HMA-SAR: Multi-Agent Search and Rescue for Unknown Located Dynamic Targets in Completely Unknown Environments

Xiao Cao
Adaptive Robotic Controls Lab (ArcLab)
Department of Mechanical Engineering
The University of Hong Kong, Hong Kong SAR, China
Email: u3009678@connect.hku.hk

***Abstract*—We present *HMA-SAR*, a heterogeneous multi-agent reinforcement learning (MARL) framework for search-and-rescue (SAR) with unknown, dynamic targets in completely unknown indoor maps. The method combines a timestamp-map state and reward shaping for sparse-reward indoor layouts, a Heterogeneous Curriculum Training (HCT) scheme with sequential policy updates and no-collision training, and a lightweight hybrid decision fallback that guarantees progress at test time. On 60×60 grid benchmarks with moving targets, HMA-SAR achieves higher success with fewer steps than HAPPO/MAPPO/MASAC and vastly outperforms frontier exploration. Gazebo and real-world TurtleBot tests corroborate feasibility.**

***Index Terms*—Multi-Agent Systems, Swarm Robotics, Search and Rescue, Reinforcement Learning, Curriculum Learning**

I. Introduction

Multi-agent search and rescue (MASAR) requires coordinated exploration, target search, and collision avoidance under partial observability and sparse rewards [1], [2]. Coverage or frontier strategies are effective for static scenes but degrade when targets move and re-enter explored regions. RL-based exploration improves efficiency [3]–[5], yet most studies assume static targets or known layouts. We address dynamic targets in unknown buildings, proposing a learning-and-planning framework that unifies exploration and target estimation, stabilizes training under teammate non-stationarity and map variability, and transfers to robots without intrinsic-curiosity modules.

II. Related Work

Frontier exploration [1] and patterned coverage [6] ignore target motion and thus revisit broad areas when targets circulate. MARL for exploration (independent encodings [7]; CTDE such as MAPPO/MADDPG [3], [8], [9]) rarely studies dynamic targets in unknown maps with sparse rewards. Curricula over size/obstacles [10], [11] and curiosity [12] help exploration; we instead use timestamp-based recency and simple shaping with a minimal fallback planner.

This work was supported by the General Research Fund (Grant 17204222) and the Seed Fund for Collaborative Research and General Funding Scheme—HKU-TCL Joint Research Center for Artificial Intelligence.

III. Method

We model MASAR as a Dec-POMDP $\langle \mathcal{S}, \{\mathcal{A}_i\}, \mathcal{T}, \{O_i\}, \mathcal{R}, \gamma \rangle$. Agents move on a 4-connected grid, $\mathcal{A}_i = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$, with unknown obstacles and M targets that step one cell per tick without crossing obstacles. Each agent observes a cropped occupancy map with timestamps and nearby agent poses. The objective is to maximize detected targets within a step budget while avoiding collisions. The overall data flow and training schedule are summarized in Fig. 1.

A. State, Reward, and Hybrid Decision

The state encodes unknown cells, obstacles, and free cells with a timestamp (time since last observation). Recent coordinates of the agent and teammates are added as distinct tags. This compact representation exposes explored structure, exposure gaps, and short-term motion traces without maintaining an explicit belief grid. The lightweight perception–control backbone used for both actor and critic is shown in Fig. 2.

The reward combines exploration and search. The exploration term counts newly uncovered free cells and penalizes collisions. An anti-stall term measures whether the A* distance (Manhattan heuristic) to the stalest region becomes shorter, nudging agents out of cul-de-sacs. The search term scales with the recency of visible free cells and gives a detection bonus. This setup targets sparse-reward indoor maps with long corridors and dead ends.

At test time, a hybrid decision prevents limit cycles: when an agent repeats nearby poses within a short window, it temporarily follows A* to the nearest unknown cell; if the area is fully explored, it moves to the stalest free cell. Control returns to the policy once progress resumes, so learned behavior is preserved while loops are avoided.

