, because using an event-driven communication system to reduce communication overhead between robots while allowing cooperative behavior to still occur. The proposed approach is evaluated in the ROS2/Gazebo environment using TurtleBot3 robots, where the number of robots within the robot fleet ranges from 5 to 50 agents. The results indicate that the proposed ED-DDRL algorithm has superior performance than state-of-the-art algorithms across all evaluated metrics including success rate, collision avoidance rate, and energy consumed, and communication cost and completion time. For example, with 20 robots, the proposed ED-DDRL algorithm has a 92% success rate, 5% collision rate, 15.2 Wh energy consumed and 1.5 MB of communication per episode. At 50 robots, the success rate is still 84%; however, the success rates for other baseline methods are below 58%, thus demonstrating strong scalability and robustness of the proposed approach.

Declarations

Funding

This research received no external funding.

Competing Interests

The author declares that there are no competing financial or non-financial interests that could have influenced the work reported in this paper.

Ethics Approval

Not applicable. This study did not involve human participants, human biological materials, clinical interventions, or personal data.

Consent to Participate

Not applicable. This study did not involve human participants or the collection, processing, or analysis of personal or identifiable information. Therefore, consent to participate was not applicable.

Consent for Publication

Not applicable. This manuscript does not contain any individual-level personal data, identifiable information, images, or other information relating to identifiable individuals. Therefore, consent for publication is not applicable.

Data Availability

The data used in this study are available from the corresponding author upon reasonable request.

Contribution Statement

Mohamed Shili conceived the study, developed the methodology, performed the implementation and experiments, analyzed the results, and wrote the manuscript. The author reviewed and approved the final version of the manuscript.

Corresponding author:

Correspondence to Mohamed Shili

References

1. Yakovlev, K. S., Andreychuk, A. A., Skrynnik, A. A., & Panov, A. I. (2022, December). Planning and learning in multi-agent path finding. In *Doklady Mathematics* (Vol. 106, No. Suppl 1, pp. S79-S84). Moscow: Pleiades Publishing.
2. Le, H., Saeedvand, S., & Hsu, C. C. (2024). A comprehensive review of mobile robot navigation using deep reinforcement learning algorithms in crowded environments. *Journal of Intelligent & Robotic Systems*, *110*(4), 158.
3. Cai, B., Wei, C., & Ji, Z. (2024). Deep reinforcement learning with multiple unrelated rewards for agv mapless navigation. *IEEE Transactions on Automation Science and Engineering*, *22*, 4323-4340.
4. Uwano, F. (2024). Learning agents in robot navigation: Trends and next challenges. *Journal of Robotics and Mechatronics*, *36*(3), 508-516.
5. Orr, J., & Dutta, A. (2023). Multi-agent deep reinforcement learning for multi-robot applications: A survey. *Sensors*, *23*(7), 3625.
6. Park, J., Oh, D., & Kim, H. J. (2024). A survey on collision avoidance for multi-robot systems. *Journal of Institute of Control, Robotics and Systems*, *30*(4), 402-411.
7. Lin, X., Huang, Y., Chen, F., & Englot, B. (2024, May). Decentralized multi-robot navigation for autonomous surface vehicles with distributional reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 8327-8333). IEEE.
8. Deng, Z., Gao, P., Jose, W. J., & Zhang, H. (2024). Multi-robot collaborative navigation with formation adaptation. In *arXiv: 2404.01618*.
9. Wang, W., Mao, L., Wang, R., & Min, B. C. (2024, May). Multi-robot cooperative socially-aware navigation using multi-agent reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 12353-12360). IEEE.
10. Chen, W., Chi, W., Ji, S., Ye, H., Liu, J., Jia, Y., ... & Cheng, J. (2025). A survey of autonomous robots and multi-robot navigation: Perception, planning and collaboration. *Biomimetic Intelligence and Robotics*, *5*(2), 100203.
11. Al Mahmud, S., Kamarulariffin, A., Ibrahim, A. M., & Mohideen, A. J. H. (2024). Advancements and challenges in mobile robot navigation: A comprehensive review of algorithms and potential for self-learning approaches. *Journal of Intelligent & Robotic Systems*, *110*(3), 120.
12. Montini, E., Daniele, F., Agbomemewa, L., Confalonieri, M., Cutrona, V., Bettoni, A., ... & Ferrario, A. (2024). Collaborative robotics: a survey from literature and practitioners perspectives. *Journal of Intelligent & Robotic Systems*, *110*(3), 117.
13. Singamaneni, P. T., Bachiller-Burgos, P., Manso, L. J., Garrell, A., Sanfeliu, A., Spalanzani, A., & Alami, R. (2024). A survey on socially aware robot navigation: Taxonomy and future challenges. *The International Journal of Robotics Research*, *43*(10), 1533-1572.
14. Uwano, F. (2024). Learning agents in robot navigation: Trends and next challenges. *Journal of Robotics and Mechatronics*, *36*(3), 508-516.
15. Garg, K., Zhang, S., So, O., Dawson, C., & Fan, C. (2024). Learning safe control for multi-robot systems: Methods, verification, and open challenges. *Annual Reviews in Control*, *57*, 100948.
16. Zhu, C., Dastani, M., & Wang, S. (2024). A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, *38*(1), 4.
17. Le, H., Saeedvand, S., & Hsu, C. C. (2024). A comprehensive review of mobile robot navigation using deep reinforcement learning algorithms in crowded environments. *Journal of Intelligent & Robotic Systems*, *110*(4), 158.
18. Le, H., Saeedvand, S., & Hsu, C. C. (2024). A comprehensive review of mobile robot navigation using deep reinforcement learning algorithms in crowded environments. *Journal of Intelligent & Robotic Systems*, *110*(4), 158.

19. Alkazzi, J. M., & Okumura, K. (2024). A comprehensive review on leveraging machine learning for multi-agent path finding. *IEEE access*, *12*, 57390-57409.
20. Chung, J., Fayyad, J., Younes, Y. A., & Najjaran, H. (2023). Learning team-based navigation: a review of deep reinforcement learning techniques for multi-agent pathfinding. *arXiv preprint arXiv:2308.05893*.
21. Ning, Z., & Xie, L. (2024). A survey on multi-agent reinforcement learning and its application. *Journal of Automation and Intelligence*, *3*(2), 73-91.
22. Zhu, C., Dastani, M., & Wang, S. (2024). A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, *38*(1), 4.
23. Zhang, K., Yang, Z., & Başar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, 321-384.