2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Graph Neural Networks for Multi-Robot Active Information Acquisition

Mariliza Tzes*, Nikolaos Bousias*, Evangelos Chatzipantazis and George J. Pappas

***Abstract*— This paper addresses the Multi-Robot Active Information Acquisition (AIA) problem, where a team of mobile robots, communicating through an underlying graph, estimates a hidden state expressing a phenomenon of interest. Applications like target tracking, coverage and SLAM can be expressed in this framework. Existing approaches, though, are either not scalable, unable to handle dynamic phenomena or not robust to changes in the communication graph. To counter these shortcomings, we propose an Information-aware Graph Block Network (I-GBNet), an AIA adaptation of Graph Neural Networks, that aggregates information over the graph representation and provides sequential-decision making in a distributed manner. The I-GBNet, trained via imitation learning with a centralized sampling-based expert solver, exhibits permutation equivariance and time invariance, while harnessing the superior scalability, robustness and generalizability to previously unseen environments and robot configurations. Numerical simulations on significantly larger graphs and dimensionality of the hidden state and more complex environments than those seen in training validate the properties of the proposed architecture and its efficacy in the application of localization and tracking of dynamic targets.**

I. Introduction

Over recent years, we have seen UAVs participating in the fighting of wildfires in California, robotic snakes assisting in localization of earthquake victims or UUVs deployed for underwater ocean mapping [1]. In such scenarios, autonomous robots need to sequentially make decisions that would allow them to better estimate the extent of the fire and decide upon extinguishing locations. The process of designing efficient paths to actively estimate a hidden state that expresses such a phenomenon of interest by utilizing measurement readings of on-board sensors is known in the literature as *Active Information Acquisition* (AIA). An exhaustive list of applications can be formulated in the AIA setup, from environmental monitoring, search & rescue, surveillance and coverage, target tracking and localization and active-SLAM [2]–[6].

A plethora of existing works [7]–[17] are proposed that solve the AIA problem. Myopic approaches [7]–[9] rely on computing controllers that incur the maximum immediate decrease of an uncertainty measure while search-based non-myopic schemes [10]–[12] solve the problem by pruning the exploration process. The latter return suboptimal solutions while their decentralized counterparts rely on coordinate descent, making them computationally intractable as the planning horizon and/or the number of robots increases. Non-myopic sampling-based approaches [13]–[17] gained popularity due to their ability to compute informative paths fast. All these approaches suffer from the following limitations: 1) they do not *scale*, e.g. for more than dozens of robots, 2) are not *robust* to changes of the problem's parameters, e.g. number of agents or changes to the connectivity of robots and/or 3) do not address dynamic phenomena of interest.

The authors are with the GRASP Lab, University of Pennsylvania, Philadelphia, PA 19104, USA, {mtzes, nbousias, vaghat, pappasg}@seas.upenn.edu.

This work was supported by the ARL grant DCIST CRA W911NF-17-2-0181.

*equal contribution

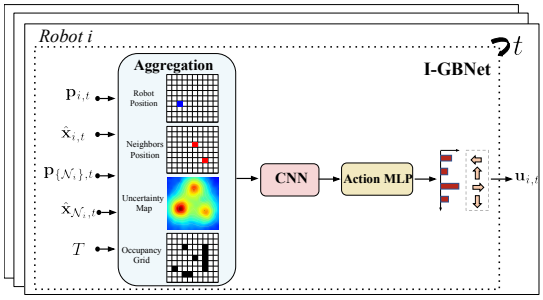


Fig. 1: In this Figure we illustrate our graph parametrization of the AIA problem with the I-GBNet. Robot i collects its neighbors' positions and estimates, along with its own attributes and occupancy map and feeds them to the network. The latter produces the next action, the robot updates its position and takes new measurements. The process is repeated per robot and timestep.

Graph Neural Networks (GNNs) are information processing architectures for signals on graphs. Thanks to their distributed structure, GNNs can easily be adapted to multi-robot applications where the nodes and edges represent the robots and communication links respectively. They follow a neighborhood aggregation scheme and are therefore inherently suitable for distributed implementations. GNNs are also promising for their transferability across different graphs and can be designed such that they generalize to previously unseen multi-robot scenarios. Their use has the potential to radically increase (a) robustness to changes in the topology of the graph and (b) scalability with respect to the number of robots.

In this paper, we propose a novel architecture based on GNNs that solves the AIA problem in a distributed manner and addresses the aforementioned challenges. Specifically, we translate the multi-robot information gathering problem to a graph representation and formulate it as a sequential decision-making problem, where an Information-aware Graph Block Network (I-GBNet) learns to derive control actions that drive the robots to actively estimate the state of the phenomenon of interest. At every timestep, each robot collects positions and estimates from its neighbors and feeds them along with its own attributes to the I-GBNet that produces the next control inputs to be applied; see also Fig.1. To train our network, we use imitation learning where the

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10160723

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:17 UTC from IEEE Xplore. Restrictions apply.

expert is a centralized sampling-based algorithm.

Literature Review: An exhaustive literature has been developed that addresses the AIA problem. A centralized sampling based approach is introduced in [15] that explores the joint physical and information space of the robots. In [16] the authors propose a decentralized Monte Carlo Tree Search [18] method that samples the individual action space of each robot and then creates a sparse approximation of the joint action space. In [17], the authors design a distributed sampling-based approach where robots build their own trees and share with their neighbors information of randomly sampled nodes of their trees. The authors in [19] propose a more scalable approach by tessellating the environment into Voronoi diagrams, but an all-to-all communication and static hidden state is assumed. Data-driven methods [20]–[23] have also been proposed to allow for online implementations. Prior work in data-driven information acquisition utilizes imitation learning via a clairvoyant oracle [21], [24]. In [22] the authors develop a deep reinforcement learning technique that combines tree search with an offline-learned neural network, however it is only applicable to a single robot.

To the best of our knowledge, GNNs have never been used before to frame an AIA setting. Noticeably, they recently started gaining popularity in multi-robot applications [25]–[32]. In [26] GNNs are employed for path planning where an attention mechanism prioritizes meaningful information and the authors in [28], [29] propose GNNs for training control policies for multi-robot coverage. GNNs have also been exploited for SLAM [30], [32] and target tracking [31] where common in these works is the absence of notion of uncertainty over the robots' observations. Compared to these works, our scheme provides directions on to where the robots should take measurements on-the-fly to reduce their underlying uncertainty.

