Resource Allocation in Fronthaul-Constrained Cell-Free Networks Using Edge-Graph Attention Networks

Xiaoyu Liu, Jian Zhao, *Senior Member, IEEE*, Furao Shen, *Member, IEEE*, Kun Yang, *Fellow, IEEE*, and Sumei Sun, *Fellow, IEEE*

***Abstract*—Cell-free networks represent a transformative approach for 6G communications. However, they are subject to stringent capacity requirements on the fronthaul links. In this study, we investigate the downlink transmission of cell-free networks with limited fronthaul capacity, aiming to jointly optimize the beamforming matrices of access points (APs) and the AP-user equipment (UE) association strategies to maximize the weighted sum rate. We first present a model-based approach leveraging alternating optimization and successive convex approximation. To facilitate practical deployment, we then propose a learning-based approach using edge-graph attention networks (EGAT). Unlike conventional graph attention networks that primarily focus on node features while overlooking crucial edge information, the proposed EGAT approach effectively captures both node and edge interactions, thereby improving learning efficiency. We also introduce the multi-head attention mechanism and residual connections to improve the expression ability of the model and to avoid over-smoothing, respectively. Moreover, we solve the constrained optimization problem through primal-dual training and adopt a policy gradient scheme to obtain the AP-UE association. Simulations show that the proposed EGAT approach achieves superior UE rate performance with very low inference time. Notably, the proposed EGAT demonstrates stronger generalization capability across network sizes compared to conventional graph neural networks, thanks to its edge-attention mechanism.**

***Index Terms*—Graph neural networks, resource allocation, cell-free networks, beamforming, capacity-limited fronthaul.**

Received 5 August 2025; revised 24 December 2025; accepted 22 February 2026. Date of current version 10 March 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62276127, Grant 62531008, and Grant 62132004; in part by Jiangsu Provincial Key Research and Development Program under Grant BE2023031; in part by the SRIBD Innovation Fund under Grant SIF20240012; in part by Jiangsu Major Project on Fundamental Research under Grant BK20243059; in part by Gusu Innovation Project under Grant ZXL2024360; and in part by the High-Tech District of Suzhou City under Grant RC2025001. The associate editor coordinating the review of this article and approving it for publication was H. Zeng. *(Corresponding author: Jian Zhao.)*

Xiaoyu Liu and Jian Zhao are with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China (e-mail: xiaoyu_liu@smail.nju.edu.cn; jianzhao@nju.edu.cn).

Furao Shen is with the State Key Laboratory of Novel Software Technology and the School of Artificial Intelligence, Nanjing University, Nanjing 210023, China (e-mail: frshen@nju.edu.cn).

Kun Yang is with the State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210023, China, and also with the Institute of Intelligent Networks and Communications (NINE), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the School of Intelligent Software and Engineering, Nanjing University, Suzhou Campus, Suzhou 215163, China (e-mail: kunyang@nju.edu.cn).

Sumei Sun is with the Institute for Infocomm Research, A*STAR, Singapore 138632 (e-mail: sunsm@a-star.edu.sg).

Digital Object Identifier 10.1109/TWC.2026.3669944

I. Introduction

THE global deployment of 5G networks and advancing 6G research have spurred widespread adoption of wireless devices and services, resulting in increasingly complex communication networks [1]. This leads to ultra-dense deployments where a large amount of data traffic competes for limited wireless resources [2]. To address these issues, there have been numerous studies on wireless resource management aiming to efficiently allocate limited resources in wireless networks. These studies address different areas of wireless resource management, such as power control, link scheduling, cell association, subcarrier allocation, and beamforming design [3], [4], [5], [6].

Cell-free networks have emerged as a compelling architecture for next-generation wireless networks, aiming to provide uniformly high-quality service to all user equipments (UEs) without relying on conventional cell boundaries [7]. In this architecture, a large number of distributed access points (APs) are connected via fronthaul links to a central processing unit (CPU), and the APs collaboratively serve UEs through joint transmission and reception. The fronthaul links support data transmission and control signaling between distributed APs and the CPU. By leveraging spatial diversity and macro-diversity gains, cell-free networks can significantly enhance spectral efficiency and system throughput.

