A Multi-Modal DRL Framework for Fair and Efficient Multi-UAV Communication Coverage

Cheng Cui*, Xiaoyu Wang*†, Ning Chen*
*School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China
Emails: ccui01@stu.suda.edu.cn, {xywang21, ningc}@suda.edu.cn

2025 IEEE 31th International Conference on Parallel and Distributed Systems (ICPADS) | 979-8-3315-4901-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICPADS67057.2025.11323046

***Abstract*—Unmanned Aerial Vehicles (UAVs) are critical for wireless communication in dynamic, infrastructure-less environments. However, optimizing trajectories to maximize coverage while ensuring fairness and minimizing energy consumption remains challenging. Relying on a single modality is a key limitation, as graph-only methods miss crucial spatial context while image-only methods ignore the underlying network structure. We propose Multi-Modal Dynamic Edge-featured Graph Attention Network (MM-DEGAT), a novel deep reinforcement learning framework that integrates image-based spatial distributions and graph-structured topological relations. Our approach includes: (1) a dynamic edge-featured mechanism for modeling relationships within the network, (2) bi-directional cross-attention for deep image-graph integration, and (3) adaptive gated fusion for dynamic multi-modal weighting. Experiments demonstrate improved communication efficiency and fairness, validating MM-DEGAT's effectiveness for UAV coverage applications.**

***Index Terms*—Unmanned Aerial Vehicle, Multi-Modal DRL, Communication Coverage, Graph Attention Network, Fairness**

I. Introduction

The deployment of Unmanned Aerial Vehicles (UAVs) as mobile base stations has emerged as a flexible and cost-effective solution for extending wireless coverage, particularly in emergency response, remote areas, and capacity enhancement for urban hotspots [1]. Unlike static infrastructure, UAVs can dynamically adjust their positions to adapt to shifting user demands and complex environmental conditions.

However, UAVs present significant trajectory optimization challenges due to their flexibility, extensively researched for communication applications. Early works employed traditional optimization techniques and heuristic approaches. Recent Deep Reinforcement Learning (DRL) methods can manage high-dimensional dynamics for adaptive trajectory control [2]. Nevertheless, existing approaches fundamentally suffer from incomplete environmental perception. Traditional and heuristic methods rely on simplified abstract models that fail to capture real-time interplay among user distributions, obstacles, and channel dynamics. While DRL methods better handle dynamic environments, they primarily use single-modal information. This perceptual limitation becomes particularly acute when addressing the core multi-objective problem: the cooperative control of a UAV fleet to provide both efficient and fair wireless coverage under stringent operational constraints. The objectives are inherently conflicting: maximizing aggregate data throughput might incentivize serving only easily reachable user clusters, thereby sacrificing service fairness for users at the periphery. Conversely, enforcing strict fairness could diminish overall system efficiency. This trade-off is further compounded by the highly dynamic environment, which includes not only shifting user states and time-varying channel conditions but also the UAVs' own finite energy reserves. Formally, this sequential decision-making task can be modeled as a large-scale Markov Decision Process (MDP) with an immense state-action space, rendering it intractable for classical optimization techniques.

To overcome the complementary weaknesses of single-modal approaches, we argue that a holistic solution can fuse information from different modalities. Graph-only methods, while capturing network topology, are blind to physical geometry; a boundary wall, for instance, is treated as an abstract disconnection, leading to suboptimal policies. Conversely, vision-based methods excel at spatial awareness but fail to explicitly model crucial topological information like link quality or interference. This paper addresses this fundamental gap by developing a multi-modal image-graph fusion DRL framework.

The primary contributions of this work are as follows:

- We propose a novel DRL framework that, for the first time, systematically fuses visual features and communication topological structures for the UAV coverage problem, enabling comprehensive environmental perception.
- We develop the Dynamic Edge-featured Graph Attention Network (DEGAT) as the backbone for graph representation learning in our MM-DEGAT framework.
- We design a novel bidirectional cross-attention mechanism with an adaptive gate to facilitate deep information coupling between modalities and dynamically balance their influence.
- We implement a scalable and efficient distributed training architecture based on PPO, making the training of our complex, high-dimensional model tractable.
- Through comprehensive experiments, we demonstrate that our proposed method outperforms representative baselines in communication efficiency and fairness.

II. Related Work

A. DRL-based UAV Communication Control

A significant body of research leverages DRL to manage key objectives such as trajectory planning, resource allocation, and

†Xiaoyu Wang is the corresponding author.

energy efficiency. For instance, value-based methods like Deep Q-Networks (DQN) have been adapted for energy-efficient data collection [3]. Concurrently, policy-gradient algorithms, particularly Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG), are widely employed for continuous control tasks like trajectory optimization and computational offloading, valued for their stability and sample efficiency [4], [5]. These frameworks can effectively handle the joint state of multiple UAVs within a centralized control paradigm. Multi-Agent DRL (MADRL) has also gained traction, often utilizing the Centralized Training with Decentralized Execution (CTDE) principle [6]. It explicitly models the interactions between individual UAV agents [7]. Algorithms such as Multi-Agent PPO (MAPPO) are commonly used to maximize collective objectives like data collection rates [8].

While DRL and MADRL frameworks are powerful, their performance depends on state representation quality. Many existing methods use low-dimensional states that inadequately capture spatial-topological nuances of deployment areas.

B. Multi-Modal Perception

To overcome the perceptual limitations of vectorized states, researchers have begun to explore richer sensory modalities.