Contributions: This paper proposes the *first* method for Active Information Acquisition using Graph Neural Networks. We designed a network that is *scalable* with respect to the number of robots and dimensionality of the hidden state to be estimated and *generalizes* to previously unseen multi-robot configurations. Our method is *robust* to communication failures and can be applied for *time-varying* communication graphs and *dynamic* hidden state. Additionally, we provide a variety of *quantitative* and *qualitative* simulations that illustrate the efficacy of our proposed architecture for the example of target localization and tracking.

II. Problem Formulation

Consider a homogeneous team of N mobile robots that reside in a complex environment $T \subset \mathbb{R}^2$. The environment is cluttered with a set of arbitrarily shaped obstacles $\mathcal{O} \subset T$, thus forming the obstacle-free area $T_{\text{free}} := T \setminus \mathcal{O}$. The robots are governed by the *robot dynamics* $\mathbf{p}_{i,t+1} = f(\mathbf{p}_{i,t}, \mathbf{u}_{i,t}),\ \forall i \in \{1, \dots, N\}$ where $\mathbf{p}_{i,t} \in T_{\text{free}}$ and $\mathbf{u}_{i,t} \in \mathcal{U}$ denote the state of robot i at time t and control action selected from a finite set of admissible control actions, $\mathcal{U}$. The robots are tasked with collaboratively estimating a hidden state $\mathbf{x}_t \in \mathbb{R}^{d_x}$ with *hidden state dynamics* $\mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{w}_t$, where $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ is the additive noise and $\mathbf{Q}_t$ is the noise covariance matrix. The robots are equipped with sensors capable of collecting noisy measurements of the hidden state as per the *observation model*

$$\mathbf{y}_{i,t} = M(\mathbf{p}_{i,t})\mathbf{x}_t + \mathbf{v}_{i,t}, \quad \mathbf{v}_{i,t} \sim \mathcal{N}(0, \mathbf{R}_i) \tag{1}$$

where $\mathbf{y}_{i,t}$ is the measurement of robot i at time t. We assume that the transition matrices $\{A, M\}$ and the noise covariance matrices $\{\mathbf{Q}_t, \mathbf{R}_i\}$ are known, similar to [17] and the robots perfectly localize themselves in a global frame. Hereafter, when subscript i is omitted we denote compactly attributes of all robots.

A. Multi-Robot Active Information Acquisition

The AIA problem requires the robots to collaboratively estimate the hidden state $\mathbf{x}_t$. To do so, the robots maintain a prior Gaussian distribution for the hidden state, $\hat{\mathbf{x}}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0)$. Given the prior distribution and measurements received up to time t, $\mathbf{y}_{0:t}$, the à-posteriori distribution is computed i.e., $\hat{\mathbf{x}}_t \sim \mathcal{N}(\boldsymbol{\mu}_t(\mathbf{y}_{0:t}), \Sigma_t(\mathbf{y}_{0:t}))$ where $\boldsymbol{\mu}_t(\mathbf{y}_{0:t})$ and $\Sigma_t(\mathbf{y}_{0:t})$ denote the à-posteriori mean and covariance matrix respectively. An indicative function of the overall quality of the measurements up to time t in estimating $\mathbf{x}_t$ is the determinant of the à-posteriori covariance matrix, i.e $\det \Sigma_t(\mathbf{y}_{0:t})$ [33]. Alternative information measures could be used such as the trace or the maximum eigenvalue of the covariance. Given initial robots' positions $\mathbf{p}_0$ and prior distribution $\hat{\mathbf{x}}_0$, the goal is to compute a planning horizon F and a sequence of control inputs $\boldsymbol{u}_{0:F}$, which solves the following deterministic optimal control problem [15]:

$$\min_{F, \boldsymbol{u}_{0:F}} \left[J(F, \boldsymbol{u}_{0:F}) = \sum_{t=0}^{F} \det \Sigma_{t+1} \right] \tag{2a}$$

$$\det \Sigma_{F+1} \leq \epsilon, \tag{2b}$$

$$\mathbf{p}_{t+1} \in T_{\text{free}}^N, \tag{2c}$$

$$\mathbf{p}_{t+1} = f(\mathbf{p}_t, \mathbf{u}_{i,t}), \quad \forall i = \{1, \dots, N\} \tag{2d}$$

$$\Sigma_{t+1} = \rho(\mathbf{p}_t, \Sigma_t) \tag{2e}$$

where the objective (2a) captures the cumulative uncertainty in the estimation of $\mathbf{x}_t$ after fusing information collected by all robots from $t = 0$ up to time F and the constraint in (2b) requires that the final uncertainty of the state $\mathbf{x}_F$ is less than a user-specified threshold ϵ. In (2c) we require obstacle free paths and (2d) describes the robot dynamics. Finally in (2e), $\rho(\cdot)$ stands for the Kalman Filter update rule that computes the à-posteriori distribution. A Centralized Sampling-Based (C-SB) method that solves Problem (2) is introduced in [15, Section III] that is probabilistically complete and asymptotically optimal.

Remark 1: The optimization problem defined in (2) resulted from a stochastic optimal control problem defined in [34] where a separation principle is applied due to the linear Gaussian assumptions.

B. Problem Statement

Given initial positions of the robots $\mathbf{p}_0$, capable of taking measurements as per the observation model (1), communicating via an underlying communication network and prior

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:17 UTC from IEEE Xplore. Restrictions apply.

estimates $\hat{\mathbf{x}}_0$, we design a GNN architecture that (i) solves (2) in a distributed manner (ii) is trained using imitiation learning to mimic the C-SB expert and (iii) derives sequential control actions $\mathbf{u}_t$ that allow the robots to actively decrease their uncertainty over $\mathbf{x}_t$.

III. Graph Neural Networks for AIA

To solve the Multi-Robot Active Information Acquisition problem described in Section II in a distributed way with GNNs, we develop a suitable graph representation of the problem. At every timestep the graph is given as an input to the network that consists of an Information-aware Graph Block Network (I-GBNet) and a Multi-Layer Perceptron (MLP). I-GBNet updates the node attributes of the incoming graph and the updated attributes are fed to the MLP that derives the control actions for all robots.