To make cell-free networks practically deployable, the user-centric paradigm has been introduced [8], where each UE is served by a subset of APs rather than by all APs in the network. This approach lowers signaling overhead, and alleviates fronthaul congestion. Under the user-centric paradigm, efficient AP selection becomes essential to balance performance and resource usage. The design of AP-UE association strategies under fronthaul constraints is a key problem that directly affects network efficiency and scalability.

A. Prior Studies

1) Cell-Free Networks: Since the proposal of cell-free networks, there have been numerous studies on resource allocation problems [9], [10], [11], [12], [13]. In particular, the authors of [14] proposed a heuristic approach to design beamforming vectors based on the uplink and downlink duality, and it outperforms the conventional maximum ratio processing in terms of performance. The authors of [15] compared conjugate

beamforming and zero-forcing schemes for max-min power control in cell-free networks, and developed a near-optimal power control algorithm for zero-forcing schemes. The authors of [16] derived a closed-form expression for the spectral efficiency in cell-free networks and designed an optimal power allocation algorithm to maximize the overall energy efficiency. They also proposed an AP selection scheme where each UE selects a subset of APs to reduce the power consumption caused by fronthaul links.

Fronthaul plays a key role in modern wireless networks, and fronthaul constraints have been a limiting factor for 5G and 6G communications. The authors of [17] considered the problems of maximizing the energy efficiency with limited fronthaul capacity in a cell-free network using coherent or non-coherent transmissions. The authors of [18] considered the selection of small cell APs in a heterogeneous network with wireless backhaul. They proposed a near-optimum iterative algorithm for the beamforming design at the wireless backhaul hub.

The authors of [19] proposed robust beamforming strategies in a reconfigurable intelligent surface aided cell-free system with channel information uncertainty and capacity-limited backhaul. The authors of [20] considered the joint deployment and resource optimization problems in an uncrewed-aerial-vehicle (UAV)-assisted wireless network subject to the constraints of wireless backhaul between the ground base station and UAVs.

2) Applications of Graph Neural Networks in Wireless Communications: Graph neural networks (GNNs) have gained growing interest in wireless networks due to their ability to exploit topological information. By modeling networks as graphs, GNNs encode system parameters (e.g., signal strength, power limits) as input features and optimization variables (e.g., power allocation, routing decisions) as outputs. In [21], GNNs were applied to large-scale radio resource management, showing that such problems can be formulated as permutation-equivariant (PE) graph optimizations, and a message-passing GNN was introduced. Subsequently, [22] proposed the random edge GNN (REGNN), which performs convolutions over random graphs formed by fading interference patterns, retaining PE and outperforming heuristic benchmarks. Similarly, homogeneous GNNs have also been shown to generalize well on different numbers of UEs in link scheduling [23], power control [24], [25], joint beam selection and link activation [26]. Moreover, graph attention networks (GATs) assign adaptive weights to neighboring nodes through attention mechanisms and have demonstrated strong performance across a wide range of tasks [27], [28], [29].

With the progress in research for GNNs, heterogeneous GNNs have shown superiority over homogeneous GNNs in the application of resource management [30], [31]. In the design of beamforming for downlink multiuser multiple-input and multiple-output (MIMO) communication systems, heterogeneous GNNs have demonstrated good adaptability [32]. The authors of [33] proposed the edge-node GNN (ENGNN), which incorporates a new updating mechanism for both node and edge features. The authors demonstrate that ENGNN outperforms the weighted minimum mean square error (WMMSE) algorithm in collaborative beamforming design and it generalizes well in different system settings. However, attention mechanisms specifically designed for edge updating have not yet been thoroughly studied.

