2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Learning Load-Balanced Distributed Coverage for Robot Swarms via Graph Attention Networks

Yun Gao*, Hao Gao*, Wenzong Ma, Hui Xiong, and Yiding Ji

***Abstract*— Coverage control for dynamic targets remains challenging in multi-robot systems due to limited communication, workload imbalance, and the lack of scalable decentralized strategies. In this paper, we propose a hybrid model-based and learning-driven framework that enables distributed coverage with load balancing under local communication constraints. We first derive a centroidal Voronoi tessellation (CVT)-based controller that explicitly incorporates load density regulation to balance resource consumption among robots. To eliminate the reliance on global target information, we embed key control variables as node features and employ a graph attention network (GAT) to learn decentralized coordination policies through multi-hop neighbor interactions. An actor-critic training scheme further refines the policy to maximize coverage performance while preserving network connectivity. Simulations demonstrate superior coverage efficiency and connectivity maintenance compared with Lloyd and GNN-based baselines, together with strong generalization across varying sensing ranges and swarm sizes. Real-world experiments validate the sim-to-real transferability of the proposed framework.**

***Index Terms*— Robot swarms, coverage control, decentralized coordination, graph attention networks, reinforcement learning.**

I. Introduction

Robot swarms (RSs) have recently emerged as ubiquitous and promising for accomplishing complex and large-scale tasks in dynamic and uncertain environments. One notable topic is coverage control, which originates from sensor networks [1] and has recently attracted significant attention in RSs due to its wide applications, such as environmental monitoring, disaster search and rescue, and infrastructure inspection, to name a few. The primary objective of coverage control is to allocate robots over a spatial domain to optimize a task-dependent coverage objective, typically defined as a spatially weighted sensing or service performance functional over the environment. Despite substantial progress, achieving distributed coverage in realistic RS deployments remains challenging due to limited communication ranges, heterogeneous target importance, and finite onboard resources. In particular, coverage strategies must simultaneously account for robotic sensing capabilities [2], load balancing [3], and scalability in large-scale deployment [4].

Some coverage control methods leverage Voronoi tessellation [5], artificial potential fields, probabilistic models, etc. Among them, the Voronoi tessellation method has demonstrated significant advantages for efficient spatial partitioning of the environment [6]. The conventional Lloyd algorithm suffers from a lack of explicit spatial partitioning, which may cause robots to converge toward highly weighted targets while neglecting others. The centroidal Voronoi tessellation (CVT) method addresses this issue by integrating spatial partitioning into the optimization process, inspiring numerous variants within the CVT framework [7]–[9].

Despite these advantages, many CVT-based coverage methods heavily rely on an idealistic assumption of CVT implementation, such as the global broadcast of target information to all robots [10]. When tasks are distributed in nature, such a framework fails to achieve distributed control laws and is not preferred. Furthermore, the sharing of global information among all robots incurs substantial resource consumption and subsequently increases the deployment cost, severely restricting the scalability of the method and making it impractical to deploy RSs on a large scale [11].

Another critical yet insufficiently formalized factor in RSs coverage control is load balancing, where "load" refers to the consumption of limited onboard resources such as battery energy, sensing operation time, computation budget, or actuation effort. Most existing studies assume there is an infinite amount of resources for the robots to take actions for coverage [12], whereas it is finite. The varying importance of different targets within a robot's current Voronoi cell leads to different execution efficiencies over time [13]. Robots with identical actuators may consume energy and computational resources at different rates due to heterogeneous target distributions within their Voronoi cells. If the load balancing issue among robots is ignored, unrealistic scenarios will arise where either certain robots continue to operate after their resources are depleted or the swarm runs out of resources before the target is covered.

In recent years, reinforcement learning and deep learning based control methods have provided promising directions to address these challenges [14]. However, they often struggle to adapt to the graph structure of RSs, thus exhibiting limitations such as poor adaptability, slow convergence, and

Yun Gao is with the Robotics and Autonomous Systems Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, and with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China (e-mail: y.gao@gaoyunailab.com). Hao Gao and Yiding Ji are with the Robotics and Autonomous Systems Thrust, Wenzong Ma and Hui Xiong are with the Artificial Intelligence Thrust, the Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China (e-mail: ghalfred39@gmail.com; wma423@connect.hkust-gz.edu.cn; xionghui@hkust-gz.edu.cn; jiyiding@hkust-gz.edu.cn). (Corresponding author: Yiding Ji)

*The authors contributed equally to this work.

This work is supported in part by National Natural Science Foundation of China grants 62303389 and 62373289; Guangdong Basic and Applied Basic Research Funding grant 2024A1515012586; Guangdong Scientific Research Platform and Project Scheme grant 2024KTSCX039; Youth Talent Support Program of Guangdong Association for Science and Technology grant SKXRC2025463 and Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No.2023B1212010007).

performance degradation in complex scenarios [15]. Graph neural networks (GNNs) have been incorporated into coverage control to overcome these obstacles since they are inherently suited to capture the intricate relations among robots in a network [16], [17]. One representative framework is to utilize graph convolutional networks (GCNs) to enhance the information exchange between robots [18]. However, GCNs typically aggregate information from neighboring nodes using fixed, normalized weights, implicitly assuming that all neighbors exert equal influence on the central node. This hinders the adaptability of GCN methods to changing tasks and environments, particularly when multiple robotic features have varying levels of importance.

To address the aforementioned challenges, we propose a hybrid framework combining CVT and a graph attention network (GAT) for load-balancing distributed coverage under communication constraints. The framework integrates model-based CVT control with graph attention-based policy learning in an actor–critic architecture. The main contributions of this work are summarized as follows:

- We develop a distributed CVT-based coverage control framework that explicitly incorporates load density feedback, enabling load-balanced distributed coverage with bounded communication radius.
- We propose a hybrid model-based and learning-based architecture that integrates structured CVT spatial partitioning with graph attention-based policy learning, forming a decentralized control strategy.
- We embed physically meaningful control variables into a GAT representation, enabling adaptive multi-hop information aggregation without global broadcast while preserving interpretability of the control structure.
- We design an actor–critic training scheme that jointly optimizes coverage performance and network connectivity, and systematically decode learned graph embeddings into executable control commands.
- Extensive simulations and real-world robot experiments validate the scalability, robustness, and deployability.