B. Probabilistic Intuition from Robot Motion

A one-dimensional example illustrates mass shifts in the absence of detections. Three adjacent cells start with equal probability. If the robot remains stationary and detects nothing, probability flows out through feasible target moves and in from neighbors; balancing these fluxes yields a 2:3:2 ratio.

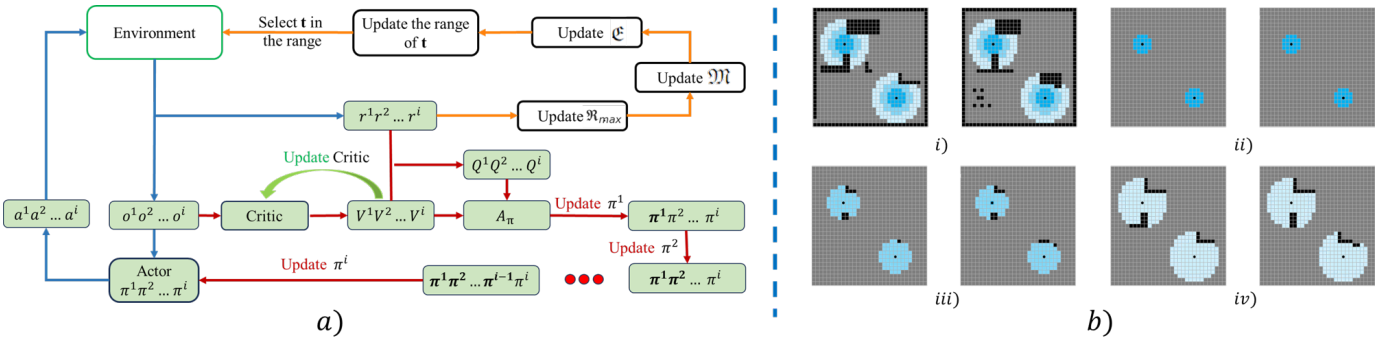

Fig. 1. HMA-SAR framework: environment interaction (blue), sequential policy updates (red), and horizon curriculum (orange).

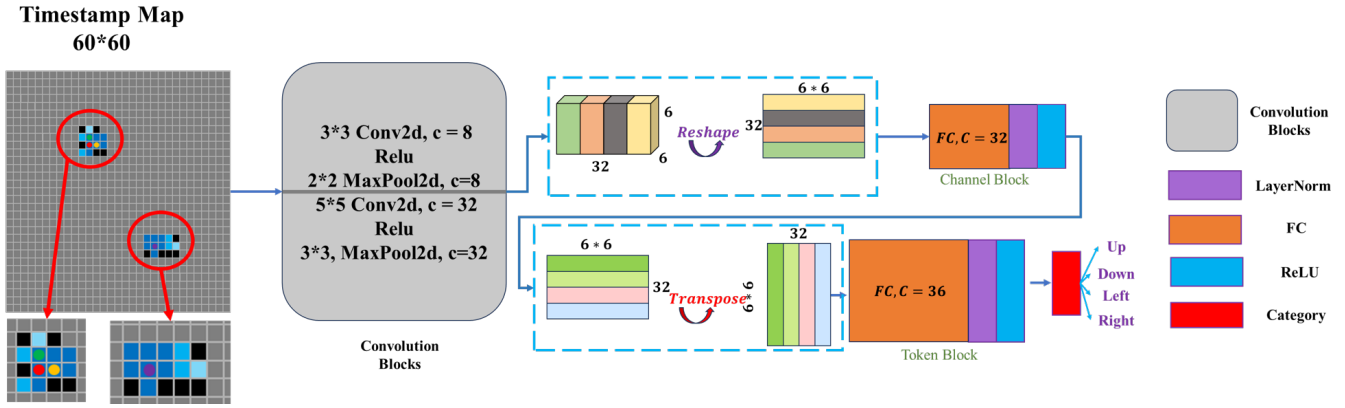

Fig. 2. Network architecture: lightweight CNN with channel/token mixing; categorical action head and scalar value head.