Most of these previous works on learning-based resource allocation only consider problems with simple constraints. Recently, schemes based on Lagrange duality have been applied to solve constrained optimization problems using GNNs. The authors of [22] proposed an unsupervised and model-free primal-duality learning algorithm to train REGNN, and proved that it can obtain superior performance and migration ability compared with the benchmark algorithms. The authors of [34] employed the primal-dual learning algorithm to solve a resilient radio resource management (RRM) problem, where learnable slack variables are introduced to relax the minimum capacity constraints for UEs that adapt to the underlying network conditions.

B. Motivation and Contribution

Motivated by the above discussions, this work investigates the joint optimization of AP-UE association strategies and downlink beamforming design in a user-centric cell-free network, subject to constraints on fronthaul capacity and transmit power. The key contributions of this paper are summarized as follows:

- We study the joint optimization of AP-UE association and beamforming in a user-centric cell-free network to maximize the weighted sum rate under fronthaul capacity and power constraints. To address this, we reformulate the problem using fractional programming and solve it via an alternating optimization framework combined with successive convex approximation (SCA). This model-based approach provides a performance benchmark for our considered problem.
- To further enhance the algorithmic efficiency and performance, we propose a novel learning-based approach using edge-graph attention networks (EGAT). Unlike conventional GATs that focus primarily on node features, our EGAT approach incorporates edge features by introducing additional edge attention coefficients and updated edge feature propagation rules. Furthermore, we adopt a multi-head attention mechanism to improve the expression ability of the model, and introduce residual connections to avoid over-smoothing.
- As for the complicated constraints of the considered problem, we first integrate the fronthaul constraints into the objective function using the Lagrangian duality. Then, we employ a policy gradient approach to handle the binary AP-UE association constraints by introducing penalty terms in the loss function, ensuring effective optimization.
- Simulation results validate the effectiveness of the proposed approaches. During the inference stage, the proposed EGAT method only incurs very low computational time. Meanwhile, EGAT generally outperforms other benchmark methods in terms of weighted sum rate under various system settings. Notably, compared to conventional GNN-based methods, the proposed EGAT method demonstrates superior generalization capability with respect to both the number of APs and UEs.

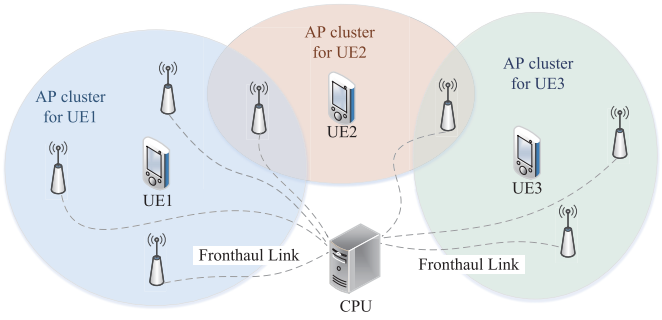


Fig. 1. System model of user-centric cell-free wireless networks.

C. Organization and Notations

The rest of this paper is organized as follows. Section II introduces the system model and formulates the optimization problem. Section III presents the model-based approach to address the formulated problem. The proposed learning-based approach using EGAT is presented in Section IV. Section V provides numerical results comparing different algorithms. Section VI draws the conclusion of this paper.

Notations: In this paper, we use bold lowercase letters and bold uppercase letters to represent vectors and matrices, respectively. $\mathbb{C}^M$ denotes an M-dimensional complex vector and $\mathbb{C}^{M\times N}$ represents an $M \times N$ dimensional complex matrix. $|\cdot|$ and $\mathrm{Tr}\,(\cdot)$ represent the determinant and trace of a matrix, respectively. $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and conjugate transpose of a matrix or vector, respectively. $\mathcal{CN}\,(\mathbf{0}, \mathbf{I}_N)$ denotes a complex normal random vector with mean $\mathbf{0}$ and covariance $\mathbf{I}_N$. $\Re\{\cdot\}$ and $\Im\{\cdot\}$ return the real and imaginary components of a complex argument, respectively. $\mathbb{E}\,[\cdot]$ denotes the expectation of random variables, and $[\cdot]_+$ denotes $\max\,(\cdot, 0)$.

