Towards Teammate-Aware Active Search for Human-Multi-Robot Teams in Adverse Environments

Sherif Bakr[1], Brian Reily[2] and Christopher Reardon[1]

***Abstract*— In the context of challenging and dangerous environments, robot teams are increasingly employed in various applications such as search-and-rescue missions in unstable, post-disaster areas, mine rescue operations, and military patrols in contested zones. This paper addresses the challenge of teammate-aware active search, focusing on the robots' ability to locate targets of interest and maintain communication with teammates to ensure safe operation under adversity. Our approach leverages multi-agent reinforcement learning techniques to enable robots to robustly search for targets of interest using multi-sensory information while maintaining communication with at least one other teammate. The robots utilize a prior map indicating probability distributions of potential targets in the environment, enhancing their search efficiency. Human operators are integrated as part of the agent team. Humans can provide real-time input and feedback to adjust the robots' strategies based on their observations and capabilities that robots do not possess. This collaboration allows for an exchange of information between the robots and the human member, utilizing both the speed of robots and the understanding of human members. This synergy between the high robotic precision and speed and the human intuition creates a robust framework for teammate-aware active search operations. By incorporating the human component into the loop, this approach insures that the human perspective remains central and critical to the mission. The interactive AI system prioritizes human situational awareness, allowing operators to make adjustments in real-time. Through this integration, we aim to create a balance between the strengths of both, humans and robots, ensuring successful outcomes in adverse and complex conditions.**

I. Introduction

The objective of this research is to address *teammate-aware active search*, where robot teams are tasked with detecting and analyzing targets of interest in the environment while maintaining robust communication with their teammates. The robots should collaborate with a human team member to keep them informed of any changes in the environment. The robots are equipped with state-of-the-art navigation and maneuver capabilities, allowing them to execute search missions effectively. However, the challenge lies in balancing the need to maintain contact with teammates and the human team member, and the necessity to exploit a deeper contextual understanding of targets of interest, a skill that human operators possess. Not only do robots need to search for changes while maintaining communication, but they also have to dynamically learn new policies as the probability distributions of where the targets occur dynamically shift as more areas are explored.

Fig. 1. A multi-robot team is deployed in a simulated environment to search for targets of interest (represented by white points). Effective teammate-aware active search will enable the robot team to identify targets while maintaining communication with teammates.

Robots are deployed based on their sensor and communication capabilities, navigating according to a shared policy that includes objectives for environmental exploration. Each robot must adjust its route and actions to ensure continuous communication with at least one other teammate, maintaining a robust network under varying environmental conditions. The collected data is processed through a centralized system for comprehensive analysis of environmental changes. We utilize a centralized Q-function, enabling coordinated decision-making based on the state and actions of all robots. The decision-making process is formulated as an optimization problem.

Figure 1 depicts multiple robots in a simulated Unity environment preparing to navigate to their goal positions to explore those areas. The simulated environment mimics real-world conditions, with varied terrains and obstacles. The white dots represent the locations known to the robots through a prior probability map, which indicates the areas with the highest likelihood of targets of interest and is updated dynamically as the robots explore more areas of the environment. The robots' movement and decisions are planned by a shared policy that ensures efficient coverage of the area while maintaining communication with at least one other robot (i.e., no robot is isolated from the rest of the team). Similarly, communication must be maintained with a human teammate, for guidance and decision making input.

A multi-objective multi-agent decision framework for multi-objective stochastic games (MOSGs) [1] addresses various algorithms for cooperative multi-objective multi-agent systems, such as local search and reinforcement learning ap-

*This work was supported by ARO W911NF-23-2-0005.

[1]University of Denver, Denver, CO, USA. {sherif.bakr, christopher.reardon}@du.edu

[2]DEVCOM Army Research Laboratory, Adelphi, MD, USA brian.j.reily.civ@army.mil

proaches. While [1] provides a comprehensive framework for multi-objective multi-agent decision-making and emphasizes the importance of cooperative game theory concepts, our approach addresses the challenge of dynamically learning new policies and adapting in changing, adverse environments by maintaining communication at all times.

An approach to multi-agent reinforcement learning by proposing a method to approximate hierarchical belief structures using recursive deep generative models to form and update beliefs about the knowledge and policies of other agents. [2] addresses the complexities of environments that are partially observable and dynamic due to the nature of interactions and learning processes of multiple agents.

Our research extends these methods by applying a similar recursive belief models to a practical scenario to solve the problem of teammate-aware active search in adverse environments by enabling autonomous robots to dynamically adapt their search strategies (using multi-sensory input) to evolving environmental conditions. We integrate the belief models into a centralized-decision making framework allowing real-time coordination and communication among agents.