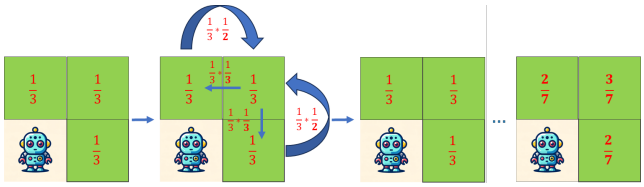

Fig. 3. Probability transition under stationary observation: undetected steps shift mass toward stale regions.

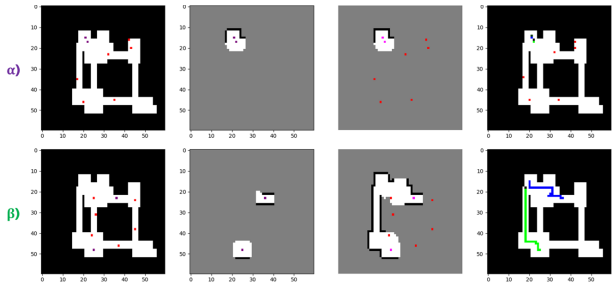

Fig. 4. Two scenarios: targets from the start; late spawns including in explored regions.

The timestamp signal approximates this effect in 2D: regions inspected less recently accumulate "suspicion," and shaping terms steer agents there when immediate cues are absent. The three-cell example and its steady-state ratio are illustrated in Fig. 3.

IV. Experimental Setup

We use 5,663 layouts normalized to 60×60. Two agents start within a small-radius cluster; six targets move one cell per step with equal directional probabilities and may stay. Visual range fluctuates between four and five cells to mimic noise. Two scenarios are evaluated: targets present from the start, and late instantiation after at least half the map is explored with at least two targets appearing in already explored areas. The two evaluation scenarios are depicted in Fig. 4. Training uses batch size 1,850 on a single 3080Ti. Baselines include frontier [1], MAPPO [3], MASAC [4], HAPPO with resets on collision, and HAPPO with "no-collision, stay-still". A* uses Manhattan heuristic and tie-breaking on path cost g. Timestamps decay linearly with steps since last observation.

V. Results

Over 250 held-out maps with a 250-step budget, HMA-SAR consistently outperforms learning and non-learning baselines. In the first scenario, it reaches a 97.6% success rate with the lowest average steps and the highest detections. Frontier lags because moving targets revisit explored regions and force repeated sweeps. In the second scenario, where targets appear late and sometimes in known areas, HMA-SAR maintains an 80.9% success rate, far above HAPPO, MAPPO, MASAC, and near-zero frontier. Timestamp recency and the anti-stall term

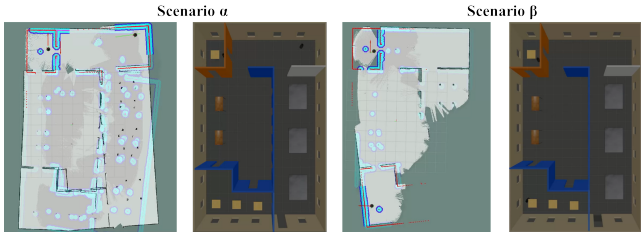

Fig. 5. Rviz/Gazebo outcomes: exploration and search against walking human targets.

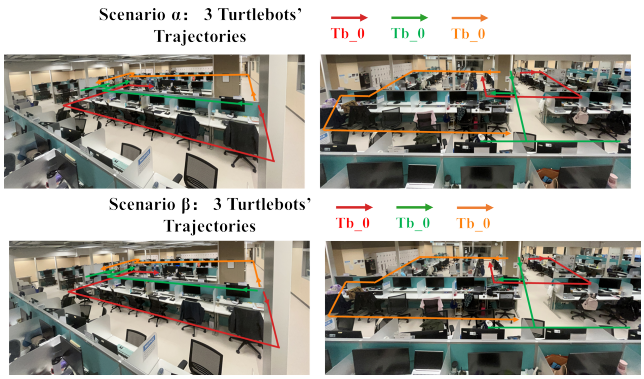

Fig. 6. Real-world TurtleBot trajectories: coverage in the first scenario and dynamic search in the second.

prioritize stale regions without abandoning expansion, which is crucial when targets oscillate between known and unknown space.