II. System Model and Problem Formulation

A. System Model

We consider the downlink transmission in a user-centric cell-free communication network with B APs and K UEs as shown in Fig. 1. Each AP is equipped with M antennas, while each UE has one receive antenna. Mathematically, let $\mathcal{B} \triangleq \{1, \ldots, B\}$ denote the AP set and $\mathcal{K} \triangleq \{1, \ldots, K\}$ denote the UE set. The APs jointly transmit signals to the UEs using the same time-frequency resource. A CPU is responsible for data processing, computing, and subsequently distributing the UE data to each AP via the fronthaul links. However, due to the finite fronthaul capacity, the UE data rate that each AP can support is limited. Furthermore, the user-centric approach requires a serving cluster to be constructed separately for each UE [35]. Consequently, optimizing AP-UE association and beamforming design under constrained fronthaul capacity are critical challenges in the system design.

We denote the channel from the bth AP to the kth UE as $\mathbf{h}_{bk} \in \mathbb{C}^M, \forall b \in \mathcal{B}, k \in \mathcal{K}$. The transmitted signal vector for all the UEs is denoted as $\mathbf{s} \in \mathbb{C}^K$, where $\mathbb{E}\left[\mathbf{s}\mathbf{s}^H\right] = \mathbf{I}_K$. Let $\mathbf{v}_{bk} \in \mathbb{C}^M$ be the beamforming vector at the bth AP for the kth UE. The signal received at the kth UE is given by

$$r_k = \sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bk} s_k + \sum_{j\in\mathcal{K}\setminus k} \sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bj} s_j + n_k. \tag{1}$$

Here n_k is the additive white Gaussian noise (AWGN) at the kth UE where $n_k \sim \mathcal{CN}\left(0, \xi_k^2\right)$ and ξ_k^2 denotes the noise power.

To represent the association between the APs and the UEs, we introduce an association matrix $\mathbf{X}$, whose (b, k)th element $x_{bk} \in \{0, 1\}$. $x_{bk} = 1$ means that the bth AP serves the kth UE, otherwise we have $x_{bk} = 0$. Moreover, if $x_{bk} = 0$, we require that $\mathbf{v}_{bk} = \mathbf{0}$. Therefore, the received signal-to-interference-plus-noise ratio (SINR) at the kth UE can be expressed as

$$\mathrm{SINR}_k = \frac{\left|\sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bk}\right|^2}{\sum_{j\in\mathcal{K}\setminus k} \left|\sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bj}\right|^2 + \xi_k^2}. \tag{2}$$

B. Problem Formulation

Our objective is to maximize the overall weighted sum rate of all the UEs by jointly optimizing the beamforming at each AP and the AP-UE association matrix subject to the constraints of the fronthaul capacity and transmission power. Therefore, the optimization problem can be formulated as

$$\max_{\mathbf{V},\mathbf{X}} \quad \sum_{k=1}^{K} w_k \log\left(1 + \mathrm{SINR}_k\right), \tag{3a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} x_{bk} \log\left(1 + \mathrm{SINR}_k\right) \le C_b^{\max}, \ \forall b \in \mathcal{B}, \tag{3b}$$

$$\|\mathbf{V}_b\|_F^2 \le P_b^{\max}, \ \forall b \in \mathcal{B}, \tag{3c}$$

$$x_{bk} \in \{0, 1\}, \ \forall b \in \mathcal{B}, k \in \mathcal{K}, \tag{3d}$$

