2024 21st International Conference on Ubiquitous Robots (UR)
June 24 - 27, 2024, New York University, USA

Multi-Robot Autonomous Exploration and Mapping Under Localization Uncertainty via Reinforcement Learning on Graphs

Yewei Huang[1], Xi Lin[1] and Brendan Englot[1]

Abstract— **We propose a Deep Reinforcement Learning (DRL) based autonomous exploration algorithm designed for distributed multi-robot teams, which takes into account map and localization uncertainties of range-sensing mobile robots. An *exploration graph*, incorporating current SLAM pose estimation and potential future actions, is introduced to characterize the robot state at each iteration. A Graph Neural Network (GNN) is integrated into DRL agents to enhance their understanding of the topology within the exploration graph. The results of our experiments demonstrate the algorithm's capacity to strike a balance between ensuring map uncertainty and achieving efficient exploration with a multi-robot team.**

I. Introduction

Autonomous exploration describes a mobile robot navigating an unknown environment with no prior knowledge, utilizing onboard sensors to perceive its surroundings, and building an environment map based on the data gathered from these sensors. In contrast to Simultaneous Localization and Mapping (SLAM), where a robot's trajectory is typically predetermined by human experts, in this scenario, the robot autonomously determines its waypoints without any human intervention. Due to its full autonomy, autonomous exploration has great potential for applications such as exploring underwater environments, rapid rescue operations, and investigating outer space. In underwater environments, characterized by uncertainties arising from coastal ocean hydrodynamics and sensor noises, GNSS is often unavailable. The key to a successful mission lies in autonomous exploration that can guarantee accurate and precise localization.

In this paper, we introduce an autonomous exploration algorithm supported by reinforcement learning on graphs to strike a balance between exploration efficiency and localization accuracy for multi-robot teams. An exploration graph is introduced to abstract the robot's current status and historical trajectory. The policy is trained centrally with input from various robots and global SLAM map always available, enabling the algorithm to adapt to a range of scenarios. However, the trained policy is versatile and can be used for both centralized and distributed robot teams. It demonstrates robust performance even when consistent communication is not guaranteed and provided with partial information about the global SLAM map.

II. Problem Formulation and Approach

We address an autonomous exploration problem that is tightly coupled with a SLAM factor graph for a team of n robots. We make the assumption that the initial locations of all robots are sufficiently close to each other, enabling mutual observation among group members and facilitating an efficient map initialization process. Additionally, we impose a boundary on the exploration task, where the exploration process terminates upon fully exploring the enclosed environment.

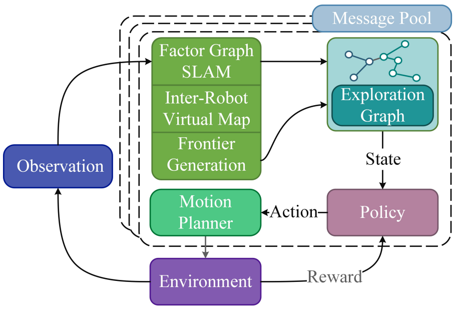

Fig. 1: **System Architecture.** The pipeline of the proposed approach.

A. Simultaneous Localization and Mapping

Let $N = \{1, 2, \cdots, n\}$ be the set of n robots. For each robot $\alpha \in N$, we denote its pose at timestamp i as $\mathbf{x}_{\alpha,i}$. The robot odometry observation between present pose $\mathbf{x}_{\alpha,i}$ and previous pose $\mathbf{x}_{\alpha,i-1}$ is described by the equation:

$$\mathbf{z}_{\alpha,i}^{\alpha,i-1} = f(\mathbf{x}_{\alpha,i-1}, \mathbf{x}_{\alpha,i}) + \epsilon_{\alpha,i}^{\alpha,i-1}. \tag{1}$$

Assume robot α observes the position of a landmark $\mathbf{l}_j$ at timestamp i, we can describe the landmark observation:

$$\mathbf{z}_{j}^{\alpha,i} = g(\mathbf{x}_{\alpha,i}, \mathbf{l}_j) + \epsilon_{j}^{\alpha,i}. \tag{2}$$