A. Graph Representation of AIA problem

In the distributed setting, we assume that each robot maintains a local estimate over the hidden state expressed via its à-posteriori Gaussian distribution $\hat{\mathbf{x}}_{i,t} \sim \mathcal{N}(\boldsymbol{\mu}_{i,t}, \Sigma_{i,t})$. For simplicity reasons, we define $Z_{i,t} := M(\mathbf{p}_{i,t})^T \mathbf{R}_{i,t}^{-1} M(\mathbf{p}_{i,t})$ and the information matrix $\Omega_{i,t} = \Sigma_{i,t}^{-1}$.

Let the robots communicate through an underlying communication network modeled as a directed *communication graph* $\mathcal{G}_{c,t} = (\mathcal{V}_c, \mathcal{E}_{c,t})$. The set of vertices $\mathcal{V}_c \coloneqq \{1, \dots, N\}$ are indexed by the robots and an edge $(i,j) \in \mathcal{E}_{c,t} \subseteq \mathcal{V}_c \times \mathcal{V}_c$ defines a direct communication between robots i and j. Robot i can exchange information with the set of its neighbors $\mathcal{N}_{i,t}$. Hereafter we define $\tilde{\mathcal{N}}_{i,t} \coloneqq \mathcal{N}_{i,t} \cup \{i\}$.

Given the aforementioned communication graph $\mathcal{G}_{c,t}$, let the state of the problem at timestep t be described by the following homogeneous, weighted, directed *network graph*:

$$\mathcal{G}_t = \{\mathcal{V}_c, \mathcal{V}_t, \mathcal{E}_t, \mathcal{W}_t; \mathbf{g}\} \tag{3}$$

where $\mathcal{V}_t = \{\mathbf{V}_{i,t}\}_{i=1}^N$ is the set of node attributes of cardinality $|\mathcal{V}_t| = N$, indexed by the robots $i \in \mathcal{V}_c$, $\mathcal{E}_t = \mathcal{E}_{c,t} \cup \{(i,i), \forall i \in \mathcal{V}_c\}$ is the set of edges, $\mathcal{W}_t \in \mathbb{R}^{N \times N}$ is the weight matrix and $\mathbf{g}$ represents the global attributes of the graph (e.g. environment T). Each node (robot) i contains the following *node attribute*:

$$\mathbf{V}_{i,t} = \{\mathbf{p}_{i,t}, Z_{i,t}, \boldsymbol{\mu}_{i,t}, \Omega_{i,t}\} \tag{4}$$

The weight matrix quantifies the importance of information being propagated from robot i to j and is defined as $[\mathcal{W}_t]_{ij} = \kappa_{ij} \mathbb{1}_{(i,j) \in \mathcal{E}_t}$, where $\kappa_{ij} > 0$ and $\sum_{j \in \tilde{\mathcal{N}}_{i,t}} \kappa_{ij} = 1$.

B. Information-aware Graph Block Network

The core of our GNN is I-GBNet Φ, illustrated in Fig.2 and inspired by [35]. I-GBNet receives a graph signal $\mathcal{G}_t$ and updates the node attributes by propagating information through the graph, while keeping the graph topology unchanged, i.e. $\mathcal{G}'_t = \Phi(\mathcal{G}_t)$. For the design of I-GBNet, we finely discretize the environment $T \subset \mathbb{R}^2$ and represent it via an occupancy grid-map $\mathcal{T}$ of size $\mathcal{T}_H \times \mathcal{T}_W$. The network Φ comprises of three parts i) two aggregation functions $\rho^{\Omega}(\cdot), \rho^{p}(\cdot)$, (ii) a heatmap projection function $\mathcal{H}$ and (iii) a node-update function ϕ^v.

Aggregation: We assume that at timestep t, the robot i receives information of the form $\{(\Omega_{j,t}, \mathcal{I}(\mathbf{p}_{j,t})), \forall j \in \tilde{\mathcal{N}}_{i,t}\}$, where $\mathcal{I}(\cdot)$ is a function that transforms the robot's position into a binary robot-grid map. Let $\boldsymbol{\Omega}_{\tilde{\mathcal{N}}_{i,t},t} = \{\Omega_{j,t}, \forall j \in \tilde{\mathcal{N}}_{i,t}\}$ and $\mathbf{I}_{\tilde{\mathcal{N}}_{i,t},t} = \{\mathcal{I}(\mathbf{p}_{j,t}), \forall j \in \tilde{\mathcal{N}}_{i,t}\}$. Inspired by the active information problem setup and constraint (2e), the aggregation function ρ^{Ω} is defined as the Distributed Kalman Filter (DKF), proposed in [33], where robot i updates its à-posteriori covariance matrix as follows:

$$\hat{\Omega}_{i,t} = \rho^{\Omega}(\boldsymbol{\Omega}_{\tilde{\mathcal{N}}_{i,t},t}, Z_{i,t}; \mathcal{W}_t) = \sum_{j \in \tilde{\mathcal{N}}_{i,t}} \kappa_{ij} \Omega_{j,t} + Z_{i,t} \tag{5}$$

Each robot, also, aggregates the positions of its neighbours into a neighbors-grid map as:

$$\hat{\mathbf{p}}_{i,t} = \rho^{p}(\mathbf{I}_{\tilde{\mathcal{N}}_{i,t},t}) = \sum_{j \in \tilde{\mathcal{N}}_{i,t}} \mathcal{I}(\mathbf{p}_{j,t}) - \mathcal{I}(\mathbf{p}_{i,t}) \tag{6}$$

The selected aggregation functions are invariant to permutations of their inputs by construction.

Heatmap Projection: Once the robot computes the aggregated covariance matrix $\hat{\Omega}_{i,t} \in \mathbb{R}^{d_x \times d_x}$, it projects it into the occupancy grid map $\mathcal{T}$ via the projection function $\mathcal{H}\colon \{\hat{\Omega}_{i,t}, \boldsymbol{\mu}_{i,t}\} \rightarrow H_{i,t}$, where $H_{i,t}$ is the resulting grid heatmap that represents the magnitude of uncertainty in $\mathcal{T}$. In the case of multi-target localization, the projection module would produce a combination of the posterior Gaussian heatmaps.