On one hand, graph neural networks (GNNs) have been introduced to explicitly model the complex topological relationships in wireless networks [9]. In multi-UAV systems, GNNs can be used to optimize resource allocation [10]. However, while GNNs excel at processing relational data [11], they are inherently blind to the physical environment, lacking awareness of geometric obstacles or spatial user distributions not encoded in the graph. On the other hand, vision-based methods employ environmental images for UAV control. Convolutional neural networks (CNNs) enable UAVs to learn navigation and obstacle avoidance directly from vision inputs [12]. While excelling in spatial awareness, this approach fails to explicitly encode abstract network optimization information such as signal-to-noise ratio or interference levels.

Given the complementary weaknesses of single modalities, with graph methods lacking spatial perception and vision methods lacking topological precision, their fusion presents a promising direction. However, effectively combining environmental images with communication topology graphs for UAV coverage optimization remains rarely explored. Our work aims to fill this critical gap.

III. System Model and Problem Formulation

In this section, we establish the model for a multi-UAV wireless communication system, defining the environment, communication links, energy dynamics, and formally stating the optimization problem.

A. Scenario Description

As shown in Fig. 1, we consider a square geographical area of size $L \times L$ square meters, populated by three types of entities. The set of N static ground users is denoted by $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$, where the j-th user is located at a position $\mathbf{p}_j = (x_j, y_j, 0)$ and has an accumulated throughput of $\Theta_j(t)$ at time slot t. The set of K UAVs, acting as aerial base stations, is denoted by $\mathcal{A} = \{a_1, a_2, \ldots, a_K\}$. The state of the k-th UAV is characterized by its 3D position $\mathbf{q}_k(t) = (x_k(t), y_k(t), h_k)$ at a fixed altitude h_k, and its remaining battery energy $E_k(t) \in [0, E_{\max}]$. All UAVs are equipped with N_s orthogonal subchannels for communication. Finally, the set of M charging stations is denoted by $\mathcal{S} = \{s_1, s_2, \ldots, s_M\}$. The c-th station is located at a fixed location $\mathbf{l}_c = (x_c, y_c, 0)$ and is characterized by its constant transmit power P_c and antenna gain G_c. The system's operation is discretized into a sequence of time slots, $t \in \{1, 2, \ldots, T\}$. Each time slot, with a total duration of Δt, is composed of a movement phase, Δt_m, followed by a service phase (for communication or charging), Δt_s.

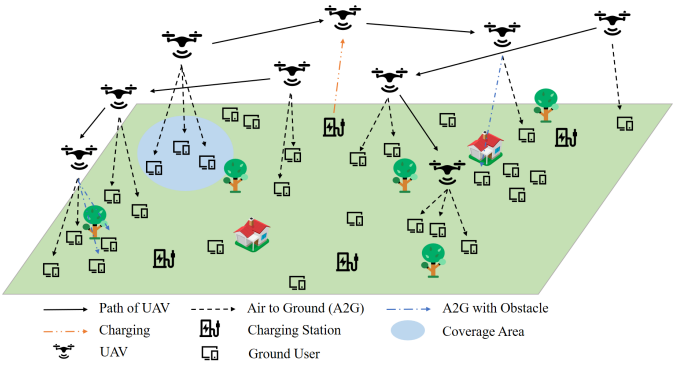

Fig. 1. An illustration of the multi-UAV communication coverage scenario, where UAVs serve ground users and recharge at designated stations.

B. Communication Model

We model the air-to-ground (A2G) communication link from the UAVs to the ground users.

1) Channel Model: The A2G channel is characterized by both line-of-sight (LoS) and non-line-of-sight (NLoS) propagation. The probability of a LoS link between the k-th UAV and the j-th user depends on the elevation angle $\theta_{k,j} = \arctan(h_k/d_{k,j})$, where $d_{k,j}$ is the horizontal distance between them. The LoS probability is given by the widely-adopted sigmoidal function [13]:

$$P_{\text{LoS}}(\theta_{k,j}) = \frac{1}{1 + a \exp\left(-b\left(\frac{180}{\pi}\theta_{k,j} - a\right)\right)}, \tag{1}$$

where a and b are environment-specific constants. The path loss exponent $\alpha_{k,j}$ is then a weighted average of the LoS and NLoS exponents:

$$\alpha_{k,j} = P_{\text{LoS}}(\theta_{k,j}) \cdot \alpha_{\text{LoS}} + (1 - P_{\text{LoS}}(\theta_{k,j})) \cdot \alpha_{\text{NLoS}}. \tag{2}$$

2) Signal-to-Noise Ratio (SNR): Assuming the total UAV transmission power P_{tx} is uniformly distributed across its N_s subchannels, the power on the m-th subchannel is $P_{tx,m} = P_{tx}/N_s$. The SNR of the link from the k-th UAV to the j-th user on the m-th subchannel is:

$$\omega_{k,j,m} = \frac{P_{tx,m} G_t G_r G_{k,j}}{N_0 B_m}, \tag{3}$$

where $G_{k,j}$ is the channel gain, which is dependent on the path loss $\alpha_{k,j}$ and distance. G_t and G_r are the transmit and receive antenna gains. N_0 is the noise power spectral density, and B_m is the bandwidth of a subchannel.

3) Achievable Throughput: The achievable data rate for the link is given by the Shannon capacity formula:

$$R_{k,j,m}(t) = B_m \log_2(1 + \gamma \cdot \omega_{k,j,m}(t)), \tag{4}$$

where γ is a constant SNR gap.