where $C_b^{\max}$ and $P_b^{\max}$ denote the capacity limit of the fronthaul link from the CPU to the bth AP and the maximum transmit power of the bth AP, respectively. w_k denotes the weight for the kth UE data rate. $\mathbf{V}_b = [\mathbf{v}_{b1}, \ldots, \mathbf{v}_{bK}] \in \mathbb{C}^{M\times K}$ denotes the beamforming matrix at the bth AP and $\mathbf{V} = [\mathbf{V}_1^T, \ldots, \mathbf{V}_B^T]^T \in \mathbb{C}^{MB\times K}$. The UE data is distributed from the CPU to the APs before AP transmission. Constraint (3b) ensures that the total UE data rate transmitted by the bth AP does not exceed its fronthaul capacity. Constraint (3c) shows the transmit power constraint for the bth AP. Additionally, constraint (3d) ensures that the elements of the association matrix $\mathbf{X}$ are binary. Problem (3) is a mixed integer nonlinear programming (MINLP) problem, which is non-convex and NP-hard. In the following, we propose two approaches to solve problem (3): One is model-based and the other is learning-based.

III. Model-Based Approach for Joint Optimization

We first propose a model-based approach to solve problem (3) using mathematical optimization algorithms. The idea is to first transform problem (3) into an equivalent formulation using quadratic transform for fractional programming (FP) [36], and then solve the problem using alternating optimization

combined with SCA. Therefore, this algorithm is named *AO-SCA*.

A. Quadratic Transform for FP

Because $x_{bk} = 0$ if and only if $\mathbf{v}_{bk} = \mathbf{0}$, the value of x_{bk} can be determined by that of $\mathbf{v}_{bk}$. Therefore, we have the following equality

$$x_{bk} = I(\|\mathbf{v}_{bk}\|) = \begin{cases} 0, & \text{if } \|\mathbf{v}_{bk}\| = 0, \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

Here $I(\cdot)$ denotes the indicator function. Furthermore, we introduce an auxiliary variable γ_k to replace the term SINR_k in (3) and rewrite problem (3) as

$$\max_{\mathbf{V},\boldsymbol{\gamma}} \quad \sum_{k=1}^{K} w_k \log\left(1+\gamma_k\right), \tag{5a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} I(\|\mathbf{v}_{bk}\|) \log\left(1+\gamma_k\right) \le C_b^{\max}, \ \forall b \in \mathcal{B}, \tag{5b}$$

$$\gamma_k \le \frac{\left|\sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bk}\right|^2}{\sum_{j \in \mathcal{K}\setminus k} \left|\sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bj}\right|^2 + \xi_k^2}, \forall k \in \mathcal{K}, \tag{5c}$$

$$\text{(3c)}, \tag{5d}$$

where $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_K]^T$.

We introduce the dual variable λ_k for each inequality constraint in (5c). The partial Lagrangian can be written as

$$\mathcal{L}_1(\mathbf{V}, \boldsymbol{\gamma}, \boldsymbol{\lambda}) = \sum_{k=1}^{K} \Bigg[w_k \log\left(1+\gamma_k\right) - \lambda_k \left(\gamma_k - \frac{\left|\sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bk}\right|^2}{\sum_{j \in \mathcal{K}\setminus k} \left|\sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bj}\right|^2 + \xi_k^2} \right) \Bigg]. \tag{6}$$

Let $(\boldsymbol{\gamma}^\star, \boldsymbol{\lambda}^\star)$ be the saddle point of the dual problem of (5) for given values of $\mathbf{V}$. According to the first order condition $\partial \mathcal{L}_1 / \partial \gamma_k = 0$ and the SINR expression (2), we have

$$\lambda_k^\star = w_k \frac{\sum_{j \in \mathcal{K}\setminus k} \left|\sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bj}\right|^2 + \xi_k^2}{\sum_{j=1}^{K} \left|\sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bj}\right|^2 + \xi_k^2}, \ \forall k \in \mathcal{K}. \tag{7}$$

Therefore, problem (5) can be reformulated as

$$\max_{\mathbf{V},\boldsymbol{\gamma}} \quad \mathcal{L}_1(\mathbf{V}, \boldsymbol{\gamma}, \boldsymbol{\lambda}^\star), \tag{8a}$$

$$\text{s.t.} \quad \text{(5b)}, \text{(3c)}. \tag{8b}$$

Using the quadratic transform technique [36], we can rewrite (8) in an equivalent form as follows

$$\max_{\mathbf{V},\mathbf{y},\boldsymbol{\gamma}} \quad f(\mathbf{V}, \boldsymbol{\gamma}, \mathbf{y}), \tag{9a}$$

$$\text{s.t.} \quad \text{(5b)}, \text{(3c)}, \tag{9b}$$

where $f(\mathbf{V}, \boldsymbol{\gamma}, \mathbf{y})$ is shown in (10), at the bottom of the page, and $\bar{y}_k$ denotes the complex conjugate of y_k.