If another robot β is observed at timestamp i by robot α,we refer to this as a *robot rendezvous observation*:

$$\mathbf{z}_{\beta,j}^{\alpha,i} = f(\mathbf{x}_{\alpha,i}, \mathbf{x}_{\beta,j}) + \epsilon_{\beta,j}^{\alpha,i}, \tag{3}$$

where $f(\cdot)$ denotes the pose transformation between different robot poses, $g(\cdot)$ is the pose transformation from robot pose to landmark position, and $\epsilon_{\alpha,i}^{\alpha,i-1}$, $\epsilon_{j}^{\alpha,i}$ and $\epsilon_{\beta,j}^{\alpha,i}$ are zero-mean Gaussian noise variables.

At present timestamp t, $\mathcal{X} = \{\mathbf{x}_i^{\alpha} | \alpha \in N, i \in [0,t]\}$ represents the set containing the poses of all n robots from the initial timestamp 0 to the present timestamp t. $\mathcal{L} = \{\mathbf{l}_0, \mathbf{l}_1, \ldots, \mathbf{l}_m\}$ is the set of landmarks observed until timestamp t. Additionally, let $\mathcal{Z}$ be the set containing odometry observations, landmark observations and robot rendezvous observations from all timestamps. The SLAM problem can be framed as a maximum a posteriori estimation problem [1]:

$$\mathcal{X}^*, \mathcal{L}^* = \underset{\mathcal{X},\mathcal{L}}{\arg\max}\, P(\mathcal{X}, \mathcal{L} | \mathcal{Z}). \tag{4}$$

[1] Y. Huang, X. Lin and B. Englot are with Stevens Institute of Technology, Hoboken, NJ, USA. {yhuang85, xlin26, benglot}@stevens.edu

This work was supported by ONR Grants N00014-20-1-2570 and N00014-21-1-2161.

979-8-3503-6107-0/24/$31.00 ©2024 IEEE

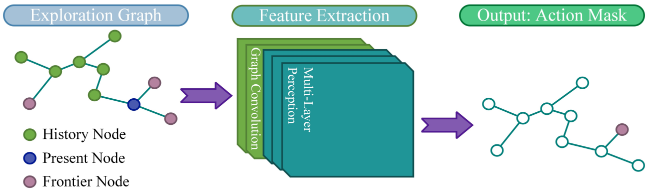

Fig. 2: The processing pipeline of the GCN-DQN structure.

B. Autonomous Exploration with Reinforcement Learning

Building upon our previous work in single-robot exploration [2], we formulate the autonomous exploration problem as a Markov Decision Process (MDP) $< S, A, P, R, \gamma >$, where $S = \{s_i\}$ is the set of states expressed by the exploration graph, and $A = \{a_i\}$ indicates the sets of actions. $R(\cdot)$ is a reward function, $P(s_{i+1}|s_i, a_i)$ is the transition function and γ is the discount factor.

At timestamp t when the robot reaches the current goal position, the agent constructs an exploration map s_t by utilizing the current SLAM solution and considering potential new goals. Subsequently, it chooses an action a_t, determining a new goal position. After taking action a_t, in this instance, when the robot reaches its next goal position, the agent transitions to a new state s_{t+1} and receives a corresponding reward $\mathcal{R}(s_t, a_t)$. In RL, the selection of actions is determined by a policy π, and this policy is derived based on the Bellman equation [3]:

$$Q^*(s_t, a_t) = \\ \mathbb{E}[R(s_t, a_t) + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})], \tag{5}$$

where $Q^*(\cdot)$ is the optimal value function, which estimates the future reward associated with state s_t and action a_t. The optimal policy is then acquired by $\pi(s) = \arg\max_{a\in A} Q^*(s, a)$.

III. Methodology and Experiments

In this section, we present the details of the proposed system. A visual representation of our approach is illustrated in Fig. 1. We consider a collaborative group of mobile robots, equipped with identical range sensors. A centralized or decentralized factor-graph based SLAM algorithm is assumed to operate at a specific frequency.