Compared with learning-based swarm coverage approaches that rely on end-to-end policy optimization, our framework preserves an explicit CVT backbone, providing structured spatial partitioning and analytically interpretable load-balancing feedback. In contrast to Voronoi-based methods defined on complex environments, which focus on spatial partitioning, our framework further integrates decentralized learning to adaptively optimize coverage performance under communication constraints and resource limitations.

The remainder of the paper is organized as follows. Section II introduces the RSs model, formulates the coverage control problem, and defines the load balancing condition. Section III develops the GAT-based coverage control framework and an actor-critic method to train the GAT. Section IV includes numerical simulation and experiment results to show the actual performance of our approach. Finally, Section V concludes the paper and lists future research directions.

II. Preliminaries and Problem Formulation

In this section, we first provide an intuitive description of the considered coverage scenario. A group of robots operates in a bounded environment containing multiple dynamic targets that generate spatial information density. The objective is to distribute the robots such that they maximize the reduction of this density field while maintaining balanced workload and limited communication constraints.

We then present the mathematical model of the RSs, introduce the dynamic target coverage problem, and review canonical coverage approaches in the literature. Finally, we propose the concept of load balancing and formulate the coverage control problem under load balancing requirements.

A. Robot Swarms

Consider a team of $I \in \mathbb{Z}_+$ robots that navigate in a compact and convex two-dimensional plane $\mathcal{Q} \subset \mathbb{R}^2$. Let $k \in \mathbb{Z}_{\geq 0}$ denote the discrete control time index. At time k, the RS position configuration is given by the vector $P^{\mathrm{r}}(k) = [p_1^{\mathrm{r}}(k), \ldots, p_I^{\mathrm{r}}(k)] \in \mathbb{R}^{2\times I}$. The movement of each robot is governed by the kinematic model (1).

$$p_i^{\mathrm{r}}(k+1) = p_i^{\mathrm{r}}(k) + T_s u_i(k), \tag{1}$$

where T_s is the sampling interval, $u_i(k) \in \mathbb{R}^2$ is the control input bounded within $(v_{\min}, v_{\max})$, with $v_{\max}$ and $v_{\min}$ being the maximum and minimum allowed linear velocities.

Each robot aims to perceive and interact with targets while communicating with fellow robots in the task environment. One robot can establish direct communication links with other robots if they lie within a circular region of radius $R_c \in \mathbb{R}_+$. Let $d_{ij}(k) = \|p_i^{\mathrm{r}}(k) - p_j^{\mathrm{r}}(k)\|$ represent the Euclidean distance between the i^{th} and j^{th} robot, then the above limitation can be expressed as $d_{ij}(k) \leq R_c$. We place robots that meet these conditions into a set $\mathcal{N}_i^1(k)$, which is referred to as the i^{th} robot's 1-hop neighbor set.

Each robot interacts with targets within a radius R_a, forming a disk-shaped coverage area:

$$C_a(k) = \{q \in \mathcal{Q} \mid \|q - p_i(k)\| \leq R_a\}. \tag{2}$$

The following function describes the impact of this interaction on the target:

$$f(q, p_i^{\mathrm{r}}(k)) = \begin{cases} \beta e^{-\lambda \|q - p_i^{\mathrm{r}}(k)\|^2} - \zeta & q \in C_a(k), \\ 0 & q \notin C_a(k), \end{cases} \tag{3}$$

where $q \in \mathcal{Q}$ is a point in the task environment, $\beta \in (0,1)$ is the maximum affect coefficient, $\lambda \in (0, \infty)$ is the attenuation coefficient, and $\zeta = \beta e^{-\lambda R_a^2}$.

B. Dynamic Targets

There are $L(k) \in \mathbb{Z}_+$ dynamic targets randomly moving within $\mathcal{Q}$, and let $p_l^{\mathrm{t}}(k) \in \mathbb{R}^2$ denote the location of the l^{th} target at time k. The attributes of each target can be regarded as an information source, described by Gaussian function:

$$g(q, p_l^{\mathrm{t}}(k)) = \mu_l \exp(-\frac{1}{2\varrho_l^2} \left\|q - p_l^{\mathrm{t}}(k)\right\|^2), \tag{4}$$

where ϱ_l is the speed of information attenuation that determines the sensing range, and μ_l is the peak value. This

allows us to view $\mathcal{Q}$ as a non-uniform information field and compute the information density at any q as follows:

$$\phi(q,k)=\sum_{l=1}^{L(k)} g(q,p_l^{\mathrm{t}}(k)). \tag{5}$$

C. Problem Formulation

In this section, we formally define the distributed coverage problem considered in this paper. From a qualitative perspective, the objective is to deploy a group of robots to cooperatively cover a set of spatial targets with heterogeneous importance, while respecting limited communication ranges and ensuring balanced resource utilization across robots.

Consider an information density field defined over $\mathcal{Q}$, where due to robot–target interactions, the information density at location q evolves according to:

$$\phi(q,k+1)=e^{\sum_i f(q,p_i^{\mathrm{r}}(k))}\phi(q,k). \tag{6}$$

Our objective is to maximize the spatial reduction of information density, which can be interpreted as maximizing $|\phi(q,k+1)-\phi(q,k)|$, leading to Problem 1 [1].