C. User Association

Let $x_{k,j,m}(t) \in \{0, 1\}$ be the binary association variable, where $x_{k,j,m}(t) = 1$ if the j-th user is served by the k-th UAV on its m-th subchannel at time t, and 0 otherwise. Furthermore, an association is only possible if the user is within the UAV's communication coverage range. The association must satisfy the following constraints:

$$\sum_{k=1}^{K} \sum_{m=1}^{N_s} x_{k,j,m}(t) \leq 1, \qquad \forall j, t, \tag{5}$$

$$\sum_{j=1}^{N} x_{k,j,m}(t) \leq 1, \qquad \forall k, m, t, \tag{6}$$

$$x_{k,j,m}(t) = 1 \Rightarrow \|\mathbf{q}_k(t) - \mathbf{p}_j\| \leq R_{cov}, \quad \forall k, j, m, t. \tag{7}$$

Constraint (5) ensures each user is served by at most one UAV subchannel. Constraint (6) ensures each UAV subchannel serves at most one user. Constraint (7) ensuring that an associated user is within the UAV's coverage radius to meet the QoS requirements of ground users. In this work, we adopt a greedy association strategy based on the highest SNR for users within range, followed by a backfilling mechanism.

D. Energy Consumption and Charging Model

We model the energy dynamics of each UAV, considering both the energy consumed for propulsion and the energy replenished during charging.

1) Propulsion Energy Consumption: The propulsion power of the k-th UAV, $P(v_k)$, is a non-linear function of its flight speed v_k. We adopt the detailed aerodynamic model for rotary-wing UAVs presented in [14]:

$$P(v_k) = P_0 \left(1 + \frac{3v_k^2}{U_{tip}^2}\right) + P_I \sqrt{\sqrt{1 + \frac{v_k^4}{4v_0^4}} - \frac{v_k^2}{2v_0^2}} + \frac{1}{2} d_0 \rho s A v_k^3, \tag{8}$$

where U_{tip}, v_0, d_0, ρ, s, and A respectively represent the tip speed of the rotor blade, the average rotor induced velocity during hovering, the fuselage drag ratio, the air density, the rotor solidity, and the rotor disc area. P_0 and P_I represent the blade profile and induced power, which are the main components of hovering power.

The total energy consumed by the k-th UAV in a time slot t, denoted as $E_{cs,k}(t)$, consists of two parts: the energy for flight during the movement phase (Δt_m) and the energy for hovering during the service phase (Δt_s).

During the movement phase, the UAV travels at speed $v_k(t)$, and the energy consumed is:

$$E_{move,k}(t) = P(v_k(t)) \cdot \Delta t_m. \tag{9}$$

During the service phase, the UAV hovers at its position, so its speed is zero ($v_k = 0$). The hovering power is thus $P(0) = P_0 + P_I$. The energy consumed for hovering is:

$$E_{hover,k}(t) = P(0) \cdot \Delta t_s. \tag{10}$$

Therefore, the total propulsion energy consumption in time slot t is the sum of these two components:

$$E_{cs,k}(t) = E_{move,k}(t) + E_{hover,k}(t). \tag{11}$$

2) Charging Model: To accurately model the energy dynamics, we define the power received by a UAV as a function of its distance from the charging station and several key physical parameters [15]–[22].

The power received by the k-th UAV from the c-th charging station at time t is:

$$P_{cg,k,c}(t) = P_c G_c G_k^{\text{RX}} \eta \left(\frac{\lambda_c}{4\pi \|\mathbf{q}_k(t) - \mathbf{l}_c\|}\right)^2, \tag{12}$$

where G_k^{RX} is the UAV's receive antenna gain; η is the rectifier efficiency; and λ_c is the signal wavelength. This model assumes line-of-sight (LoS) conditions.

A UAV can replenish its battery if it is within the charging range R_{cg} of a station. Let $C_k(t) \in \{0, 1\}$ be a binary indicator, where $C_k(t) = 1$ if the k-th UAV is charging at time t. The energy replenished during the service phase is $E_{cg,k}(t) = C_k(t) \cdot P_{cg,k,c}(t) \cdot \Delta t_s$. The battery level of the k-th UAV evolves according to:

$$E_k(t+1) = \min\left(E_{\max}, E_k(t) - E_{cs,k}(t) + E_{cg,k}(t)\right), \tag{13}$$

where $E_{\max}$ is the maximum battery capacity.

E. Problem Formulation

The goal is to jointly maximize system-wide communication efficiency and fairness by planning the trajectory of the UAVs, represented by $\{\mathbf{q}_k(t)\}_{k=1,t=1}^{K,T}$. The formalized problem is defined as follows:

$$\max_{\{\mathbf{q}_k(t)\}} \quad w_{\mathcal{F}} \mathcal{F}_{\text{norm}}(T) + w_{\mathcal{E}} \mathcal{E}_{\text{norm}}(T) \tag{14}$$

$$\text{s.t.} \quad E_k(t) \geq E_{\min}, \qquad \forall k, t, \tag{15}$$

$$\frac{\|\mathbf{q}_k(t') - \mathbf{q}_k(t)\|}{\Delta t_m} \leq V_{\max}, \qquad \forall k, t, \tag{16}$$

$$0 \leq x_k(t), y_k(t) \leq L, \qquad \forall k, t, \tag{17}$$

$$\text{Constraints (5), (6), and (7),}$$

where t' denotes the end of the movement phase, and $w_{\mathcal{F}}$ and $w_{\mathcal{E}}$ are non-negative weights that balance the two objectives. In this work, we set $w_{\mathcal{F}} = w_{\mathcal{E}} = 0.5$ to assign equal importance to both KPIs. The agent is guided by an instantaneous reward function $r(t) = w_{\mathcal{E}} r_{\mathcal{E}_{\text{norm}}}(t) + w_{\mathcal{F}} r_{\mathcal{F}_{\text{norm}}}(t) + w_p r_p(t)$. The components $r_{\mathcal{E}_{\text{norm}}}(t)$ and $r_{\mathcal{F}_{\text{norm}}}(t)$ are the instantaneous (per-timeslot) versions of the normalized KPIs defined below.