Once a robot α successfully reaches its current goal at timestamp t, its virtual map is updated. Subsequently, a set of potential new goals, referred to as frontiers $\mathcal{F}_t$, is selected from this virtual map. The virtual map represents the robot's knowledge of the environment, and its confidence in that knowledge, at the timestamp when decision-making is required. Three types of frontiers are identified for selection: exploration frontiers close to the robot's latest position, revisiting frontiers near previously visited landmarks, and rendezvous frontiers, which are the current goal positions of neighboring robots. The policy network selects a new goal from the frontiers. Finally, the motion planner is activated to formulate a series of actions guiding robot α to the new goal.

We adopt a Deep Q-learning (DQN) [3] policy-based DRL algorithm with a Graph Convolutional Network (GCN) [4] structure, based on the experiments in our earlier work [2], following the framework outlined in Fig. 2. We employ a centralized training and decentralized action strategy. The robot team collaboratively shares optimized SLAM trajectories, landmark positions, and latest goal positions among its members during the training process. However, individual robots have the capability to take actions locally, even in situations where they may experience delays in message passing from their neighbors.

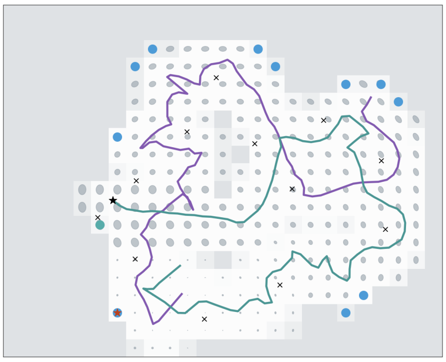

Fig. 3: A simple example of a virtual map with the observed region in white, and the gray ellipses representing the covariances of each grid cell's center.

A. Virtual Map

The virtual map $\mathcal{M}$ of a finite environment is generated from the robot poses $\mathcal{X}$ and their associated observations $\mathcal{Z}$. Assuming that $\mathcal{M}$ comprises b map cells $\mathbf{m}_i$ referred to as *virtual landmarks*, the posterior can be redefined as follows:

$$P(\mathcal{M}|\mathcal{X}, \mathcal{Z}) = \prod_{\mathbf{m}_i \in \mathcal{M}} P(\mathbf{m}_i|\mathcal{X}, \mathcal{Z}). \tag{6}$$

The likelihood of virtual landmark $\mathbf{m}_i$ being observed is $q(\mathbf{m}_i) = \mathbb{E}[P(\mathbf{m}_i|\mathcal{X}, \mathcal{Z})]$. When this virtual landmark is observed multiple times, the procedure for updating $q(\mathbf{m}_i)$ is similar to updating map cell values in an occupancy grid map [5]. We assume the virtual landmark $\mathbf{m}_i$ is observed by a robot pose $\mathbf{x}_{\alpha,j}$, with its estimated value denoted $\hat{\mathbf{x}}_{\alpha,j}$, and a marginal covariance $\Sigma_{\mathbf{x}_{\alpha,j}}$. We can compute the covariance: $\Sigma_{\mathbf{m}_i} = \mathbf{H} \cdot \Sigma\mathbf{x}_{\alpha,j} \cdot \mathbf{H}^\intercal$. Here, $\mathbf{H} = \frac{\partial g(\mathbf{x}_{\alpha,j}, \mathbf{m}_i)}{\partial \mathbf{x}_{\alpha,j}}|\hat{\mathbf{x}}_{\alpha,j}$ represents the Jacobian matrix obtained by differentiating the landmark observation model $g(\mathbf{x}_{\alpha,j}, \mathbf{m}_i)$ with respect to estimated robot pose $\hat{\mathbf{x}}_{\alpha,j}$. We utilize Covariance Intersection to compute the covariance of a virtual landmark that is observed by multiple robot states, as detailed in [6]. Fig. 3 shows inter-robot virtual maps built from both local robot states and neighbors' robot states received by local robot α. The observed regions are highlighted in white; gray ellipses show covariances describing the uncertainty of the map's cells.