Problem 1 (Coverage control)**.** *$L(k)$ dynamic targets are randomly moving in a convex and compact region $\mathcal{Q}\subset\mathbb{R}^2$. These targets have constantly changing locations $p_l^{\mathrm{t}}(k)$, attenuation parameters ϱ_l, and peak value μ_l, rendering $\mathcal{Q}$ an uneven information density field. Then consider a team of I robots in $\mathcal{Q}$, with limited communication range R_c, interaction range R_a and interaction capability defined in* (3)*. They start from $P^{\mathrm{r}}(0)$ to find the optimal location configuration to maximize the interactive impact of the robots on the targets, that is, maximize the following cost funcion:*

$$\mathcal{H}(P^{\mathrm{r}}(k),V(k),k)=\sum_{i=1}^{I}\mathcal{H}_i(p_i^{\mathrm{r}}(k),V_i(k),k), \tag{7}$$

where

$$\mathcal{H}_i(p_i^{\mathrm{r}}(k),V_i(k),k)=\int_{V_i(k)} f(q,p_i^{\mathrm{r}}(k))\phi(q,k)dq, \tag{8}$$

and $V(k)$ refers to the Voronoi tessellation [1] define as:

$$V_i(k)=\{q\in\mathcal{Q}\mid \|q-p_i^{\mathrm{r}}(k)\|\le\left\|q-p_j^{\mathrm{r}}(k)\right\|,\forall j\neq i\}. \tag{9}$$

A gradient-based approach is applied to maximize $\mathcal{H}$, while simultaneously considering the limited interaction range. This leads to the following nominal control law:

$$u_i^{\mathrm{c}}(k)=-2\kappa M_{U_i}(k)(p_i^{\mathrm{r}}(k)-C_{U_i}(k)), \tag{10}$$

where κ is position control gain, $U_i(k)=V_i(k)\cap C_a$ denotes the constrained Voronoi cell, $M_{U_i}(k)$ and $C_{U_i}(k)\in\mathbb{R}^2$ respectively represent the mass and centroid within $U_i(k)$.

$$M_{U_i}(k)=\int_{U_i} 2\beta\lambda e^{-\lambda\|q-p_i^{\mathrm{r}}(k)\|^2}\phi(q,k)dq, \tag{11}$$

$$C_{U_i}(k)=\frac{1}{M_{U_i}(k)}\int_{U_i} 2\beta\lambda e^{-\lambda\|q-p_i^{\mathrm{r}}(k)\|^2}\phi(q,k)qdq. \tag{12}$$

It is important to note that the problem is not fully resolved in this formulation. While the control strategy (10) of each robot is computed in a distributed manner, the protocol relies on shared target information, effectively assuming global information accessibility. This assumption introduces implicit centralized dependencies, thereby undermining the decentralization that the algorithm ostensibly aims to achieve.

Furthermore, we define $e_i(k)=p_i^{\mathrm{r}}(k)-C_{U_i}(k)\in\mathbb{R}^2$ as the position tracking error and $\sigma_i(k)\in\mathbb{R}_+$ as the load density for the i^{th} robot which is calculated by (13).

$$\sigma_i(k)=\frac{M_{U_i}(k)}{A_{U_i}(k)}, \tag{13}$$

where $A_{U_i}(k)=\int_{U_i(k)}dq$ denotes the area of $U_i(k)$.

The load represents normalized resource utilization, which may correspond to energy consumption, sensing workload, or task execution effort. Obviously, we cannot guarantee that $\sigma_i(k)=\sigma_j(k)$ for $\forall j\neq i$ at time k, which will lead to an imbalance in resource consumption. Thus, we define the average load density $\bar{\sigma}(k)$ at time k as calculated by (14).

$$\bar{\sigma}(k)=\frac{\sum_{i=1}^{I}M_{U_i}(k)}{\sum_{i=1}^{I}A_{U_i}(k)}. \tag{14}$$

Although the ideal load average $\bar{\sigma}(k)$ is defined as a global quantity for analytical clarity, it is not assumed to be directly accessible. In practice, each robot maintains a local estimate obtained through 1-hop neighbors' information aggregation over the communication graph, which is implicitly facilitated by the GAT-based learning mechanism introduced later.

Definition 1 (Load balancing)**.** *During the execution of the coverage task, the i^{th} robot ensures that $\lim_{k\to+\infty}|\sigma_i(k)-\bar{\sigma}(k)|=0$ is maintained.*

To this end, we extend λ in (3) from a globally constant to a time-varying heterogeneous parameter $\lambda_i(k)$ and adjust it to change the load density of each robot. We also set an acceptable load density deviation $\epsilon>0$. That is to say, when $|\sigma_i(k)-\bar{\sigma}(k)|\le\epsilon$, the robot remains at the centroid of the Voronoi cell. Otherwise, the robot will be driven to adjust (3) according to the following control strategy:

$$\lambda_i(k)=\begin{cases}\lambda_0 e^{-\xi\Delta_i(k)} & \Delta_i(k)\ge\epsilon,\\ \lambda_0 & \text{otherwise},\end{cases} \tag{15}$$

where λ_0 is the reference attenuation value, $\xi>0$ is the load density control gain, and $\Delta_i(k)=|\sigma_i(k)-\bar{\sigma}(k)|$.

Assumption 1 (Attenuation boundedness)**.** *Suppose the CVT guarantees convergence of $p_i^{\mathrm{r}}(k)$ to the centroid under fixed λ. Then the adaptive update* (15) *ensures bounded $\lambda_i(k)$ and preserves convergence provided ξ is sufficiently small.*

To enable load-aware coverage behaviors, we consider a more general control input $\{u_i(k)\}_{i=1}^{I}$, which will be addressed through the methods proposed later.

Problem 2 (Coverage control with load balancing)**.** *Consider I robots tasked with covering $L(k)$ dynamic targets, as described in Problem 1. Our objective is to design a decentralized control strategy $\{u_i(k)\}_{i=1}^{I}$ to achieve a maximal cost function* (7)*, while satisfying the load balancing requirement outlined in Definition 1. The policy must rely solely on local information obtained through communication with each robot's 1-hop neighbors.*

III. Learning based Coverage Control

In this section, we propose a GAT-based framework to address Problem 2. The approach integrates model-based CVT partitioning with a graph attention policy that enables adaptive information aggregation over limited communication graphs. A decentralized actor–critic scheme optimizes the residual policy to improve coverage and load balancing while respecting connectivity constraints. Unlike purely learning-based methods, the framework preserves the geometric structure of CVT, ensuring interpretability.