B. Alternating Optimization With SCA

In this subsection, we use alternating optimization method combined with SCA to solve problem (9) or its equivalent form (8). The alternating optimization method divides the variables into several blocks and optimizes each block while fixing the values of variables in the other blocks.

1) Update of $\boldsymbol{\gamma}$ With Fixed $\mathbf{V}$: In this case, we consider the form of (8) with fixed beamforming matrix $\mathbf{V}$. Unlike the derivations in [36], the constraint in (5b) is non-convex because the log function is concave in γ_k. Therefore, we cannot obtain closed form solutions in this case, and we propose to replace $\log(1+\gamma_k)$ with its first-order approximation as the surrogate function, i.e.,

$$\log\left(1+\gamma_k\right) \le \log\left(1+\hat{\gamma}_k\right) + \frac{\gamma_k - \hat{\gamma}_k}{1+\hat{\gamma}_k} \triangleq z_1(\gamma_k; \hat{\gamma}_k) \tag{11}$$

where $\hat{\gamma}_k$ denotes a given value of γ_k. By substituting the log function in (5b) with $z_1(\gamma_k; \hat{\gamma}_k)$, we can approximate (8) by the following

$$\max_{\boldsymbol{\gamma}} \quad \mathcal{L}_1(\mathbf{V}, \boldsymbol{\gamma}, \boldsymbol{\lambda}^\star), \tag{12a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} I\big(\|\mathbf{v}_{bk}\|\big) z_1(\gamma_k; \hat{\gamma}_k) \le C_b^{\max}, \ \forall b \in \mathcal{B}. \tag{12b}$$

Problem (12) is now a convex problem and can be solved using convex optimization software [37].

2) Update of $\mathbf{y}$ With Fixed $\mathbf{V}$, $\boldsymbol{\gamma}$: When the values of variables $\mathbf{V}$, $\boldsymbol{\gamma}$ are fixed, problem (9) is a convex problem about $\mathbf{y}$. The optimal solution of $\mathbf{y}$ can be obtained by setting $\partial f(\mathbf{V}, \boldsymbol{\gamma}, \mathbf{y}) / \partial \mathbf{y} = 0$. Therefore, we obtain

$$y_k^\star = \frac{\sqrt{w_k\left(\gamma_k+1\right)} \sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bk}}{\sum_{j=1}^{K} \left|\sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bj}\right|^2 + \xi_k^2}. \tag{13}$$

3) Update of $\mathbf{V}$ With Fixed $\mathbf{y}$, $\boldsymbol{\gamma}$: In this case, problem (9) is difficult to solve because the indicator function $I(\cdot)$ in (5b) is non-smooth. We propose to approximate the indicator function $I(\cdot)$ using the following smooth concave function [38]

$$I(\|\mathbf{v}_{bk}\|) \approx \frac{\log\left(\|\mathbf{v}_{bk}\|/\delta + 1\right)}{\log\left(1/\delta + 1\right)}, \tag{14}$$

where δ is a small positive constant to provide a smooth transition as $\|\mathbf{v}_{bk}\|$ approaches zero.