The term $r_p(t)$ is a negative penalty applied when violating constraints, such as the minimum battery (Eq. 15) or flying outside the boundary (Eq. 17). The key performance indicators (KPIs) and their normalized versions are defined as:

1) **System Communication Efficiency ($\mathcal{E}(T)$):**

$$\mathcal{E}(T) = \frac{\sum_{t=1}^{T}\sum_{j=1}^{N} R_j(t)}{\sum_{t=1}^{T}\sum_{k=1}^{K} E_{cs,k}(t)}, \tag{18}$$

where $R_j(t) = \sum_k \sum_m x_{k,j,m}(t) R_{k,j,m}(t)$. The efficiency is normalized by a predetermined fixed value $\mathcal{E}_{\max}$, which represents a target or a practical upper bound for the system's efficiency, to scale the value to a comparable range:

$$\mathcal{E}_{\text{norm}}(T) = \frac{\mathcal{E}(T)}{\mathcal{E}_{\max}}. \tag{19}$$

2) **Jain's Fairness Index ($\mathcal{F}(T)$):**

$$\mathcal{F}(T) = \frac{\left(\sum_{j=1}^{N} \Theta_j(T)\right)^2}{N\sum_{j=1}^{N} \Theta_j^2(T)}, \tag{20}$$

where $\Theta_j(T) = \sum_{t=1}^{T} R_j(t)$. As Jain's Index is inherently bounded between [0, 1], it can be used directly in its normalized form:

$$\mathcal{F}_{\text{norm}}(T) = \mathcal{F}(T). \tag{21}$$

IV. Proposed Solution: MM-DEGAT

To solve the complex optimization problem formulated in Section III-E, we model the sequential control task as a Markov Decision Process (MDP) and propose a novel Deep Reinforcement Learning (DRL) solution. The goal is to learn a control policy π that maps system states to actions, maximizing a cumulative reward aligned with the objectives in Eq. (14). It is specifically designed to address the challenges of fusing multimodal information for optimizing UAV communication coverage. This section details the three core components of our solution: the multi-modal observation space, the MM-DEGAT network architecture, and the policy learning method including the reward function design.

A. Multi-Modal Observation Space Design

A comprehensive understanding of the environment is critical for effective decision-making. We design a multi-modal observation space that captures both the continuous, spatial layout of the environment and the discrete, topological relationships within the communication network.

1) Image Observation: To capture the global spatial context, we use a two-channel image $\mathbf{I}(t) \in \mathbb{R}^{W\times H\times 2}$, where W and H are the width and height of the grid-based representation of the operational area. The two channels separate dynamic user state from the more static physical layout:

- **Channel 0 (User Map):** This channel represents the spatial distribution and service priority of users. We use Gaussian kernels centered at each user's location, with the intensity weighted by their accumulated throughput $\Theta_j(t)$. This allows the agent to visually identify underserved areas.
- **Channel 1 (Entity Location Map):** Encodes positions of UAVs, charging stations, and any no-fly zones.

2) Graph Structure Observation: While images excel at spatial context, a graph structure, $\mathcal{G}(t) = (\mathcal{X}, \mathcal{L})$, is better suited for modeling the explicit, dynamic relationships within the network. The graph is constructed at each time step t as follows:

1) **Node Set ($\mathcal{X}$):** The node set $\mathcal{X}$ is the union of all entities defined in the scenario description: $\mathcal{X} = \mathcal{A} \cup \mathcal{U} \cup \mathcal{S}$. Each node is endowed with a feature vector containing its crucial attributes:
   - *UAV Node ($a_k \in \mathcal{A}$):* Its feature vector includes 2D position, altitude, remaining energy ratio, and a binary charging indicator. $\mathbf{h}_{a_k} = [x_k, y_k, h_k, E_k/E_{\max}, C_k(t)]$.
   - *User Node ($u_j \in \mathcal{U}$):* Its feature vector includes its 2D position and accumulated throughput. $\mathbf{h}_{u_j} = [x_j, y_j, \Theta_j(t)]$.
   - *Station Node ($s_c \in \mathcal{S}$):* Its feature vector simply contains its 2D position. $\mathbf{h}_{s_c} = [x_c, y_c]$.
2) **Edge Set ($\mathcal{L}$):** Edges represent potential connections between entities and are established based on distance thresholds, defining the graph's topology:
   - An edge exists between a UAV node a_k and a user node u_j if the user is within the UAV's communication range ($d_{kj} \le R_{cov}$).
   - An edge exists between a UAV node a_k and a station node s_c if the UAV is within the station's charging range ($d_{kc} \le R_{cg}$).
3) **Edge Features ($\mathbf{e}_{\zeta\xi}$):** For each edge $(\zeta, \xi) \in \mathcal{L}$ connecting two nodes, we define a feature vector to capture their spatial relationship. A simple yet effective feature vector is the relative distance vector:

$$\mathbf{e}_{\zeta\xi} = [x_\xi - x_\zeta, y_\xi - y_\zeta, d_{\zeta\xi}], \tag{22}$$

where $d_{\zeta\xi}$ is the Euclidean distance between nodes ζ and ξ.

B. MM-DEGAT Network Architecture

The core of our agent is the MM-DEGAT network, which is designed to effectively encode and fuse the multi-modal observations, as illustrated in Fig. 2. It consists of three main components: two parallel feature encoders, a fusion module, and the policy/value heads.

1) Image Encoder: A standard convolutional neural network (CNN) [23] is used to process the image observation $\mathbf{I}(t)$. It consists of several convolutional layers followed by fully connected layers, which extract hierarchical spatial features and produce a compact image embedding vector $\mathbf{f}_{\text{img}} \in \mathbb{R}^{d_{img}}$. In our implementation, this embedding dimension d_{img} is set to 256.