Node-Update: Given the aggregated neighbors-grid map $\hat{\mathbf{p}}_{i,t}$, the robot-grid map $\mathcal{I}(\mathbf{p}_{i,t})$, the heatmap-grid $H_{i,t}$ and the occupancy-grid map $\mathcal{T}$, the node-update function ϕ^v produces the control actions

$$\mathbf{u}_{i,t} \coloneqq \phi^v(\hat{\mathbf{p}}_{i,t}, \mathcal{I}(\mathbf{p}_{i,t}), H_{i,t}; \mathcal{T}) \tag{7}$$

hence the updated node attributes are $\mathbf{V}'_{i,t} = \{\mathbf{u}_{i,t}\}$. The node-update function ϕ^v is parametrized by a Convolutional Neural Network with learnable parameters θ_{CNN} composed with an Action-MLP of parameters θ_{MLP}. The output of the Action MLP produces a categorical distribution over the set of admissible control inputs $\mathcal{U}$ from which the most probable action $\mathbf{u}_{i,t}$ is selected.

C. Properties of Graph Block Network

The graph topology is expected to vary over time owing to the mobility of the agents. Therefore, the Graph Block is required to perform consistently for all permutations of $\tilde{\mathcal{N}}_{i,t} \subseteq \mathcal{V}_c$, i.e. swapped order of robot indices, and regardless of the time shift. The permutation equivariance property for the aforementioned Graph Block network, directly limits the required sample complexity for training the network.

Proposition 1: (Permutation equivariance of Graph Block) Given the aforementioned graph $\mathcal{G}_t$, for any permutation $\pi : [n] \rightarrow [n]$ of $\mathcal{V}_c$, with corresponding permutation matrix P_π, and the Graph Block defined in subsection III-B, it holds that $\Phi(\pi(\mathcal{G}_t)) = \pi(\Phi(\mathcal{G}_t))$.

Proof: For the purposes of the proof, we denote $\hat{\boldsymbol{\Omega}}_t := \oplus_{i \in [N]} \hat{\Omega}_{i,t}^T$, $\boldsymbol{\Omega}_{\mathbf{t}} := \oplus_{i \in [N]} \boldsymbol{\Omega}_{i,t}^T$, $\mathbf{Z}_t := \oplus_{i \in [N]} Z_{i,t}^T$, $\hat{\mathbf{P}}_t := \oplus_{i \in [N]} \hat{\mathbf{p}}_{i,t}^T$, and $\mathbf{I}_t := \oplus_{i \in [N]} \mathcal{I}(\mathbf{p}_{i,t})^T$, where the operator $\oplus_i$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:17 UTC from IEEE Xplore. Restrictions apply.

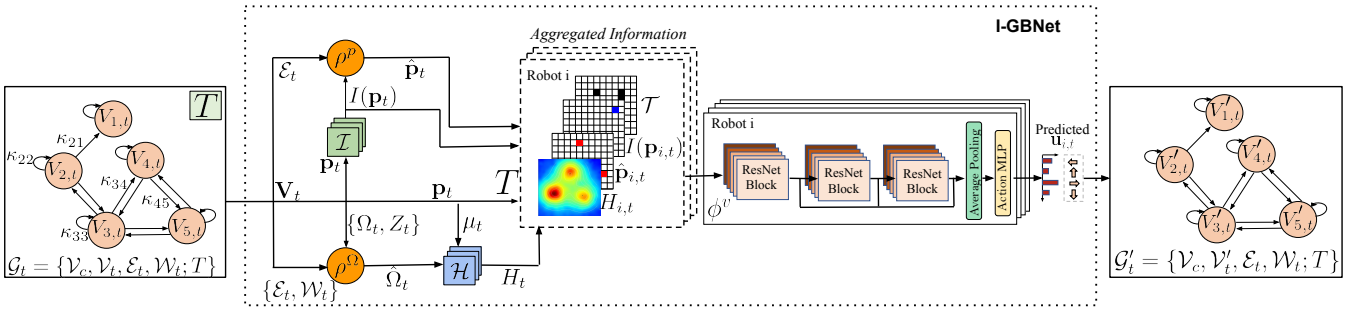


Fig. 2: I-GBNet structure comprising of aggregation functions ρ^{Ω}, ρ^{p}, a heatmap projection module $\mathcal{H}$ that projects the posterior distribution of the hidden state on the grid, and a node update function that produces admissible control actions. It only updates the node attributes of the graph, leaving the latter topologically unscathed. The update node function ϕ^{v} comprises of 3 consecutive ResNet Blocks with skip connections, an adaptive average pooling layer and an Action MLP.

denotes a concatenation of matrices along the first dimension. Let $\mathbb{A} \in \mathbb{R}^{N\times N}$ denote the adjacency matrix of the graph $\mathcal{G}_t$. The per-agent aggregation process, described by (5) and (6), can be reformulated as

$$\hat{\mathbf{\Omega}}_t = \left[\rho^{\Omega}(\mathbf{\Omega}_{\tilde{\mathcal{N}}_{1,t},t}, Z_{1,t}; \mathcal{W}_t)^T \dots \rho^{\Omega}(\mathbf{\Omega}_{\tilde{\mathcal{N}}_{N,t},t}, Z_{N,t}; \mathcal{W}_t)^T\right]^T = (\mathcal{W}_t \otimes \mathbb{I}_{d_x})\mathbf{\Omega_t} + \mathbf{Z}_t$$

$$\hat{\mathbf{P}}_t = \left[\rho^{p}(\mathbf{I}_{\tilde{\mathcal{N}}_{1,t},t})^T \dots \rho^{p}(\mathbf{I}_{\tilde{\mathcal{N}}_{N,t},t})^T\right]^T = ((\mathbb{A} - \mathbb{I}_N) \otimes \mathbb{I}_{\mathcal{T}_H})\mathbf{I}_t$$