B. Environment

We design a simulation environment with landmarks. Each robot is equipped with inertial dead reckoning with a gyro error of 0.5° and an accelerometer yield error of $0.05m$. Additionally, the robots are equipped with sonar sensors with a range error of $0.002m$, bearing error of 0.5°, and a maximum sensing range of $7.5m$. During each step, the robot is capable of rotating by 15°, maintaining a constant speed of 1 meter per second. Movement is constrained to the direction of the current heading. The initial positions of the robots are strategically placed within the sensing range of each other to ensure a proper SLAM initialization. We employ the Artificial Potential Field (APF) [7] method as the local planner for obstacle avoidance. To prevent robots from exiting the mission area, the boundaries along the environment's edges are restricted from selection as frontiers.

C. States

In our method the state is expressed as an *exploration graph* $S =< \mathcal{V}, \mathcal{E} >$. The set of vertices $\mathcal{V}$ consist of four different types: current robot pose vertex, historical robot pose vertex, landmark vertex and frontier vertex. Each vertex $\mathbf{v}_i = (\mathbf{v}_{\text{dist}}, \mathbf{v}_{\text{trace}}, \mathbf{v}_{\text{type}})$, with $\mathbf{v}_{\text{dist}} = \|\mathbf{v}_i - \mathbf{x}_{\alpha,t}\|$ being the Euclidean distance between the vertex and the current robot location $\mathbf{x}_{\alpha,t}$. $\mathbf{v}_{\text{trace}} = \phi_A(\Sigma_{\mathbf{v}_i})$ is the A-Optimality criterion of the covariance matrix $\Sigma_{\mathbf{v}_i}$ of vertex $\mathbf{v}_i$. For current robot pose vertices, historical robot pose vertices and landmark vertices, $\Sigma_{\mathbf{v}_i}$ is the inverse of the marginal information matrix for the associated poses or positions from the SLAM factor graph. The covariance matrix $\Sigma_{\mathbf{v}_i}$ for a frontier vertex is identical to that of the nearest grid cell's center in the virtual map.

$$\mathbf{v}_{\text{type}} = \begin{cases} 1, & \mathbf{v}_i = \mathbf{x}_{\alpha,t} \\ 2, & \mathbf{v}_i \in \mathcal{F}_t \\ 3, & \mathbf{v}_i \neq \mathbf{x}_{\alpha,t}, \mathbf{v}_i \notin \mathcal{F}_t \end{cases} \tag{7}$$

The edge set $\mathcal{E}$ consists of five types; all edges are weighted with the Euclidean distances between those vertices:

- $\mathbf{e}(\mathbf{x}_{\gamma,i}, \mathbf{x}_{\gamma,i+1})$: Edges connecting adjacent robot poses.
- $\mathbf{e}(\mathbf{x}_{\gamma,i}, \mathbf{l}_j)$: Edges connecting a robot pose and a landmark.
- $\mathbf{e}(\mathbf{f}_j, \mathbf{x}_{\alpha,t})$: Edges connecting the current robot pose and an exploration frontier.
- $\mathbf{e}(\mathbf{f}_j, \mathbf{l}_i)$: Edges connecting a landmark and a revisiting frontier.
- $\mathbf{e}(\mathbf{f}_j, \mathbf{x}_{\gamma,i})$: Edges connecting the current robot pose and a rendezvous frontier.

D. Actions

In this problem, the action is a binary vector where 1 indicates an allowed action and 0 indicates a forbidden action. The action's dimension matches the number of frontiers at each timestamp t.

E. Reward

We design a reward function aiming to motivate the learning agent to obtain a higher final exploration ratio, minimize traverse distance, all while maintaining a consistently high level of localization accuracy:

$$\mathbf{r}_t = \begin{cases} r_{\text{base,t}} + r_{\text{end}}, & \text{if } \sigma_t > \sigma_{\max} \\ r_{\text{base,t}} + r_{\text{goal}}, & \text{if } \eta_t > \eta_{\min} \\ r_{\text{base,t}} + r_{\text{incre}}, & \text{if } \sigma_{t-1} - \sigma_t > \Delta\sigma_{\min} \\ r_{\text{base,t}}, & \text{otherwise} \end{cases} \tag{8}$$

$$r_{\text{base,t}} = r_{\text{step}} + \alpha(d_t - d_{t-1}) + \beta(s_t - s_{t-1}). \tag{9}$$