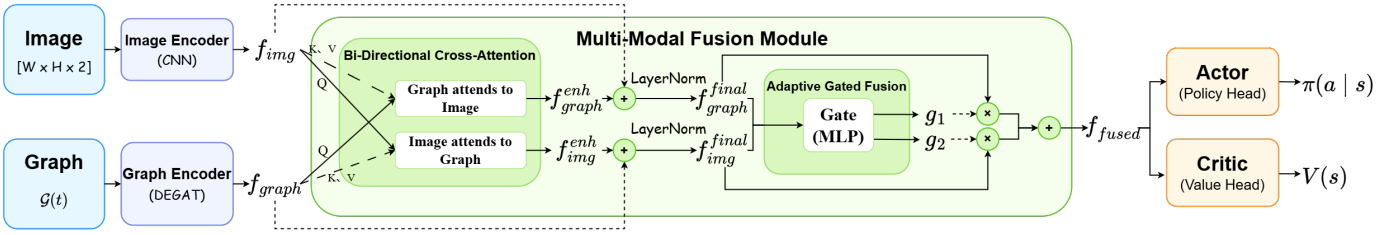

Fig. 2. The overall network architecture of the proposed MM-DEGAT model.

2) Graph Structure Encoder: To process the graph observation $\mathcal{G}(t)$, we propose the Dynamic Edge-featured Graph Attention Network (DEGAT), which integrates two key mechanisms:

1) **Dynamic Attention [24]:** Computes dynamic attention weights that depend on both the source and target nodes, making the model more expressive.
2) **Edge-featured Mechanism [25]:** Explicitly incorporates edge features $\mathbf{e}_{\zeta\xi}$ into the attention computation, which is crucial as features like distance directly determine link quality.

The update rule for a node ζ in the l-th DEGAT layer, aggregating information from its neighbors $\xi \in \mathcal{N}(\zeta)$, is:

$$\mathbf{h}_{\zeta}^{(l+1)} = \sigma\left(\sum_{\xi\in\mathcal{N}(\zeta)} \alpha_{\zeta\xi}^{(l)} \mathbf{W}^{(l)}[\mathbf{h}_{\zeta}^{(l)} \| \mathbf{h}_{\xi}^{(l)} \| \mathbf{e}_{\zeta\xi}]\right), \tag{23}$$

where $\alpha_{\zeta\xi}^{(l)}$ is the dynamic edge-featured attention weight.

After several layers of message passing, we use an attention-based pooling layer to obtain a global graph embedding $\mathbf{f}_{\text{graph}} \in \mathbb{R}^{d_{graph}}$.

3) Multimodal Fusion Module: This module is designed to deeply integrate the information from the two modalities rather than simply concatenating them.

- **Bi-Directional Cross-Attention:** We employ a cross-attention mechanism where the image and graph embeddings mutually inform one another. Symmetrically, the graph embedding $\mathbf{f}_{\text{graph}}$ queries the image embedding $\mathbf{f}_{\text{img}}$ (as key/value) to produce an image-enhanced graph feature $\mathbf{f}^{\text{enh}}_{\text{graph}}$, and vice-versa to produce a graph-enhanced image feature $\mathbf{f}^{\text{enh}}_{\text{img}}$.
- **Adaptive Gated Fusion:** After enhancement, the features are fused using a dynamic gate. First, we apply residual connections to incorporate the original information and then use Layer Normalization (LN) to stabilize the activations:

$$\mathbf{f}^{\text{final}}_{\text{graph}} = \text{LN}(\mathbf{f}_{\text{graph}} + \mathbf{f}^{\text{enh}}_{\text{graph}}), \tag{24}$$
$$\mathbf{f}^{\text{final}}_{\text{img}} = \text{LN}(\mathbf{f}_{\text{img}} + \mathbf{f}^{\text{enh}}_{\text{img}}). \tag{25}$$

Next, to dynamically balance their influence, these final features are concatenated and passed through a small gating network to generate a gating vector $\mathbf{g}$:

$$\mathbf{g} = \text{softmax}(\mathbf{W}_g[\mathbf{f}^{\text{final}}_{\text{graph}} \| \mathbf{f}^{\text{final}}_{\text{img}}] + \mathbf{b}_g). \tag{26}$$

where $\mathbf{W}_g$ and $\mathbf{b}_g$ are the trainable weight and bias of the gate. The output $\mathbf{g} = [g_1, g_2]$ provides dynamic weights for the graph and image modalities. Finally, the fused representation $\mathbf{f}_{\text{fused}}$ is computed as their weighted sum:

$$\mathbf{f}_{\text{fused}} = g_1 \mathbf{f}^{\text{final}}_{\text{graph}} + g_2 \mathbf{f}^{\text{final}}_{\text{img}}. \tag{27}$$

since the gated fusion produces a weighted sum of the two final features, the fused representation $\mathbf{f}_{\text{fused}} \in \mathbb{R}^{d_{\text{fused}}}$, where $d_{\text{fused}} = 256$ is chosen to match the image embedding dimension.

C. Actor-Critic Network

The fused feature vector $\mathbf{f}_{\text{fused}}$ is fed into an Actor-Critic network trained with PPO.

- **Actor Network:** Generates a continuous action (e.g., movement vector) by modeling the policy as a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$.
- **Critic Network:** Estimates the state-value function $V(\mathbf{s})$ to help stabilize training by reducing gradient variance.