A. Graph Attention Networks for Coverage Tasks

To address the decentralized coordination requirement in Problem 2, we adopt a GAT to enable adaptive information aggregation over the time-varying communication graph. Unlike GNNs with fixed normalized weights, GAT assigns state-dependent attention coefficients, allowing each robot to prioritize neighbors according to real-time coverage and load conditions [19]. This adaptability is essential under heterogeneous target distributions and load imbalances.

To preserve the structure induced by the CVT-based controller, we construct the node feature vector for each robot:

$$h_i(k) = [e_i(k), M_{U_i}(k), \Delta_i(k)]^T. \tag{16}$$

The components of $h_i(k) \in \mathbb{R}^4$ are designed to be locally computable while capturing key aspects of the coverage task: the tracking error $e_i(k)$ captures spatial convergence behavior and is derived from the CVT-based reference position; the local mass $M_{U_i}(k)$ characterizes local task density and is computed from the locally constructed Voronoi cell using neighbor position exchange; and the load deviation $\Delta_i(k)$ explicitly encodes resource imbalance, estimated based on locally available load information. This feature design forms a structured interface between the model-based CVT dynamics and the learning-based policy, introducing physics-informed inductive bias instead of purely latent embeddings.

The attention mechanism operates on the nodes after they are linearly transformed by a matrix $W_{\mathrm{g}} \in \mathbb{R}^{4\times4}$. The layer then computes the attention coefficients α_{ij} according to (17) , which quantify the significance features of robot j to i.

$$\alpha_{ij} = \underset{j\in\mathcal{N}_i^1(k)}{\mathrm{softmax}}(\mathrm{LeakyReLU}(d^T[W_{\mathrm{g}}h_i(k) \| W_{\mathrm{g}}h_j(k)])), \tag{17}$$

where $\|$ is the concatenation operation and d is attention mechanism vector. Through this mechanism, robots with larger load deviations or higher task density exert stronger influence on neighboring policies, enabling decentralized compensation of imbalance without global coordination.

To improve stability and capture diverse relational patterns, we employ a multi-head attention scheme:

$$h_i'(k) = \Big\|_{n=1}^{N} \mathrm{ELU}\Big(\sum_{j\in\mathcal{N}_i^1(k)} \alpha_{ij}^{(n)} W_{\mathrm{g}}^{(n)} h_j(k)\Big) \tag{18}$$

where $h_i'(k)$ represents the output vector, N denotes the number of attention heads, $\alpha_{ij}^{(n)}$ are normalized attention coefficients computed by the n^{th} attention head, and $Z^{(n)}$ is the corresponding weight matrix of input linear transformation. The aggregated representation $h_i'(k)$ is then fed into the decentralized actor network to generate control actions that refine the CVT-based motion toward improved coverage and load balancing. Meanwhile, unlike generic GAT implementations, the proposed feature construction embeds physically meaningful CVT variables, introducing inductive bias that improves sample efficiency and interpretability.

B. Actor-Critic Learning

To refine the CVT-based controller, we adopt an actor–critic framework that learns a residual control policy on top of the model-based motion [20]. Unlike standard formulations, the policy operates on graph-structured embeddings generated by the GAT encoder and outputs decentralized motion corrections that improve coverage efficiency, load consistency, and connectivity preservation.

Unlike standard actor–critic formulations operating on independent observations, the policy gradient here is conditioned on graph-structured embeddings that encode time-varying neighborhood interactions. This, together with residual control learning over CVT dynamics and reward shaping incorporating load deviation and connectivity penalties, constitutes the key distinction of the framework.

At each time step k, the i^{th} robot observes the encoded local state $h_i'(k)$ produced by the GAT. The actor generates a continuous residual action $a_i(k) \in \mathbb{R}^2$, representing a corrective velocity added to the CVT reference motion [21]:

$$u_i(k) = u_i^{\mathrm{c}}(k) + a_i(k). \tag{19}$$

To learn such residual corrections, we parameterize the actor as a policy network $\pi_\theta(a_i(k) \mid h_i'(k))$ (hereafter π_θ) with parameters θ, optimized via policy gradient. The update follows the advantage-based equation:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta A(h_i'(k), a_i(k))], \tag{20}$$

with ∇ the gradient operator, $\mathbb{E}$ the expectation operator, $A(h_i'(k), a_i(k))$ the advantage function approximated using the temporal-difference (TD) error produced by the critic. The objective function $J(\pi_\theta)$ represents the expected cumulative reward, serving as a comprehensive performance metric that guides the policy network to simultaneously maximize spatial coverage efficiency, minimize inter-robot workload imbalances, and ensure the robust preservation of the communication network topology.

The critic evaluates state–action pairs using graph-aware aggregation with parameter φ. For each time k, a joint embedding $\psi_i(k)$ is computed as:

$$\psi_i(k) = \mathrm{LeakyReLU}(W_{\mathrm{c}}[h_i'(k) \| a_i(k)] + b_c), \tag{21}$$

where W_{c} and b_c are learnable weight matrix and bias. This embedding is averaged over the 1-hop neighborhood:

$$\bar{\psi}_i(k) = \frac{1}{|\mathcal{N}_i^1(k)| + 1}(\psi_i(k) + \sum_{j\in\mathcal{N}_i^1(k)} \psi_j(k)). \tag{22}$$