Then, the permutation of the aggregated node attributes is the aggregation of the permuted node attributes as

$$\begin{bmatrix} \rho^{\Omega}(\mathbf{\Omega}_{\tilde{\mathcal{N}}_{\pi(1),t},t}, Z_{\pi(1),t}; \pi(\mathcal{W}_t)) \\ \vdots \\ \rho^{\Omega}(\mathbf{\Omega}_{\tilde{\mathcal{N}}_{\pi(N),t},t}, Z_{\pi(N),t}; \pi(\mathcal{W}_t)) \end{bmatrix} =$$

$$= (P_\pi \mathcal{W}_t P_\pi^T \otimes \mathbb{I}_{d_x})(P_\pi \otimes \mathbb{I}_{d_x})\mathbf{\Omega_t} + (P_\pi \otimes \mathbb{I}_{d_x})\mathbf{Z}_t$$
$$= (P_\pi \mathcal{W}_t P_\pi^T P_\pi \otimes \mathbb{I}_{d_x})\mathbf{\Omega_t} + (P_\pi \otimes \mathbb{I}_{d_x})\mathbf{Z}_t$$
$$= (P_\pi \otimes \mathbb{I}_{d_x})(\mathcal{W}_t \otimes \mathbb{I}_{d_x})\mathbf{\Omega_t} + (P_\pi \otimes \mathbb{I}_{d_x})\mathbf{Z}_t$$
$$= (P_\pi \otimes \mathbb{I}_{d_x})\left((\mathcal{W}_t \otimes \mathbb{I}_{d_x})\mathbf{\Omega_t} + \mathbf{Z}_t\right) = \pi\left(\hat{\mathbf{\Omega}}_t\right)$$

and similarly,

$$\left[\rho^{p}(\mathbf{I}_{\tilde{\mathcal{N}}_{\pi(1),t},t})^T \quad \dots \quad \rho^{p}(\mathbf{I}_{\tilde{\mathcal{N}}_{\pi(N),t},t})^T\right]^T$$
$$= (P_\pi(\mathbb{A} - \mathbb{I}_N)P_\pi^T \otimes \mathbb{I}_{\mathcal{T}_H})(P_\pi \otimes \mathbb{I}_{\mathcal{T}_H})\mathbf{I}_t$$
$$= (P_\pi \otimes \mathbb{I}_{\mathcal{T}_H})[((\mathbb{A} - \mathbb{I}_N) \otimes \mathbb{I}_{\mathcal{T}_H})\mathbf{I}_t] = \pi(\hat{\mathbf{P}}_t)$$

Since the update function is applied on a per–node basis on the aggregated permuted node attributes

$$\begin{bmatrix} \mathbf{V}'_{\pi(1),t} \\ \vdots \\ \mathbf{V}'_{\pi(N),t} \end{bmatrix} = \begin{bmatrix} \phi^{v}(\hat{\mathbf{p}}_{\pi(1),t}, \mathcal{I}(\mathbf{p}_{\pi(1),t}), H_{\pi(1),t}; \mathcal{T}) \\ \vdots \\ \phi^{v}(\hat{\mathbf{p}}_{\pi(N),t}, \mathcal{I}(\mathbf{p}_{\pi(N),t}), H_{\pi(N),t}; \mathcal{T}) \end{bmatrix}$$

$$= (P_\pi \otimes \mathbb{I}_u)\begin{bmatrix} \phi^{v}(\hat{\mathbf{p}}_{1,t}, \mathcal{I}(\mathbf{p}_{1,t}), H_{1,t}; \mathcal{T}) \\ \vdots \\ \phi^{v}(\hat{\mathbf{p}}_{N,t}, \mathcal{I}(\mathbf{p}_{N,t}), H_{N,t}; \mathcal{T}) \end{bmatrix} = \pi\left(\begin{bmatrix} \mathbf{V}'_{1,t} \\ \vdots \\ \mathbf{V}'_{N,t} \end{bmatrix}\right)$$

Thus, as the topology of the graph remains unchanged, we conclude $\Phi(\pi(\mathcal{G}_t)) = \pi(\Phi(\mathcal{G}_t))$. ■

***Proposition** 2:* (Time invariance of Graph Block) For $t_1, t_2 \in \mathbb{R}^+$, $t_1 \neq t_2$, given $\mathcal{G}_{t_1} = \{\mathcal{V}_c, \mathbf{V}_{t_1}, \mathcal{E}_{t_1}, \mathcal{W}_{t_1}; \mathbf{g}\}$, $\mathcal{G}_{t_2} = \{\mathcal{V}_c, \mathbf{V}_{t_2}, \mathcal{E}_{t_2}, \mathcal{W}_{t_2}; \mathbf{g}\}$ where $\mathbf{V}_{t_1} \equiv \mathbf{V}_{t_2}, \mathcal{E}_{t_1} \equiv \mathcal{E}_{t_2}, \mathcal{W}_{t_1} \equiv \mathcal{W}_{t_2}$, and the Graph Block Φ defined in subsection III-B, it holds that $\Phi(\mathcal{G}_{t_1}) = \Phi(\mathcal{G}_{t_2})$.

Proof: This proposition is satisfied naturally owing to the imitation learning method explained in the following Section and the fact that none of the components in Φ is an explicit function of time. The Graph Block network is trained to enable each robot to predict actions consistently and independently of t, given a graph $\mathcal{G}_t$, imitating expert decision making. ■

IV. Training and Architecture

A. Dataset and Training

For the dataset creation, we run the C-SB expert for 1500 randomly generated $20m \times 20m$ environments cluttered with a fixed number of obstacles, 10 robots with first-order dynamics and sensing radius of $2m$, static hidden state $\mathbf{x} \in \mathbb{R}^{20}$ and $\mathcal{U} = \{left, up, right, down\}$ by $0.5m$. The uncertainty threshold ϵ was fixed to $1e^{-1}$. For each instantiation we run entire episodes and randomly select 10 timesteps out of each to collect 15000 sets $\{(\mathbf{p}^k, Z^k, \boldsymbol{\mu}^k, \Omega^k, T^k), \boldsymbol{u}^k\}$. The latter sets are then transformed via the graph representation process (Section III-A) and result in our dataset $\mathcal{D} = \{\mathcal{G}^k, \boldsymbol{u}^k\}$. Given dataset $\mathcal{D}$ and the learnable parameters of the I-GBNet, θ_{CNN} and θ_{MLP}, we minimize the cross-entropy loss:

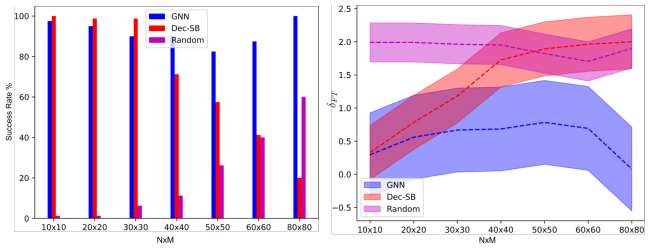
$$\min_{\theta_{\text{CNN}}, \theta_{\text{MLP}}} \left[\sum_{(\mathcal{G}^k, \mathbf{u}^k)\in\mathcal{D}} \mathcal{L}_{ce}(\Phi(\mathcal{G}^k), \mathbf{u}^k)\right]$$

For training, the dataset is divided into a training set (90%) and a validation set (10%). The training is conducted with the Adam optimizer and the learning rate is scheduled to decay from 10^{-4} to 10^{-6} within 20 epochs with cosine annealing. Finally, we trained with batch size 16 and weight decay equal to 10^{-5}.