A probabilistic recursive reasoning for multi-agent reinforcement learning framework, where agents consider how their opponents might react to their future actions is presented in [3]. Bayesian methods are used to calculate conditional policies of opponents, leading to the development of decentralized-training-decentralized-execution algorithms, demonstrating the value of recursive reasoning in achieving convergence where conventional gradient-based methods fail. Our work leverages the concept of recursive reasoning and multi-agent reinforcement learning to enable agents to adapt to the actions of others in a shared environment. We apply these principles to teammate-aware active search in partially observable environments and integrate belief models into a centralized decision-making framework, ensuring coordination and communication at all times, under varying environmental conditions. Our approach involves updating strategies as the agents' knowledge about the environment increases, demonstrating the feasibility of implementing the recursive belief models in more complex applications.

In assessing the current status, challenges, and future directions of multi-robot systems in search and rescue (SAR) missions, [4] motivates the importance and potential of multi-robot systems to enhance disaster response through improved coverage, redundancy, and task decomposition. Key challenges include: decentralized coordination, translating the results of learning from simulation to real-world scenarios, realistic evaluations to include issues like scalability, and algorithmic bias. Our work builds on these foundations by addressing the challenges of maintaining robust communication and coordination among robot teams in dynamic and contested conditions.

Additional research [5] has explored the application of deep reinforcement learning in the context of decentralized cooperative control for multi-agent robotic systems in continuous action spaces, with partial observability. This aligns with our work in the domain of coordination among multiple autonomous agents using reinforcement learning. We also seek to address the challenges of maintaining communication and coordination in partially observable environments.

II. Approach

In our MARL framework, each robot operates autonomously with the ability to learn and adapt. The agents navigate according to a shared policy designed to maximize environmental scanning and coverage. The framework leverages a centralized Q-function to coordinate decision-making among the agents, ensuring that actions are taken with a holistic view of the system's state and objectives. Inspired by [2], we model the problem as a Partially Observable Markov Decision Process (POMDP) for a team of robots. Each robot operates in a partially observable environment.

We derive and present the following equations to provide details about the state, action, history, policy, transition, and observation models used in our approach:

State and Observation:

- $s_t \in S$: The state of the environment at time t.
- $o_t^i \in O$: The observation made by robot i at time t.

Action:

- $a_t^i \in A$: The action taken by robot i at time t.

History:

- $h_t^i = (o_1^i, o_2^i, \ldots, o_{t-1}^i, o_t^i)$: The history of observations and actions for robot i up to time t.

Policy:

- $\pi^i(a_t^i|h_t^i)$: The policy of robot i.

Transition and Observation Models:

- $p(s_{t+1}|s_t, a_t^1, a_t^2, \ldots, a_t^n)$: The state transition model, dependent on the actions of all robots.
- $p(o_t^i|s_t, a_t^i)$: The observation model for robot i.

Objective Function: The objective is to maximize the expected cumulative reward while maintaining communication and efficient exploration.

$$\max_{\pi} \left\{ \mathbb{E} \left[\sum_{l=0}^{\infty} \gamma^l \left(R(s_t, a_t, o_t, s_{t+1}) - \sum_i C^i(p_{ik}, p_{jl}, a_t, q_r) \right) \middle| \pi \right] \right\} \tag{1}$$

Where:

- γ: Discount factor
- $R(s_t, a_t, o_t, s_{t+1})$: Reward function
- $C^i(p_{ik}, p_{jl}, a_t, q_r)$: Cost function

Reward Function: The reward function incorporates the following:

1) **Change Detection:**

$$R_{cd} = \sum_{i=1}^{n} I(\text{change detected by robot } i) \tag{2}$$

Where I is an indicator function that is 1 if a change is detected and 0 otherwise.

2) **Communication Maintenance:**

$$R_{cm} = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n-1} I(\text{robot } i \text{ communicating with robot } j) \tag{3}$$

3) **Exploration Efficiency:**

$$R_{ee} = \sum_{i=1}^{n} \frac{\text{area explored by robot } i}{\text{total area}} \tag{4}$$

The total reward is:

$$R(s_t, a_t, o_t, s_{t+1}) = w_{cd} R_{cd} + w_{cm} R_{cm} + w_{ee} R_{ee} \tag{5}$$

Where w_{cd}, w_{cm}, w_{ee} are weights for each component.

Cost Function:

The cost function is represented by $C^i(p_{ik}, p_{jl}, a_t, q_r)$ which is the cost for robot i to move from cell k to cell l in the environment. The overall cost function is the sum of all individual cost functions.