After replacing the indicator function using (14), problem (9) is still non-convex because the left-hand-side (LHS) of constraint (5b) is concave. Therefore, we propose to replace

$$f(\mathbf{V}, \boldsymbol{\gamma}, \mathbf{y}) = \sum_{k=1}^{K} \Bigg[w_k \log\left(1+\gamma_k\right) - w_k \gamma_k - |y_k|^2 \left(\sum_{j=1}^{K} \left|\sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bj}\right|^2 + \xi_k^2 \right) + 2\sqrt{w_k\left(\gamma_k+1\right)} \Re\left(\bar{y}_k \sum_{b=1}^{B} \mathbf{h}_{bk}^H \mathbf{v}_{bk} \right) \Bigg]. \tag{10}$$

Algorithm 1 AO-SCA Algorithm to Solve Problem (9)

1: **Initialization**: Choose feasible values for $\mathbf{V}^{(0)}$ and $\boldsymbol{\gamma}^{(0)}$. Choose accuracy $\varepsilon > 0$, and set iteration index $t = 0$.
2: **repeat**
3: Set $t = t + 1$;
4: Update $\boldsymbol{\gamma}^{(t)}$ by solving (12) with $\mathbf{V} = \mathbf{V}^{(t-1)}$ and $\hat{\boldsymbol{\gamma}} = \boldsymbol{\gamma}^{(t-1)}$;
5: Update $\mathbf{y}^{(t)}$ according to (13) with $\mathbf{V} = \mathbf{V}^{(t-1)}$ and $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{(t)}$;
6: Update $\mathbf{V}^{(t)}$ by solving (16) with $\mathbf{y} = \mathbf{y}^{(t)}$, $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{(t)}$ and $\hat{\mathbf{V}} = \mathbf{V}^{(t-1)}$;
7: **until** $|f(\mathbf{V}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\gamma}^{(t)}) - f(\mathbf{V}^{(t-1)}, \mathbf{y}^{(t-1)}, \boldsymbol{\gamma}^{(t-1)})| < \varepsilon$ or maximum number of iterations is reached.
8: **Output**: The solution for $\mathbf{V}$, $\boldsymbol{\gamma}$, $\mathbf{y}$.

the function $\log(\|\mathbf{v}_{bk}\|/\delta + 1)$ in (14) with its first-order approximation as the surrogate function, i.e.,

$$\log(\|\mathbf{v}_{bk}\|/\delta + 1) \leq \log(\|\hat{\mathbf{v}}_{bk}\|/\delta + 1) + \frac{(\|\mathbf{v}_{bk}\| - \|\hat{\mathbf{v}}_{bk}\|)}{(\|\hat{\mathbf{v}}_{bk}\| + \delta)} \triangleq z_2(\mathbf{v}_{bk}; \hat{\mathbf{v}}_{bk}), \tag{15}$$

where $\hat{\mathbf{v}}_{bk}$ denotes a given value of $\mathbf{v}_{bk}$.

For fixed $\boldsymbol{\gamma}$, $\mathbf{y}$, we substitute the log functions in (14) with its first-order approximation (15). Therefore, problem (9) can be approximated by

$$\max_{\mathbf{V}} \quad f(\mathbf{V}, \boldsymbol{\gamma}, \mathbf{y}) \tag{16a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \frac{\log(1+\gamma_k)}{\log(1/\delta+1)} \cdot z_2(\mathbf{v}_{bk}; \hat{\mathbf{v}}_{bk}) \leq C_b^{\max}, \ \forall b \in \mathcal{B}, \tag{16b}$$

$$\text{(3c)}. \tag{16c}$$

Problem (16) is now a convex problem and can be solved using convex optimization software.

The overall AO-SCA algorithm to solve problem (9) is shown in **Algorithm** 1. Starting from a feasible point, **Algorithm** 1 alternately optimizes the variables $\boldsymbol{\gamma}$, $\mathbf{y}$, and $\mathbf{V}$.