D. Policy Learning with Proximal Policy Optimization (PPO)

To train the actor-critic network, we employ Proximal Policy Optimization (PPO) [26]. PPO is an on-policy algorithm that learns by collecting a batch of experience using the current policy and then performing several epochs of updates on that data. The final PPO objective function is:

$$L(\theta) = \mathbb{E}_t\left[L^{\text{CLIP}}(\theta) - c_1 L^{VF}(\theta) + c_2 S[\pi_\theta](s_t)\right], \tag{28}$$

where $L^{\text{CLIP}}(\theta)$ is the clipped surrogate objective, $L^{VF}(\theta)$ is the value function loss, and $S[\pi_\theta]$ is the policy entropy. The advantage function required for the policy loss is estimated using Generalized Advantage Estimation (GAE) [27]. The PPO training loop is outlined in Alg. 1.

E. Scalable Distributed Training with PPO

While the standard PPO algorithm (Alg. 1) provides a robust learning framework, its single-threaded nature of data collection creates a bottleneck for complex problems like ours. To accelerate training and improve scalability, we implement a distributed Actor-Learner architecture based on PPO. This framework decouples experience collection from policy learning, allowing us to generate parallel data.

The framework consists of a single centralized Learner process and multiple, parallel Actor processes.

Algorithm 1: Proximal Policy Optimization (PPO)

Input : Hyperparameters: learning rate α, trajectory length T_{max}, epochs E, max episodes N_{epi}
Output: Trained policy parameters θ

Initialize policy parameters θ using Orthogonal Initialization for weights and zero for biases;
for *episode* $\leftarrow$ *1 to* N_{epi} **do**
  Initialize trajectory buffer $\mathcal{D}$;
  Receive initial state s_0;
  for $t \leftarrow 0$ *to* $T_{max} - 1$ **do**
    Sample action $\mathbf{a}_t \sim \pi_\theta(\cdot|s_t)$;
    Execute $\mathbf{a}_t$, get reward r_t and next state s_{t+1};
    Store transition $(s_t, \mathbf{a}_t, r_t, s_{t+1})$ in $\mathcal{D}$;
    $s_t \leftarrow s_{t+1}$;
  Compute advantages $\hat{A}_t$ and value targets V_t^{target} for all transitions in $\mathcal{D}$ using GAE;
  for *epoch* $\leftarrow$ *1 to E* **do**
    for *each minibatch in* $\mathcal{D}$ **do**
      Compute PPO loss $L(\theta)$;
      Update policy parameters θ via gradient descent;

Algorithm 2: Distributed PPO: Actor Process

Input : Reference to global network π_θ; Hyperparameters: trajectory length T_{max}, max episodes N_{epi}
Output: Gradients, sent to the Learner

Initialize local network $\pi_{\theta_{\text{local}}}$ and environment;
for *episode* $\leftarrow$ *1 to* N_{epi} **do**
  $\theta_{\text{local}} \leftarrow$ SyncWeights(π_θ);
  Initialize local trajectory buffer $\mathcal{B}$;
  Receive initial state s_0;
  for $t \leftarrow 0$ *to* $T_{max} - 1$ **do**
    Sample action $\mathbf{a}_t \sim \pi_{\theta_{\text{local}}}(\cdot|s_t)$;
    Execute $\mathbf{a}_t$, get reward r_t and next state s_{t+1};
    Store transition $(s_t, \mathbf{a}_t, r_t, s_{t+1})$ in $\mathcal{B}$;
    $s_t \leftarrow s_{t+1}$;
  Compute advantages $\hat{A}_t$ using GAE for all data in $\mathcal{B}$;
  Compute gradients $\nabla\theta_{\text{local}}$ based on PPO loss over multiple epochs;
  SendGradients($\nabla\theta_{\text{local}}$);
  WaitForLearnerSignal();

Algorithm 3: Distributed PPO: Learner Process

Input : Hyperparameters: learning rate α, update threshold U
Output: Updated global weights θ

Initialize global network π_θ with Orthogonal Initialization and gradient buffer G;
while *training is not complete* **do**
  Wait for gradients $\nabla\theta_{\text{local}}$ from an actor;
  Add gradient to buffer: $G \leftarrow G + \nabla\theta_{\text{local}}$;
  Increment gradient counter C;
  if $C \geq U$ **then**
    Average gradients: $G_{\text{avg}} \leftarrow G/C$;
    Update global weights: $\theta \leftarrow \theta - \alpha G_{\text{avg}}$;
    Signal waiting actors to sync weights;
    Clear buffer G and reset counter $C \leftarrow 0$;

1) Actor's Role: Each of the N_{actor} actors (Alg. 2) operates independently. It first synchronizes its local policy with the Learner's global policy. It then interacts with its own environment instance to collect a trajectory, computes gradients locally based on this experience, and sends these gradients to the Learner. Finally, it waits for a signal from the Learner before starting the next cycle.

2) Learner's Role: The Learner (Alg. 3) orchestrates the training. It asynchronously aggregates gradients from Actors. Once enough gradients are collected, it updates the global policy and signals waiting Actors to sync. The process terminates once all Actors have completed their training episodes.

This division of labor allows for continuous, high-throughput data collection and efficient training.

V. EXPERIMENTS AND ANALYSES

In this section, we conduct a series of experiments to rigorously evaluate the performance of our proposed MM-DEGAT framework. We first introduce the experimental setup and then present a unified quantitative analysis that compares our model against several baselines and validates our core design choices through ablation results.

A. Experimental Setup

1) Simulation Environment: The experiments were conducted on a self-developed UAV communication coverage simulation platform. The simulation environment is configured with key parameters detailed in Table I, defining the physical, communication, and energy dynamics of the scenario.