This aggregation improves robustness of value estimation and mitigates variance caused by local observation noise in dynamic graphs. Based on this representation, the critic produces the Q-value estimate:

$$Q_\varphi(h_i'(k), a_i(k)) = z^T \bar{\psi}_i(k), \tag{23}$$

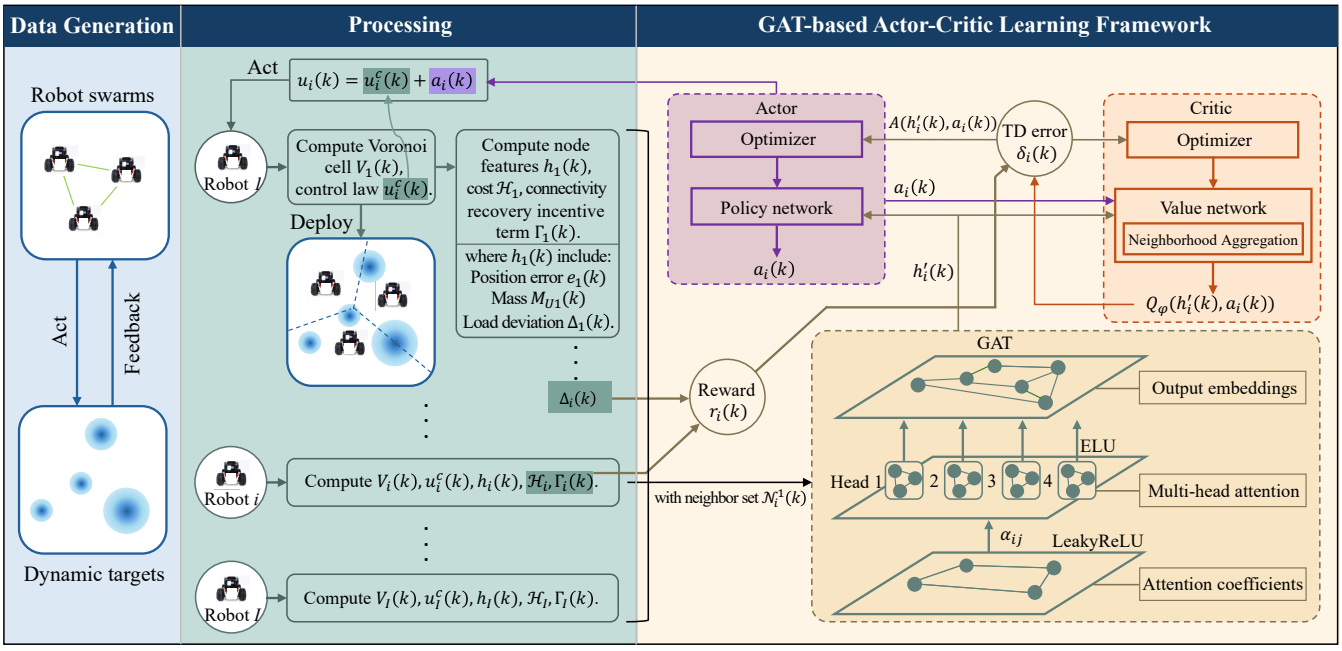


Fig. 1: Architecture of the GAT-based actor–critic learning framework for decentralized coverage. Task-informed node features derived from CVT dynamics are encoded through multi-layer graph attention. The actor outputs residual velocity corrections added to the nominal CVT controller, while the critic evaluates performance based on coverage, load balancing, and connectivity-aware rewards.

where z denotes the trainable weight vector of the critic's output layer. To enforce Bellman consistency, we compute the TD error $\delta_i(k)$, which measures the discrepancy between the current estimate and the updated target value:

$$\delta_i(k) = r_i(k) + \gamma Q_\varphi(h_i'(k+1), a_i(k+1)) - Q_\varphi(h_i'(k), a_i(k)), \quad (24)$$

where $\gamma \in (0,1)$ is the discount factor. Thus, we use the TD error as an unbiased estimator of the advantage, *i.e.*, $A(h_i'(k), a_i(k)) \approx \delta_i(k)$. The reward function $r_i(k)$ is designed to optimize a composite objective comprising coverage performance, load consistency, and communication connectivity maintenance, and is defined as:

$$r_i(k) = w_c \mathcal{H}_i(p_i^{\mathrm{r}}(k), V_i(k), k) - w_l \Delta_i(k) - w_n \Gamma_i(k), \quad (25)$$

where $w_c, w_l, w_n > 0$ are the scaling weights for coverage, load balancing, and connectivity, respectively. The term $\Gamma_i(k) = \sum_{j \in \mathcal{N}_i^1(k)} \Psi(d_{ij}(k))$, where $\Psi(\cdot)$ is computed as in (26), serves as a connectivity recovery incentive. It is designed as a potential-like barrier and aims to address robust decentralized coordination by both preserving existing communication links and providing a steep gradient for restoring broken connections.

$$\Psi(x) = \begin{cases} 0, & \text{if } x \le R_c - \epsilon \\ -\Psi_0, & \text{if } x > R_c \\ -\vartheta e^{(x - R_c)}, & \text{otherwise} \end{cases} \quad (26)$$

where ϵ is a communication safety margin, Ψ_0 is a large constant penalty for disconnection, and ϑ denotes sensitivity.

The critic parameters are updated by minimizing the mean squared TD error, thereby enforcing consistency with the Bellman equation $\mathcal{L} = \mathbb{E}_{\pi_\theta}[\delta_i(k)^2]$.

C. Learning based Coverage Control Architecture

The proposed framework integrates CVT-based coverage control with a GAT-enhanced actor–critic architecture. As illustrated in Fig. 1, the data flow proceeds through three stages: state construction, graph-based representation learning, and residual policy optimization.

All learning-based methods operate on the same task-informed node feature vector (16), constructed locally from each robot's Voronoi cell and 1-hop neighborhood. Synthetic terrain maps with spatially distributed peaks are generated offline to produce diverse load distributions, while robots are initialized in collision-free annular formations under time-varying communication graphs.

Given node features $\{h_i(k)\}_{i=1}^I$ and the adjacency matrix, a three-layer multi-head GAT produces embeddings $h_i'(k)$ that encode both geometric CVT errors and neighborhood load imbalance. The GAT does not replace the coverage controller; rather, it reshapes the state representation to expose relational load structure to the learning module.

On top of the nominal controller (10), a learnable residual action (19) is introduced. The actor is a two-layer MLP with *LeakyReLU* mapping $h_i'(k)$ to a 2D residual velocity. The critic is an independent two-layer MLP approximating the state–action value $Q_\varphi(h_i'(k), a_i(k))$, with a linear readout layer as defined in (23).