B. Network Architecture

The architecture features a single-layer Graph Block, described in Section III-B, to provide the back bone of the distributed coordination scheme. We parametrize the learnable, node update function $\phi^{v}(\cdot)$, illustrated in Fig.2 as follows: Initially, a ResNet module implements feature extraction from the 4 channels and comprises of 3 sequential 2-layer (*Conv2d-BatchNorm2d-ReLU*) Residual Blocks with skip connections, followed by an adaptive average-pooling

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:17 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 3: In these figures we plot the success rate Fig. 3a and flowtime increase Fig. 3b for $N \times M$ robots and targets, where each problem instance is run for 100 random environments and initializations. We compare our GNN algorithm to two baselines a Decentralized Sampling-Based (Dec-SB) and a Random-Walker.

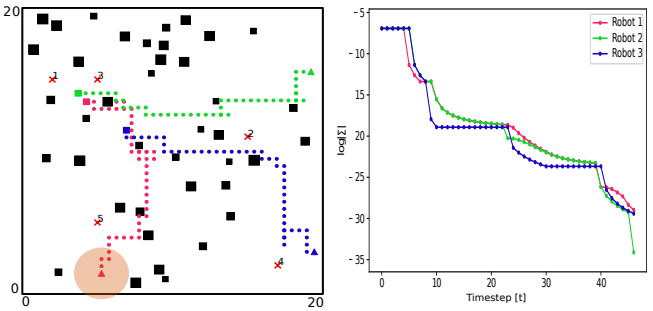
module. The latter provides the network with the flexibility to process environments of variable size by projecting to a fixed representation. The final layer is an MLP with parameters θ_{MLP}, that translates this representation into probabilities per control action.

V. Experiments

In this Section, we provide simulations to demonstrate the efficacy of our method. All case studies have been implemented using Python3 on a computer with a 3.2 GHz, 8-Core, Intel Core i11-800H CPU, 64GB RAM and an Nvidia RTX 3080Ti GPU, 16GB RAM. The experiments were conducted assuming that the robots can take noisy measurements with a camera that provides the xy-coordinates of any visible object within its sensing range, expressed in the robot's frame, where the sensing radius is selected to be equal to $2m$ and the noise covariance matrix $\mathbf{R}_{i,t} = (0.05 \odot \mathbf{x}_t)^2\mathbf{I}$. We assume first-order dynamics robots that select any action from the set $\mathcal{U} = \{\textit{left}, \textit{right}, \textit{up}, \textit{down}\}$ by $0.5m$ and the DKF parameters are chosen as $\kappa_{ii} = 0.75$ and $\kappa_{ij} = 0.25/|\mathcal{N}_i|$. We test our algorithm on the applications of target localization and tracking, where the hidden state $\mathbf{x}_t$ collects the positions of all targets at time t, i.e $\mathbf{x}_t = [\mathbf{x}_{1,t}^T, \dots, \mathbf{x}_{M,t}^T]^T$, where $\mathbf{x}_{k,t}$ is the position of target k at time t and M is the total number of targets. In this setting, we require to drop the uncertainty for all targets by at least one robot, i.e. $\det \Sigma_i^k \leq \epsilon$. To achieve collision avoidance when an invalid action is given, the robot randomly selects a collision-free action from $\mathcal{U}$, otherwise stays idle. A more sophisticated collision avoidance method [36]–[38] could be used instead. Unless stated differently, we assume a fully connected graph. It is important to stress that for all the experiments below we use the same network, trained as described in IV-A and without any fine-tuning per task.

A. Scalability and Generalization

We are interested in examining how our network, trained on $20m \times 20m$ (200×200 occupancy grid maps) with $N = 10$ robots and a hidden state $\mathbf{x}_t \in \mathbb{R}^{20}$ would perform on previously unseen $20m \times 20m$ environments and varying number of robots N and targets M. For this experiment, we run the Expert (C-SB) for each of the following instances $N \times M = [10 \times 10, 20 \times 20, 30 \times 30, 40 \times 40, 50 \times 50, 60 \times 60, 80 \times 80]$ for $n = 100$ random initializations



(a) (b)

Fig. 4: This Figure illustrates qualitatively the efficacy of our scheme to the target localization problem. Fig. 4a represents in different colors the executed trajectories of 3 robots equipped with a limited range onboard sensor in orange, tasked to localize 5 static targets. The initial and final positions of the robots are illustrated as a triangle and a square respectively, while the targets as red x-markers. The evolution of their corresponding uncertainties is depicted in Fig. 4b.

(obstacles, robots and target positions) and compare our GNN approach with two baselines (a) a Random-Walker and (b) a Distributed Sampling-Based (Dec-SB) AIA algorithm [17]. We evaluate the algorithms similar to [25] based on the following metrics:

1) Flowtime Increase $\delta_{FT} = (F - F^*)/F^*$, expresses the percentage change between the returned planning horizon F and the expert's horizon F^*. For each of the configurations, we collect n of the δ_{FT}'s and return as the flowtime increase of the configuration the mean.

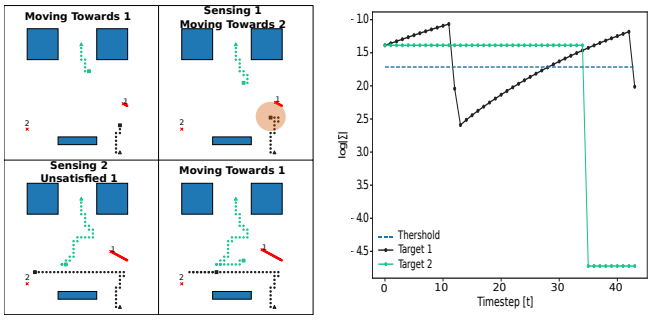
2) Success Rate $= n_{\text{success}}/n$, where n_{success} is the number of successful cases out of the n cases of the expert. A case is considered successful when a solution is acquired with a planning horizon $F \leq 3F^*$.