The integration of recursive belief models and Bayesian updates for belief states into reinforcement learning is achieved, as in [2], by representing the belief states as neural codes, learning hierarchical belief structures, and updating these beliefs using Bayesian inference and reinforcement learning techniques. Each agent maintains a belief state that not only reflects its understanding of the environment but also its beliefs about other agents' beliefs. Each agent i maintains a belief state of the environment and a history of observations and actions of all other agents. Agents form higher-order beliefs recursively. This is done through a recurrent neural network (RNN) to encode the history into a neural belief state. The transition probabilities are updated using Bayesian inference, reflecting new observations.

a) Zeroth-Order Belief::

$$B_t^0(i) = p(S_t | H_t(i))$$

where S_t is the state of the environment and $H_t(i)$ is the history of observations and actions for agent i.

b) Higher-Order Beliefs:: Agents form higher-order beliefs recursively. The first-order belief $B_t^1(i)$ represents agent i's belief about the zeroth-order beliefs of other agents:

$$B_t^1(i) = p(B_t^0(j \neq i) | H_t(i))$$

and so on.

Agent i learns its zeroth-order belief $B_t^0(i)$ using an RNN to encode the history $H_t(i)$ into a neural belief state $b_t^0(i)$:

$$b_t^0(i) = \text{RNN}(b_{t-1}^0(i), Y_t(i))$$

where $Y_t(i)$ is the observation of agent i at time t. The belief state $B_t^0(i)$ is then obtained by mapping $b_t^0(i)$ through a generative model. The transition probabilities $P(s_t, a_t, o_t, s')$ are updated using Bayesian inference, reflecting the new observations o_t. The updated belief state $B_t^l(i)$, where l is an arbitrary higher-order belief, is computed as:

$$B_{t+1}^l(i) = \text{BayesUpdate}(B_t^l(i), a_t, o_t)$$

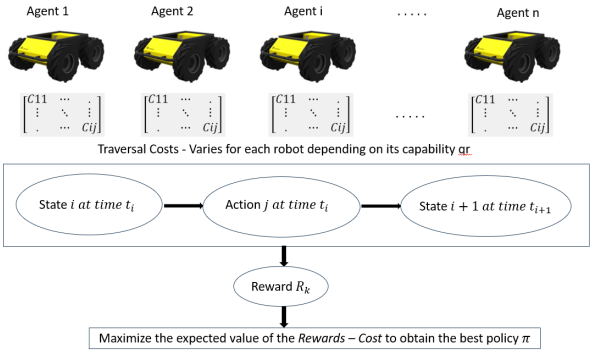

Fig. 2. Overview of the mathematical formulation

The optimization formulation can hence be extended to incorporate the recursive belief models (developed by [2]) and Bayesian updates as:

$$\max_{\pi} \left\{ \mathbb{E} \left[\sum_{l=0}^{\infty} \gamma^l \left(R(s_t, a_t, o_t, s_{t+1}) - \sum_i C^i(p_{ik}, p_{jl}, a_t, q_r) \right) \middle| \pi, B_l \right] \right\} \tag{6}$$

The policy is updated using reinforcement learning techniques. Figure 2 presents an overview of the layout of the optimization formula.

Agents must collaborate to maximize overall coverage and efficiency. A common policy that aligns the objectives of all agents facilitates coordinated actions. All data collected by individual agents are processed through a centralized system, which analyzes the changes in the environment and updates the shared policy accordingly. In addition, ensuring continuous communication under varying environmental conditions is necessary to enable real-time information sharing and coordination and to guarantee that no agent is out of the loop.

III. Experimental Design

Our evaluation of this approach uses a realistic 3D simulator capable of both simulating environmental physics and robot sensor and platform dynamics. This simulation is implemented in Unity to create a detailed and dynamic environment with different terrains, obstacles, and areas of interest. We incorporate elements that simulate adverse conditions, such as low-light areas or varying visibility levels. Each robot is equipped with simulated sensors for detecting areas of interest in the environment and communication tools for maintaining contact with other robots. We leverage the ML-Agents package to train the robots using reinforcement learning algorithms[1]. Each robot is an autonomous agent capable of learning from its environment and updating its policies based on the new information discovered by the robot itself, as well as its teammates. We set random probability distributions for target locations (prior beliefs) that are updated as the robots gather more data by exploring more areas in the environment.

[1]ML-Agents: https://github.com/Unity-Technologies/ml-agents

Fig. 3. The progression of a multi-robot team during a teammate-aware active search mission. Frame 1 shows the robots starting together, ready to begin their search. In Frame 2, the robots split up to explore multiple targets. Frame 3 illustrates the robots maintaining communication while being spread out, ensuring they stay connected.

Figure 3 illustrates the behavior of a multi-agent team during a teammate-aware search operation. In the most left frame, the entire human-robot team starts together, preparing for the search mission. In the middle frame. The robots split up to cover multiple targets, temporarily disconnecting from each other. Using our algorithm, despite being spread out, the robots reposition themselves to maintain communication with at least one other teammate.