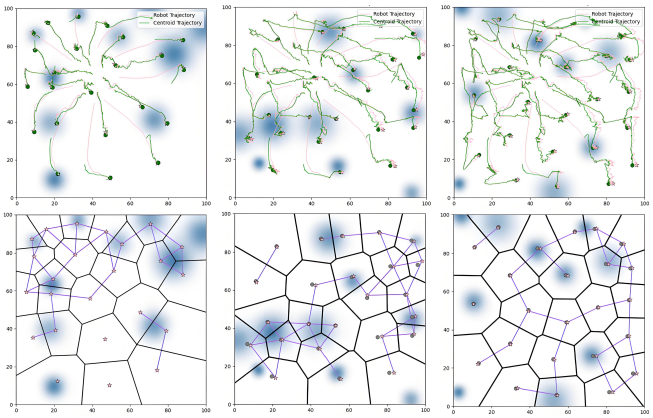
The critic evaluates load-balancing performance via temporal-difference learning, while the actor updates parameters through policy gradients. By restricting learning to residual load redistribution and preserving CVT convergence guarantees, this architecture achieves graph-aware coverage rather than end-to-end replacement of nominal controller.

IV. Evaluation

A. Numerical Simulation

We evaluate the proposed framework in a square workspace $\mathcal{Q} = [0, 100] \times [0, 100]$ with a time-varying density distribution defined in (5). The density parameters $\mu_l(k)$ vary within $[0, 1]$ and $\varrho_l(k)$ vary within $[2, 4]$, yielding dynamic targets requiring adaptive redistribution of sensing resources. Each robot has interaction radius $R_a = 8$ and communication radius $R_c = 12$, ensuring connectivity preservation remains nontrivial during coverage expansion. We terminate each episode at a final time $k = 100$ with sampling time $T_s = 1\ s$. For comparison, all learning-based baselines share identical actor–critic architectures and reward structures, differing only in the graph aggregation mechanism. The Lloyd algorithm serves as a non-learning baseline without explicit connectivity constraints. To ensure statistical reliability, all results are averaged over 100 independent runs with randomized initial RS configurations.

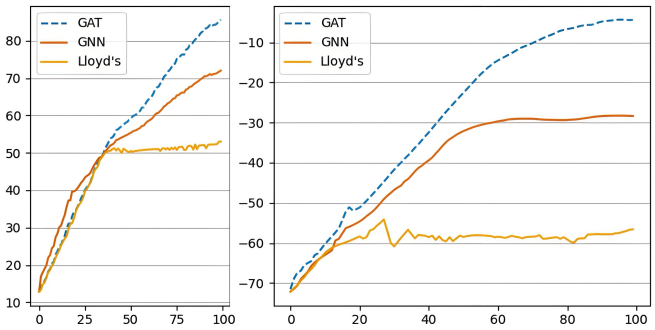
Experiment 1 (Benchmark performance): To systematically assess the effectiveness of the proposed strategy, we evaluate coverage efficiency and connectivity preservation. Fig. 2–Fig. 4 illustrate these aspects, including spatial deployment patterns, coverage performance, and network connectivity evolution. To provide spatial intuition, Fig. 2 presents a simulation snapshot. Robots distribute around targets while preserving sufficient inter-robot proximity for communication maintenance. Compared to the patterns produced by Lloyd's algorithm, the learned formation under GAT appears task-adaptive and importance-driven rather than purely centroid-based.



(a) k = 30 (b) k = 60 (c) k = 90

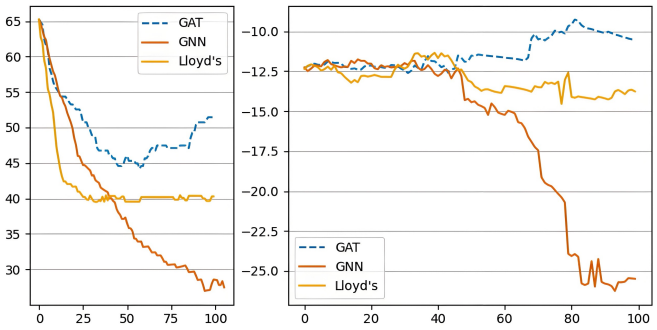
Fig. 2: Spatial distribution induced by the proposed strategy. The snapshot illustrates the learned importance-driven allocation of 25 robots in a 100×100 environment.

As illustrated in Fig. 3 (a), the coverage percentage over time reveals a statistically consistent advantage of the proposed strategy. Unlike GNN and Lloyd's algorithm [1], which exhibit slower early-stage expansion and premature saturation, GAT achieves faster initial growth and higher asymptotic coverage. By the end of the episode, the coverage percentage of our method is more than 90%, while GNN stabilizes around 72%, and Lloyd's method converges to 52%. Notably, the performance gap widens over time rather than emerging only at convergence, indicating that attention-based neighbor weighting improves the exploration–coordination balance during the full trajectory evolution. Fig. 3 (b) presents the coverage reward evolution. The proposed strategy converges to an average reward of -4.4, substantially outperforming GNN (-28.4) and Lloyd's method (-56.6). Since the reward penalizes uncovered targets, this quantitative difference confirms that proposed strategy not only spreads robots spatially but also prioritizes high-importance areas more effectively. The improvement can be attributed to the adaptive attention mechanism, which modulates inter-robot influence dynamically and prevents redundant clustering in already well-covered regions.



(a) Coverage percentage (b) Coverage reward

Fig. 3: Coverage performance under time-varying density fields.



(a) Connectivity percentage (b) Connectivity reward

Fig. 4: Connectivity performance under decentralized coverage.