Scaling from two to five agents increases success and reduces steps by parallelizing coverage/search and reducing stalls at chokepoints. Removing no-collision training causes early penalties that bias policies away from narrow corridors; fixing a long horizon slows convergence and overfits to large rooms, whereas a staircase horizon stabilizes learning across shapes and sizes. Disabling the hybrid fallback yields occasional stalls on maze-like layouts; the fallback triggers rarely but reliably restores progress.

A. Extended Details and Runtime

PPO settings are standard: discount 0.99, GAE 0.95, clipping 0.2, Adam at 3×10^{-4}, minibatches of 256 with three epochs, entropy $\sim 10^{-3}$, and gradient norm clip 0.5. Perception uses a centered crop sized to the padded map; timestamps reset upon observation. The A* planner includes swap-prevention to avoid immediate reversals unless distance strictly reduces. With teammate map/pose sharing limited to a small radius or temporarily disabled, performance drops modestly because timestamps and fallback provide guidance. Inference is real-time on a commodity GPU; the planner runs on a cropped grid and is invoked only on detected loops. Representative outcomes in simulation are shown in Fig. 5, and real-world trajectories are visualized in Fig. 6.

VI. Discussion and Limitations

Timestamp recency serves as a lightweight surrogate for belief updates, while a horizon curriculum normalizes geometry and sequential updates temper non-stationarity from teammates. The fallback planner contributes robustness without dominating the learned policy. Main limitations are the discrete grid abstraction, simplified target dynamics, and no explicit bandwidth or latency model. Future work includes pyramid encoders for arbitrary map sizes, continuous control with richer dynamics, and explicit communication constraints.

References

[1] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping," in *AAAI/IAAI*, 2000, pp. 852–858.

[2] F. Niroui, B. Sprenger, and G. Nejat, "Robot exploration in unknown cluttered environments when dealing with uncertainty," in *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 2017, pp. 224–229.

[3] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 611–24 624, 2022.

[4] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[5] J. G. Kuba, R. Chen, M. Wen, Y. Wen, F. Sun, J. Wang, and Y. Yang, "Trust region policy optimisation in multi-agent reinforcement learning," *arXiv preprint arXiv:2109.11251*, 2021.

[6] Y. Gabriely and E. Rimon, "Spiral-STC: An on-line coverage algorithm of grid environments by a mobile robot," in *Proceedings 2002 IEEE International Conference on Robotics and Automation*, vol. 1, 2002, pp. 954–960.

[7] C. Wakilpoor, P. J. Martin, C. Rebhuhn, and A. Vu, "Heterogeneous multi-agent reinforcement learning for unknown environment mapping," *arXiv preprint arXiv:2010.02663*, 2020.

[8] H. Zhang, J. Cheng, L. Zhang, Y. Li, and W. Zhang, "H2GNN: Hierarchical-hops graph neural networks for multi-robot exploration in unknown environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3435–3442, 2022.

[9] A. H. Tan, F. P. Bejarano, Y. Zhu, R. Ren, and G. Nejat, "Deep reinforcement learning for decentralized multi-robot exploration with macro actions," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 272–279, 2023.

[10] Y. Yan, X. Li, X. Qiu, J. Qiu, J. Wang, Y. Wang, and Y. Shen, "Relative distributed formation and obstacle avoidance with multi-agent reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1661–1667.

[11] Z. Li, J. Xin, and N. Li, "Autonomous exploration and mapping for mobile robots via cumulative curriculum reinforcement learning," *arXiv preprint arXiv:2302.13025*, 2023.

[12] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research. PMLR, 2017, pp. 2778–2787.