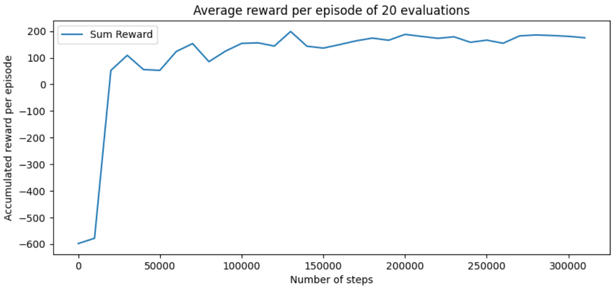

Fig. 4: The mean accumulated reward per episode at different steps.

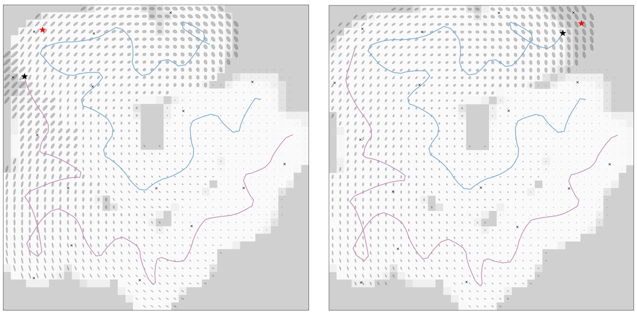

Fig. 5: In the $80m \times 80m$ virtual map created by the robot team, gray ellipses depict the uncertainty of each cell. The current robot position is denoted by a black star, and the newly selected target state is represented by a red star. Landmarks are expressed by black crosses. The agent chooses to revisit a landmark previously observed by its neighbor (left) and successfully lowers the map uncertainty (right).

d_t represents the distance the local robot traversed from timestamp 0 to timestamp t. s_t denotes the area explored by the local robot from timestamp 0 to timestamp t. σ_t represents the Root Mean Square Error (RMSE) of the local SLAM map for robot at timestamp t. The maximum allowable RMSE for SLAM is denoted as $\sigma_{\max}$, and we set $\sigma_{\max} = 1$ in our experiments. $\Delta\sigma_{\min}$ indicates the minimum decrease in SLAM RMSE required for a reward $r_{\text{incre}} = 10$. η_t indicates the exploration ratio calculated from the virtual map at timestamp t, and $\eta_{\min} = 85\%$ represents the minimum final exploration ratio for a successful exploration. For our experiments, we set $r_{\text{step}} = -1$, $r_{\text{end}} = -100$ and $r_{\text{goal}} = 300$.

F. Experiments and Results

We train our policy in a $80m \times 80m$ environment with 20 landmarks and visualize the overall learning performance in Fig. 4. In every training episode, the robot teams are initialized at a random position within the sensing range of their teammates. An episode is terminated when the RMSE of SLAM $\sigma_t > \sigma_{\max}$, a minimum exploration ratio $\eta_{\min}$ is reached, or a robot has moved than 300 steps. To evaluate the proposed system's performance, 20 environments of size $80m \times 80m$ are generated in advance, each containing 20 randomly located landmarks and a set of initial robot positions for robot team members. A representative result, illustrating a landmark revisiting action, is shown in Fig. 5.

References

[1] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[2] F. Chen, J. D. Martin, Y. Huang, J. Wang, and B. Englot, "Autonomous exploration under uncertainty via deep reinforcement learning on graphs," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 6140–6147.

[3] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, G. Dulac-Arnold, J. Agapiou, J. Leibo, and A. Gruslys, "Deep q-learning from demonstrations," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[4] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[5] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[6] J. Wang, F. Chen, Y. Huang, J. McConnell, T. Shan, and B. Englot, "Virtual maps for autonomous exploration of cluttered underwater environments," *IEEE Journal of Oceanic Engineering*, vol. 47, no. 4, pp. 916–935, 2022.

[7] X. Fan, Y. Guo, H. Liu, B. Wei, and W. Lyu, "Improved artificial potential field method applied for AUV path planning," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–21, 2020.