2) Baseline and Ablation Models: To comprehensively evaluate our framework, we compare it against a heuristic baseline and several DRL-based variants, which also serve as ablation models to validate our design:

- **Greedy**: A rule-based policy where UAVs move towards the center of the largest nearby user cluster and fly to the nearest charging station below a 30% battery threshold.
- **CNN-PPO**: A model using a convolutional neural network (CNN) [23] that processes only image-based state information. The CNN architecture is identical to the one used in MM-DEGAT's image processing branch.
- **DEGAT-PPO**: A model that uses the same DEGAT architecture as our proposed method to process graph

TABLE I
SIMULATION PARAMETER CONFIGURATION

Parameter	Value
Operational Area Dimensions	1000×1000 m^2
Number of Charging Stations (M)	4
UAV Coverage Range (R_{cov})	350 m
Maximum UAV Speed ($V_{\max}$)	20 m/s
Charging Station Range (R_{cg})	80 m
Charging Threshold	20% of battery capacity
Time Slot Duration (Δt)	20 s
Movement Phase (Δt_m)	10 s
Service Phase (Δt_s)	10 s
Maximum Episode Length (T_{max})	500 time slots

information but omits the multimodal fusion module, relying solely on the graph data.

- **GCN-PPO**: A single-modality baseline using only the graph-based observation, processed by a standard Graph Convolutional Network (GCN) [28].

3) Evaluation Metrics and Implementation Details: The system performance is evaluated using the two primary KPIs defined in Section III-E: System Communication Efficiency ($\mathcal{E}$) and Jain's Fairness Index ($\mathcal{F}$). The hyperparameters used for training all PPO-based agents are listed in Table II. All reported results are averaged over 10 independent runs to ensure statistical significance.

TABLE II
TRAINING HYPERPARAMETER CONFIGURATION

Parameter	Value
Learning rate (α)	1×10^{-4}
Batch size (B)	64
PPO epoch (E)	3
Clipping parameter (ϵ)	0.2
Discount factor (γ)	0.99
GAE parameter (λ)	0.95
Maximum gradient norm	250
Number of parallel environments	15
Maximum training episodes	2500

B. Results and Analysis

First, we validate the effectiveness of our distributed training framework in Fig. 3, which compares the training process of the MM-DEGAT model with and without the distributed PPO implementation. The curves clearly demonstrate that the distributed approach not only accelerates convergence but also enables the agent to achieve a higher final average reward. This result highlights that leveraging parallel environments is crucial for efficiently exploring the vast state-action space and learning a more optimal policy. Having established the efficacy of our training methodology, we proceed to analyze the performance of the trained models.

The comprehensive performance results are presented in Table III and Table IV. The data consistently validates the superiority of our model and the effectiveness of its core design components across different scalability dimensions.

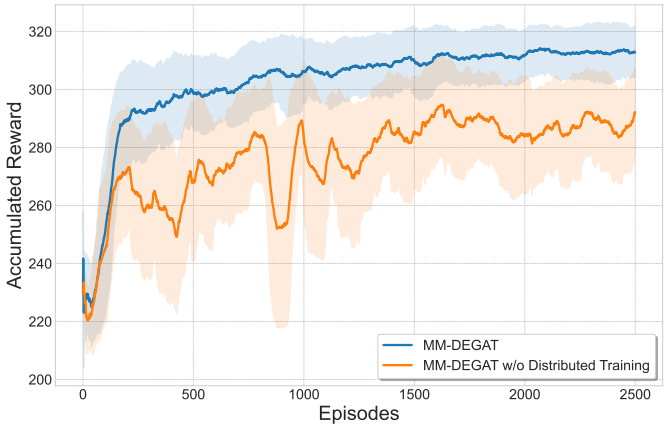

Fig. 3. Accumulated reward curves in training process.

TABLE III
PERFORMANCE COMPARISON FOR DIFFERENT NUMBERS OF UAVS (100 USERS)

Method	3 UAVs			5 UAVs		
	Fair.	Eff.	Comp.	Fair.	Eff.	Comp.
MM-DEGAT	**0.840**	0.5035	**0.672**	**0.907**	0.4925	**0.700**
DEGAT-PPO	0.792	0.4892	0.641	0.884	0.4545	0.669
CNN-PPO	0.599	**0.5439**	0.571	0.892	**0.5031**	0.698
GCN-PPO	0.259	0.3757	0.317	0.371	0.3135	0.342
Greedy	0.188	0.2421	0.215	0.192	0.1500	0.171

1) Overall Performance and Ablation Analysis: Across the tested scenarios, our full MM-DEGAT model demonstrates a superior balance between communication efficiency and fairness. In the representative 4-UAV, 100-user scenario, MM-DEGAT achieves a composite score of 0.702, representing a 5.2% improvement over the next-best DRL method (CNN-PPO) and a remarkable 275% improvement over the Greedy baseline. This confirms the significant value of DRL for this complex planning problem.

The effectiveness of our core architectural designs is validated through ablation. The performance gap between MM-DEGAT and its graph-only counterpart, DEGAT-PPO, highlights the value of multimodal fusion; in the 4-UAV, 100-user scenario, adding image data improves the composite score by 12.1%. Furthermore, DEGAT-PPO's consistent outperfor-

TABLE IV
PERFORMANCE COMPARISON FOR DIFFERENT NUMBERS OF USERS (4 UAVS)

Method	100 Users			150 Users		
	Fair.	Eff.	Comp.	Fair.	Eff.	Comp.
MM-DEGAT	**0.912**	**0.4911**	**0.702**	**0.867**	**0.5403**	**0.704**
DEGAT-PPO	0.780	0.4726	0.626	0.714	0.5210	0.618
CNN-PPO	0.849	0.4857	0.667	0.828	0.5095	0.669
GCN-PPO	0.224	0.2355	0.230	0.246	0.4068	0.326
Greedy	0.190	0.1846	0.187	0.206	0.2842	0.245

mance of GCN-PPO underscores the advanced capabilities of our proposed graph encoder.