From Fig. 3a, we can see that the success rate of I-GBNet is equally well and occasionally better than the Dec-SB algorithm and always outperforms the Random Walker's. The Random-Walker performs better on more condensed environments (robots randomly initialized closer to targets) while the Dec-SB could not frequently return solutions with planning horizons less than $3F^*$. Even in the case of condensed environments where we expect smaller planning horizons returned by the expert, our method manages to complete the tasks withing the horizon bounds. In Fig. 3b, the mean of the Flowtime Increases of our method is systematically smaller than those of the baselines. We conclude that our network, trained only on environments of 10 targets and 10 robots is able to learn policies that generalize to novel environments with much larger graphs and hidden states. Moreover, mimicking a centralized expert is able to learn a distributed policy that is usually more effective than hand-designed distributed policies for the same problem.

B. Target Localization and Tracking

In this experiment, we are interested in validating our method qualitatively where the uncertainty threshold is set $\epsilon = 1.8e^{-2}$ for each of the targets. In the first scenario, three robots reside on a randomly-generated $20m \times 20m$ environment cluttered with a dense number of obstacles and are tasked to localize 5 static landmarks, while avoiding the

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:17 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 5: This Figure illustrates qualitatively the efficacy of our scheme to the target tracking problem. In Fig. 5a we show spotlights of the solution with a dynamic target where the initial and final positions of the robots are illustrated as a triangle and a square respectively, while the targets as red x-markers. The dynamic target's trajectory can be seen in red solid line. In Fig. 5b the evolution of the global covariance for each of the targets is depicted.

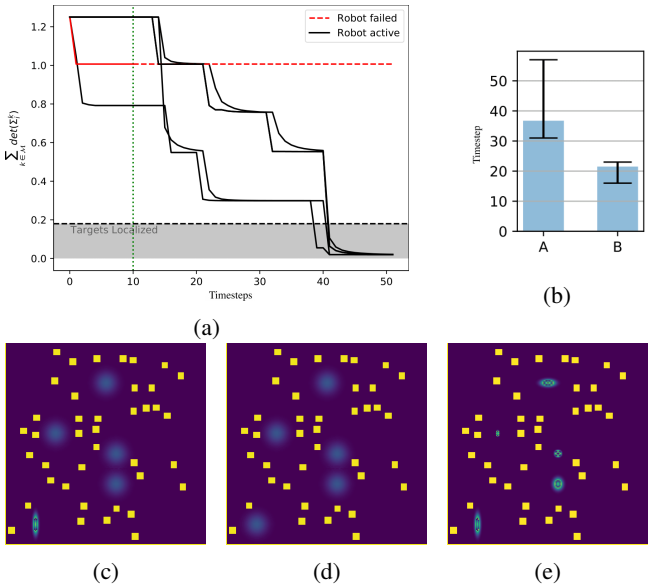
obstacles. From Fig. 4a, we can see that the robots executed successfully their task and localized the targets. In Fig. 4b, we draw the evolution of the determinants of the covariance matrices of the robots and certify that their uncertainties are decreasing with time. The robot depicted in green is the one that first achieves the desired ending condition.

In the second scenario, we present how our algorithm responds to dynamic targets. Specifically, two robots are tasked to localize a static landmark and a dynamic target of known dynamics; see Section II. In Fig. 5a we present spotlights of the online execution. From left-to-right, up-to-bottom the robots start moving towards target 1. Once they localize it, they start heading towards the next target. In the meantime, as the target moves, the increase in the uncertainty becomes evident on the robots' heatmap and thus return to re-localize it. The scope of Fig. 5b is to represent the global uncertainty for each of the targets defined at each timestep as the minimum of the determinants of the robots' covariance matrices. From the figure, it is clear that the robots successfully dropped both of the targets' uncertainty below the threshold, depicted as a dashed blue line.

The online execution of our proposed scheme allows for an online update of the estimated location $\boldsymbol{\mu}_t$ of the dynamic target when sensed, in contrast to the centralized approach where the estimated locations remain the same. This flexibility makes our algorithm more robust to poor prior estimates $\hat{\mathbf{x}}_0$ and eliminates the need for replanning like in the expert's case ([39, Section VI.C]), since the updated mean will be immediately reflected in the heatmap.

C. Robustness to Robot & Communication failure

This experiment examines the adaptability of I-GBNet to robot failure in target localization missions. In a $20m \times 20m$ obstacle riddled environment we assign $N = 5$ robots to localize $M = 5$ targets. The graph connectivity varies over time owing to the mobility of the agents for a given communication range $r_{com} = 4m$. We test for possible loss of packets during communication by introducing edge deletion based on a Poisson distribution ($\lambda = 1$). At time $t^{fail} = 10$, a randomly selected agent k exhibits catastrophic failure and



(a) (b)

(c) (d) (e)

Fig. 6: Figure 6a depicts the per-node sum of determinants over time, with black lines representing active robots, red the one that fails at $t = 10$, and gray area the localization threshold. Figures 6c, 6d and 6e illustrate the hidden state distribution at $t = 9$ for robots k and 2 and at time $t = 49$ for robot 2 respectively. The average mission time for 50 tests is illustrated in Figure 6b, where $\mathcal{A}$ refers to the case study with distance-defined communication graph, communication loss and agent failure and $\mathcal{B}$ the one with a static fully connected graph.

is permanently isolated from the swarm. Our GNN-based approach demonstrates online robustness to alterations in the synthesis of the swarm. Despite robot k failing, depriving the swarm of any information individually acquired through its sensing capabilities (see Figures 6c and 6d; bottom-left target has been localized only from agent k that is about to fail without propagating that information), the swarm re-deploys other robots to replace any acquired measurements from the failed robot not transmitted in the graph prior to failure, thus managing to localize all targets (see Figure 6e; the swarm has managed to localize the bottom-left target). All the active robots reach a precision consensus regarding the determinants of the covariance matrices, i.e. $\sum_{m \in \mathcal{M}} \det \Sigma_{i,t}^m$, below the required threshold for localization (see Fig. 6a). However, robot or communications failure appear to have prolonged the average mission time by 71.51% (see Fig.6b), as information propagation in the network is constantly impeded.