Beyond spatial efficiency, RS network connectivity is critical for decentralized coordination. As shown in Fig. (4) (a), The connectivity percentage of our method remains above 52%, versus 32.5% for GNN and 27.3% for Lloyd. More importantly, the curve under our method decays smoothly, whereas GNN and Lloyd show sharper degradation. The connectivity reward curves in Fig. (4) (b) highlight this stability advantage: our method converges to -10.6, while the other methods exhibit larger oscillations and worse penalties. The reduced oscillatory behavior indicates improved structural robustness and mitigates the classical trade-off between coverage dispersion and communication cohesion.

We further investigate its transferability across varying communication radii and scalability to different RSs, with environmental setup and density field parameters unchanged.

Experiment 2 (Communication capability generalization): We examine the sensitivity of the learned policy to different communication radii $R_c = \{10, 20, 30, 40, 50, 60\}$. The results are shown in Fig. 5 (a), where the color map indicates the reward advantage of our method and GNN over Lloyd. As the radius decreases, the coverage task becomes coordination-critical due to limited perception. Under small-radius settings, the performance gap between our method and GNN becomes more pronounced, indicating that attention-based aggregation is beneficial when robots rely on inter-robot information exchange. By dynamically assigning importance weights to neighboring robots, our strategy prioritizes informative interactions in regions with high density gradients, accelerating convergence and improving coverage performance. In contrast, the uniform aggregation mechanism of GNN limits its ability to adapt to spatial heterogeneity. As the radius increases, improved perception benefits all methods and reduces reliance on communication-driven coordination. Consequently, the relative advantage of graph-based learning diminishes. Nevertheless, our strategy achieves superior reward performance across all radii, demonstrating robustness to perception-scale variations.

	R_c=10	R_c=20	R_c=30	R_c=40	R_c=50	R_c=60
GNN	31.05	28.37	19.97	12.28	8.33	1.41
GAT	52.24	47.68	26.59	19.46	12.66	2.25

60 30 0

(a) Generalization across communication capability

	I = 5	I = 10	I = 15	I = 20	I = 25	I = 30
GNN	26.72	43.56	25.11	13.62	4.53	1.01
GAT	33.28	52.39	35.44	18.57	6.36	1.95

60 30 0

(b) Scalability across swarm sizes

Fig. 5: Generalization performance of the proposed strategy.

Experiment 3 (RS scale generalization): We evaluate scalability by varying the RS size $I = \{5, 10, 15, 20, 25, 30\}$, as illustrated in Fig. 5 (b). With few robots, spatial sparsity makes information propagation critical for load-balanced coverage. In this regime, both our method and GNN outperform Lloyd, with our method achieving the largest reward improvement due to its adaptive attention mechanism. As the RS size increases, marginal coverage gains saturate because of spatial redundancy, and the performance gap between learning-based methods and Lloyd's algorithm narrows. Nevertheless, our strategy consistently outperforms both GNN and Lloyd across all RS sizes. Importantly, performance does not degrade with graph size, suggesting that the attention mechanism mitigates the over-smoothing effect common in deep graph networks and preserves discriminative inter-robot information across RS scales.

Additionally, the algebraic connectivity of the RS network remains positive under the proposed strategy in all experiments, indicating stable topology despite varying communication capabilities and RS scales. This observation implies that the learned residual policy encodes connectivity-aware motion behaviors, enabling a balanced trade-off between spatial dispersion for coverage maximization and network cohesion for coordination. Therefore, the proposed strategy exhibits robustness to perception changes and scalable performance with increasing RS scales, without retraining or structural modification.

B. Robot Deployment

To bridge the gap between numerical simulations and real-world implementations, we conducted physical experiments to evaluate the practical feasibility and robustness of the proposed framework. The experimental platform consists of three main components: a Nokov motion capture system for high-precision localization, a team of Limo robots equipped with mecanum wheels for omnidirectional mobility, and a centralized ground station for monitoring and data logging. Real-time position and orientation data captured by the Nokov system are transmitted to each robot, which executes the proposed strategy. All inter-robot communication is implemented in a distributed manner, consistent with the assumptions adopted in simulation.

We deployed 6 robots to perform coverage tasks over a $5\ \text{m} \times 4\ \text{m}$ workspace. No additional fine-tuning or retraining was performed prior to deployment, enabling a direct evaluation of sim-to-real transferability. Each robot carries a GAT-based actor–critic learning module that processes local state features and neighbor information to compute coverage commands in real time. The partitions in Fig. 6 evolve smoothly and remain well-shaped, indicating that the CVT structure is preserved while the learned residual component refines local adjustments. This confirms that the proposed strategy successfully integrates geometric interpretability with data-driven adaptability.

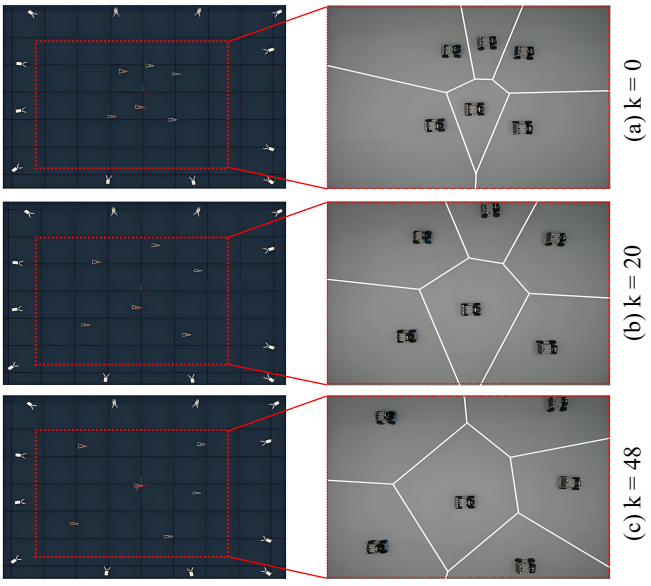
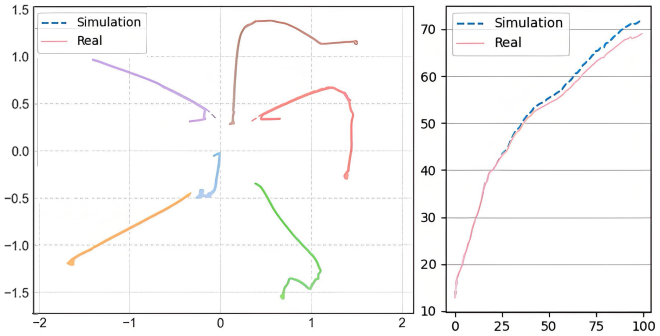


Fig. 6: Voronoi tessellations evolution during robots deployment.