Notably, in scenarios with lower complexity, such as the 3-UAV/100-user case, the single-modality CNN-PPO achieves the highest raw efficiency. We attribute this to the simpler model learning a myopic policy based purely on visual cues. However, this comes at a significant cost to fairness and composite performance, where MM-DEGAT's balanced approach proves superior. This is evidenced by its top-tier fairness and composite scores in nearly every scenario.

2) Scalability and Robustness: The scalability of our framework was evaluated by varying both the number of UAVs (Table III) and ground users (Table IV). When increasing the number of agents, MM-DEGAT maintains its performance advantage. In the most complex user scenario (4 UAVs, 150 users), our model achieves a composite score of 0.704, outperforming the next-best baseline by 5.2%.

More importantly, while all methods face challenges in maintaining fairness in denser environments, MM-DEGAT preserves the highest fairness index of 0.867 in the 150-user case, which is 4.7% higher than the runner-up. This showcases its robustness and ability to make globally-aware decisions in congested and complex situations.

VI. CONCLUSION

This paper proposes MM-DEGAT, a novel deep reinforcement learning framework for optimizing multi-UAV communication coverage. The framework maximizes user throughput and service fairness while minimizing energy consumption by fusing spatial image data with topological graphs. Our DEGAT layer processes communication graphs more effectively through edge-aware attention mechanisms. Further, a distributed PPO scheme decouples data collection (parallel actors) and policy learning (centralized learner) to build a scalable training pipeline. Experiments confirm the method's superiority, demonstrating enhanced communication efficiency and service fairness. Future work will include validating our framework in real-world experiments and extending it to larger-scale dynamic scenarios, such as those involving mobile ground users or a greater number of UAVs.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62202321 and 62502331), China Postdoctoral Science Foundation (Grant No. 2025M771496), and Jiangsu Funding Program for Excellent Postdoctoral Talent.

REFERENCES

[1] M. Mozaffari *et al.*, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[2] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[3] X. Zhu *et al.*, "Path planning of multi-uavs based on deep q-network for energy-efficient data collection in uavs-assisted iot," *Vehicular Communications*, vol. 36, p. 100491, 2022.

[4] Y. Wang *et al.*, "Computation offloading optimization for uav-assisted mobile edge computing: a deep deterministic policy gradient approach," *Wireless Networks*, vol. 27, no. 4, pp. 2991–3006, 2021.

[5] B. Li *et al.*, "Energy-efficient task offloading and trajectory planning in uav-enabled mobile edge computing networks," *Computer Networks*, vol. 234, p. 109940, 2023.

[6] R. Lowe *et al.*, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[7] B. Yin *et al.*, "Joint optimization of trajectory control, resource allocation, and user association based on drl for multi-fixed-wing uav networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 10, pp. 13 330–13 343, 2024.

[8] L. Li *et al.*, "Multi-uav path planning based on drl for data collection in uav-assisted iot," in *2024 14th International Conference on Information Science and Technology (ICIST)*, 2024, pp. 566–573.

[9] J. Suárez-Varela *et al.*, "Graph neural networks for communication networks: Context, use cases and opportunities," *IEEE Network*, vol. 37, no. 3, pp. 146–153, 2023.

[10] J. Luo *et al.*, "Gnn-based resource allocation for digital twin-enhanced multi-uav radar networks," *IEEE Wireless Communications Letters*, vol. 13, no. 11, pp. 3137–3141, 2024.

[11] M. Marwani and G. Kaddoum, "Graph neural networks approach for joint wireless power control and spectrum allocation," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 717–732, 2024.

[12] C. H. Liu *et al.*, "Distributed and energy-efficient mobile crowdsensing with charging stations by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 130–146, 2021.

[13] A. Al-Hourani *et al.*, "Optimal lap altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[14] Y. Zeng *et al.*, "Energy minimization for wireless communication with rotary-wing uav," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

[15] H. Dai *et al.*, "Optimizing wireless charger placement for directional charging," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[16] ——, "Safe charging for wireless power transfer," *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3531–3544, 2017.

[17] L. Xie *et al.*, "Uav-enabled wireless power transfer: A tutorial overview," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 4, pp. 2042–2064, 2021.

[18] H. Dai *et al.*, "Placing wireless chargers with multiple antennas," *IEEE Transactions on Mobile Computing*, vol. 23, no. 6, pp. 7517–7536, 2024.

[19] M. Ren *et al.*, "Understanding wireless charger networks: Concepts, current research, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 4, pp. 2247–2282, 2025.

[20] H. Dai *et al.*, "Omnidirectional chargability with directional antennas," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 4483–4500, 2024.

[21] ——, "Rose: Robustly safe charging for wireless power transfer," *IEEE Transactions on Mobile Computing*, vol. 21, no. 6, pp. 2180–2197, 2022.

[22] ——, "Charging task scheduling for directional wireless charger networks," *IEEE Transactions on Mobile Computing*, vol. 20, no. 11, pp. 3163–3180, 2021.

[23] A. Krizhevsky *et al.*, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[24] S. Brody *et al.*, "How attentive are graph attention networks?" *arXiv preprint arXiv:2105.14491*, 2021.

[25] Z. Wang *et al.*, "Egat: Edge-featured graph attention network," in *International Conference on Artificial Neural Networks*, 2021, pp. 253–264.

[26] J. Schulman *et al.*, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[27] ——, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[28] B. Jiang *et al.*, "Semi-supervised learning with graph learning-convolutional networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 313–11 320.