VI. Conclusions

In this paper, we reduce the multi-robot Active Information Acquisition problem into a learning framework over graphs and introduce I-GBNet, an AIA-inspired GNN, that offers distributed decision-making, imitating a centralized expert. In contrast to other works, our method can deal with dynamic phenomena and is robust to changes in the communication graph. Experiments demonstrated that despite being trained on small graphs, our network successfully generalizes to previously unseen environments/robot configurations and scales well. In the future, we shall investigate on more applications of Active Information Acquisition, like occupancy mapping and we shall explore the applicability of attention layers to further encourage scalability and time-efficiency.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:17 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419839596, 2019.

[2] K.-C. Ma, L. Liu, and G. S. Sukhatme, "Informative planning and online learning with sparse gaussian processes," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4292–4298.

[3] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *Pervasive Computing, IEEE*, vol. 3, pp. 24– 33, 11 2004.

[4] N. Bousias, S. Papatheodorou, M. Tzes, and A. Tzes, "Collaborative visual area coverage using aerial agents equipped with ptz-cameras under localization uncertainty," in *2019 European Control Conference (ECC)*, 2019.

[5] D. J. Bucci and P. K. Varshney, "Decentralized multi-target tracking in urban environments: Overview and challenges," in *2019 22th International Conference on Information Fusion (FUSION)*. IEEE, 2019, pp. 1–8.

[6] L. Carlone, J. Du, M. K. Ng, B. Bona, and M. Indri, "Active slam and exploration with particle filters using kullback-leibler divergence," *Journal of Intelligent & Robotic Systems*, vol. 75, no. 2, pp. 291–311, 2014.

[7] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 32–47, 2009.

[8] C. Guestrin, A. Krause, and A. P. Singh, "Near-optimal sensor placements in gaussian processes," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 265–272.

[9] P. Dames, M. Schwager, V. Kumar, and D. Rus, "A decentralized control policy for adaptive information gathering in hazardous environments," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 2807–2813.

[10] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, 2009.

[11] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1025–1032, 2018.

[12] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot slam," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4775–4782.

[13] D. Levine, B. Luders, and J. P. How, "Information-rich path planning with general constraints using rapidly-exploring random trees," in *AIAA Infotech at Aerospace Conference, Atlanta, GA*, 2010.

[14] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014.

[15] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Asymptotically optimal planning for non-myopic multi-robot information gathering." in *Robotics: Science and Systems*, 2019, pp. 22–26.

[16] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Dec-mcts: Decentralized planning for multi-robot active perception," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 316–337, 2019.

[17] M. Tzes, Y. Kantaros, and G. J. Pappas, "Distributed sampling-based planning for non-myopic active information gathering," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5872–5877.

[18] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[19] Y. Kantaros and G. J. Pappas, "Scalable active information acquisition for multi-robot systems," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7987–7993.

[20] H. Jeong, B. Schlotfeldt, H. Hassani, M. Morari, D. D. Lee, and G. J. Pappas, "Learning q-network for active information acquisition," *CoRR*, vol. abs/1910.10754, 2019. [Online]. Available: http://arxiv.org/abs/1910.10754

[21] S. Choudhury, A. Kapoor, G. Ranade, and D. Dey, "Learning to gather information via imitation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 908–915.

[22] J. Rückin, L. Jin, and M. Popović, "Adaptive informative path planning using deep reinforcement learning for uav-based active sensing," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4473–4479.

[23] G. Notomista, C. Pacchierotti, and P. R. Giordano, "Multi-robot persistent environmental monitoring based on constraint-driven execution of learned robot tasks," in *ICRA 2022-IEEE International Conference on Robotics and Automation*, 2022.

[24] H. He, P. Mineiro, and N. Karampatziakis, "Active information acquisition," *arXiv preprint arXiv:1602.02181*, 2016.

[25] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 785–11 792.

[26] Q. Li, W. Lin, Z. Liu, and A. Prorok, "Message-aware graph attention networks for large-scale multi-robot path planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5533–5540, 2021.

[27] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, "Learning decentralized controllers for robot swarms with graph neural networks," in *Conference on robot learning*. PMLR, 2020, pp. 671–682.

[28] E. Tolstaya, J. Paulos, V. Kumar, and A. Ribeiro, "Multi-robot coverage and exploration using spatial graph neural networks," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8944–8950.

[29] W. Gosrich, S. Mayya, R. Li, J. Paulos, M. Yim, A. Ribeiro, and V. Kumar, "Coverage control in multi-robot systems via graph neural networks," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8787–8793.

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

[31] L. Zhou, V. D. Sharma, Q. Li, A. Prorok, A. Ribeiro, P. Tokekar, and V. Kumar, "Graph neural networks for decentralized multi-robot target tracking," in *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2022, pp. 195–202.

[32] K. Ye, S. Dong, Q. Fan, H. Wang, L. Yi, F. Xia, J. Wang, and B. Chen, "Multi-robot active mapping via neural bipartite graph matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 839–14 848.

[33] N. Atanasov, R. Tron, V. M. Preciado, and G. J. Pappas, "Joint estimation and localization in sensor networks," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6875–6882.

[34] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6447–6454.

[35] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. F. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, Ç. Gülçehre, H. F. Song, A. J. Ballard, J. Gilmer, G. E. Dahl, A. Vaswani, K. R. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu, "Relational inductive biases, deep learning, and graph networks," *CoRR*, vol. abs/1806.01261, 2018. [Online]. Available: http://arxiv.org/abs/1806.01261

[36] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6252–6259.

[37] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed autonomous robotic systems*. Springer, 2013, pp. 203–216.

[38] J. Alonso-Mora, P. Beardsley, and R. Siegwart, "Cooperative collision avoidance for nonholonomic robots," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 404–420, 2018.

[39] M. Tzes, V. Vasilopoulos, Y. Kantaros, and G. J. Pappas, "Reactive informative planning for mobile manipulation tasks under sensing and environmental uncertainty," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 7320–7326.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:17 UTC from IEEE Xplore. Restrictions apply.