The robots disperse as shown in Fig. 7 (a) and the RS network connectivity is preserved, which forms a spatial configuration consistent with load-aware coverage observed in simulation. No communication disconnections occurred during the experiment, confirming that the learned residual policy implicitly encodes connectivity-preserving motion patterns under real-world actuation and sensing uncertainties. The coverage percentage over time is plotted in Fig. 7 (b). The experimental curve matches the simulation results, with a final coverage deviation of less than 5%. The discrepancy arises from actuation delays, wheel slippage, and communication latency. Nevertheless, convergence and steady-state performance remain consistent, demonstrating robust coverage and effective sim-to-real transfer.



(a) Sim-to-real trajectory comparison (b) Coverage percentage

Fig. 7: Comparison between simulation and robot deployments.

V. Conclusion

This paper presents a hybrid model-based and learning-enhanced framework for load-balanced distributed coverage in RSs. By embedding a GAT within a CVT-based control architecture, the proposed approach preserves the geometric interpretability of classical coverage control while enabling decentralized coordination through learned graph representations. The attention mechanism allows robots to selectively emphasize informative neighbors, improving coverage efficiency while maintaining communication connectivity. Reinforcement learning is employed to optimize a residual policy that complements the nominal CVT controller, enabling adaptive responses to dynamic target distributions and environmental variations. Simulations demonstrate that our approach achieves consistent improvements in coverage performance, load balance, and robustness under communication constraints compared with baseline methods. Hardware experiments with multiple mobile robots further validate the practical feasibility of the proposed framework and confirm its capability to reproduce the desired coverage behaviors in real-world environments. Future work will establish theoretical guarantees under time-varying communication graphs and extend the framework to large-scale heterogeneous RSs.

Acknowledgments

We gratefully acknowledge Beijing NOKOV Science & Technology Co., Ltd. for providing the motion capture system to facilitate the development of our experiment platform.

References

[1] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[2] M. Santos, Y. Diaz-Mercado, and M. Egerstedt, "Coverage control for multirobot teams with heterogeneous sensing capabilities," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 919–925, 2018.

[3] C. Zhai, P. Fan, and H. T. Zhang, "Sectorial coverage control with load balancing in non-convex hollow environments," *Automatica*, vol. 157, p. 111246, 2023.

[4] L. Zhang, Z. Zhang, R. Siegwart, and J. J. Chung, "Distributed PDOP coverage control: Providing large-scale positioning service using a multi-robot system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2217–2224, 2021.

[5] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked robots," in *IEEE International Conference on Robotics and Automation*, pp. 4982–4989, 2010.

[6] A. Pietrabissa, F. Liberati, and G. Oddi, "A distributed algorithm for ad-hoc network partitioning based on voronoi tessellation," *Ad Hoc Networks*, vol. 46, pp. 37–47, 2016.

[7] Y. Liu, W. Wang, B. Lévy, F. Sun, D.-M. Yan, L. Lu, and C. Yang, "On centroidal voronoi tessellation—energy smoothness and fast computation," *ACM Trans. on Graphics*, vol. 28, no. 4, pp. 1–17, 2009.

[8] Y. Gao, H. Gao, Y. Ji, J. Zhou, and Y. Shi, "A dual calibration framework for exploring environments using heterogeneous robot swarms," in *51st Annual Conference of the IEEE Industrial Electronics Society*, pp. 1–7, 2025.

[9] Y. Gao, H. Gao, Z. Wang, Y. Shi, and Y. Ji, "Event-triggered control for autonomous detection and treatment of membrane lesions using microrobot swarms," in *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1252–1257, 2025.

[10] Y. Gao, Z. Zhou, H. Gao, S. Zhang, and Y. Ji, "Hybrid control for robot swarms to detect critical nodes in heterogeneous sensor networks," in *IEEE 21st International Conference on Automation Science and Engineering*, pp. 3462–3467, 2025.

[11] N. Gao, L. Liang, D. Cai, X. Li, and S. Jin, "Coverage control for uav swarm communication networks: A distributed learning approach," *IEEE Int. of Things Journal*, vol. 9, no. 20, pp. 19854–19867, 2022.

[12] Q. Wang, W. Li, and A. Mohajer, "Load-aware continuous-time optimization for multi-agent systems: Toward dynamic resource allocation and real-time adaptability," *Comput. Netw.*, vol. 250, p. 110526, 2024.

[13] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14413–14423, 2020.

[14] S. Wu, Z. Pu, T. Qiu, J. Yi, and T. Zhang, "Deep-reinforcement-learning-based multitarget coverage with connectivity guaranteed," *IEEE Trans. Industrail Informat.*, vol. 19, no. 1, pp. 121–132, 2022.

[15] N. Dhanaraj, J. H. Kang, A. Mukherjee, H. Nemlekar, S. Nikolaidis, and S. K. Gupta, "Multi-robot task allocation under uncertainty via hindsight optimization," in *IEEE International Conference on Robotics and Automation*, pp. 16574–16580, 2024.

[16] 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 IEEE International Conference on Robotics and Automation*, pp. 8787–8793, 2022.

[17] S. Munikoti, D. Agarwal, L. Das, M. Halappanavar, and B. Natarajan, "Challenges and opportunities in deep reinforcement learning with graph neural networks: A comprehensive review of algorithms and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 15051–15071, pp. 121–132, 2023.

[18] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 11785–11792, 2020.

[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations*, 2018.

[20] 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 Info. Proces. Syst.*, vol. 30, 2017.

[21] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," in *International Conference on Robotics and Automation*, pp. 6023–6029, 2019.