Each robot shares a belief map of Gaussian probabilities indicating where the targets of interest are likely located in the environment. As the robots explore more areas of the environment, these probabilities dynamically adjust based on their findings. For instance, discovering a hidden path in a forest increases the likelihood of finding other hidden paths in similar areas (co-located changes). Conversely, finding an unexpected element, such as a building in an open field, reduces the assumed probability of similar findings in other areas like a nearby dense forest (unexpected findings). If a target is not found in one section of the environment, the probability of finding similar targets in related sections may decrease (absent evidence). However, if a target is not found in an area where it was not expected, such as a hidden path in a busy urban area, the probability of finding such targets elsewhere remains unchanged (expected absence). Additionally, failing to find a hidden path across multiple sections of the forest lowers the overall probability of finding any hidden paths (widespread absence). Finally, finding a benign or unrelated object, like a parked car on a road, does not affect the probability distribution in other areas (unrelated findings).

Performance metrics include search efficiency, robustness of communication, and the adaptability of the robots to environmental changes. We will evaluate the performance of the robots in various scenarios to assess the effectiveness of the reinforcement learning algorithm. By implementing this approach, we aim to demonstrate the feasibility of the use of reinforcement learning for teammate-aware active search in contested environments, providing an adaptable solution for real-world applications. To evaluate the effectiveness of our algorithm, we propose a concrete evaluation plan that includes:

- **Scenario Testing:** The experiments will be run in a variety of environments, with varying levels of complexity, including, but not limited to, static environment and dynamic environments.
- **Baseline Comparison:** The performance of the proposed MARL approach will be compared against other techniques and algorithms, with and without reinforcement learning. For instance, we will compare our results with recent MARL frameworks such as the Information-sharing Constrained Policy Optimization (IsCPO) method [6].
- **Real-time Monitoring:** The search efficiency will be quantified through the collected data to measure the rate of exploration as well as communication robustness and adaptability.
- **Post-Scenario Analysis:** The adaptation patterns of the agents will be evaluated after each experiment. This will include the search paths and communication logs between the agents.
- **Quantitative Metrics:** Quantifiable metrics will be used to measure performance, including search time, total distance traveled, communication uptime percentage, number of communication interruptions, time to adapt to new conditions, and success rate in target detection.

IV. Technical Progress

We have designed a custom reinforcement learning environment using the PPO (Proximal Policy Optimization) algorithm to train multiple agents to navigate towards goal positions. The current implementation, developed in a Gym-based environment, allows for agents to move within a 2D space, where they are spawned in random positions and aim to reach randomly placed goals. Each agent's movement is controlled by continuous actions, and their observations include their positions as well as the positions of the nearest goals. The reward function is designed to encourage agents to reduce their distance to the goals while penalizing movements out of bounds. So far, the implementation has demonstrated the capability of the agents to learn and navigate towards goals efficiently. However, to enhance the complexity and realism of the environment, we plan on adding additional constraints to make the reward function match the reward function presented in Section II.

V. Conclusion

In this work, we present the challenge of teammate-aware active search in adverse environments by integrating human operators into multi-robot teams. Our proposed approach leverages multi-agent reinforcement learning techniques to enable robots to robustly search for targets of interest while maintaining communication with each other and a human member of the team. The integration of recursive belief models and Bayesian updates will allow the robots to dynamically adapt their strategies as they explore more areas in the environment and gather new data. We also propose an evaluation plan to measure the effectiveness of our approach, by testing our algorithm in various scenarios, and comparing them with the state-of-the-art methods in the literature. This research will determine the feasibility of integrating multi-agent reinforcement learning with human-robot collaboration for teammate-aware active search in contested and dynamic environments.

REFERENCES

[1] C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, E. Howley, A. A. Irissappane, P. Mannion, A. Nowé, G. Ramos, M. Restelli, P. Vamplew, and D. M. Roijers, "A practical guide to multi-objective reinforcement learning and planning," *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 26, 2022.

[2] P. Moreno, E. Hughes, K. R. McKee, B. A. Pires, and T. Weber, "Neural recursive belief states in multi-agent reinforcement learning," *arXiv preprint arXiv:2102.02274*, 2021.

[3] Y. Wen, Y. Yang, R. Luo, J. Wang, and W. Pan, "Probabilistic recursive reasoning for multi-agent reinforcement learning," in *Proceedings of theInternational Conference on Learning Representations (ICLR)*, 2019.

[4] D. S. Drew, "Multi-agent systems for search and rescue applications," *Current Robotics Reports*, vol. 2, no. 2, pp. 189–200, 2021.

[5] F. Köpf, S. Tesfazgi, M. Flad, and S. Hohmann, "Deep decentralized reinforcement learning for cooperative control," *arXiv preprint arXiv:1910.13196*, 2019.

[6] Y. Okawa, H. Dan, N. Morita, and M. Ogawa, "Multi-agent reinforcement learning with information-sharing constrained policy optimization for global cost environment," *IFAC PapersOnLine*, vol. 56, no. 2, pp. 1558–1565, 2023.