C. Convergence and Complexity Analysis

The convergence of the AO-SCA algorithm can be proved similar to that in [36]. In each iteration, the algorithm alternately updates the variables $\boldsymbol{\gamma}$, $\mathbf{y}$, and $\mathbf{V}$. For the updates of $\boldsymbol{\gamma}$ and $\mathbf{V}$, we employ the SCA method, where the surrogate functions $z_1(\gamma_k; \hat{\gamma}_k)$ and $z_2(\mathbf{v}_{bk}; \hat{\mathbf{v}}_{bk})$ are locally tight approximations of the original functions at the current points $\hat{\gamma}_k$ and $\hat{\mathbf{v}}_{bk}$, respectively. These properties ensure that each iteration provides a feasible update. Moreover, the alternating optimization framework ensures that the sequence of objective values obtained in each iteration of the AO-SCA algorithm satisfies the monotonic relationship, i.e., $f(\mathbf{V}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\gamma}^{(t)}) \leq f(\mathbf{V}^{(t+1)}, \mathbf{y}^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}))$, because each subproblem yields a feasible solution that does not decrease the objective value. Since the objective function is upper bounded due to the power and fronthaul constraints, the monotone convergence theorem guarantees that the sequence converges. At convergence, the limit point is a stationary point that satisfies the first-order optimality conditions of the original problem.

The complexity per iteration mainly comes from two steps in **Algorithm** 1, i.e., Step 4 and Step 6. Step 4 can be solved by the interior-point method, with complexity $\mathcal{O}(K^{3.5})$. Similarly, the complexity of Step 6 is $\mathcal{O}((BKM)^3)$. Therefore, the total computational complexity of the AO-SCA algorithm is $\mathcal{O}(T_{\text{iter}}(K^{3.5} + (BKM)^3))$, where T_{iter} is the number of iterations.

In each iteration of the AO-SCA algorithm, we must solve the sub-problems using convex optimization algorithms, e.g., interior-point methods. Moreover, the model-based approach needs to re-solve the problem whenever the channel changes. To improve algorithm performance and enhance its practicality, we propose a learning-based approach to solve the problem in the following section.

IV. Learning-Based Approach for Joint Optimization

In this section, we propose a GNN-based architecture called EGAT to solve problem (3). The standard GAT framework is primarily designed for graphs with only node features, making it unsuitable for our scenario. To address this limitation, EGAT enhances the traditional GAT by incorporating edge features, introducing additional edge attention coefficients, and implementing updated edge feature propagation rules, thereby improving its effectiveness in our problem setting.

A. Graph Representation

We model our considered network as a heterogeneous graph, where the APs and UEs are considered as two types of nodes, namely AP-nodes and UE-nodes. Moreover, an edge connects an AP-node and a UE-node if there is a communication link between them. Since we focus on the downlink data transmission, all edges in the heterogeneous graph are directed from the AP-node to the UE-node. Therefore, the heterogeneous graph can be represented as $\mathcal{G} = \{\mathcal{B}, \mathcal{K}, \mathcal{E}\}$, where $\mathcal{B} = \{1, \ldots, B\}$, $\mathcal{K} = \{1, \ldots, K\}$ and $\mathcal{E} \subseteq \{(b,k)\}_{b\in\mathcal{B}, k\in\mathcal{K}}$ denote the set of AP-nodes, UE-nodes and edges, respectively.

Each node and edge in the heterogeneous graph is associated with specific features and variables. We denote the feature on the bth AP node, the kth UE node and the (b,k)th edge as $\mathbf{f}_{\text{AP},b} \in \mathbb{R}^{d_{\text{AP}}}$, $\mathbf{f}_{\text{UE},k} \in \mathbb{R}^{d_{\text{UE}}}$ and $\mathbf{e}_{b,k} \in \mathbb{R}^{d_{\text{e}}}$, respectively, where d_{AP}, d_{UE} and d_{e} denote the corresponding feature dimensions. Then the feature matrix of the AP nodes, the UE nodes, and the edges can be expressed as $\mathbf{F}_{\text{AP}} = [\mathbf{f}_{\text{AP},1}, \ldots, \mathbf{f}_{\text{AP},B}]^T \in \mathbb{R}^{B\times d_{\text{AP}}}$, $\mathbf{F}_{\text{UE}} = [\mathbf{f}_{\text{UE},1}, \ldots, \mathbf{f}_{\text{UE},K}]^T \in \mathbb{R}^{K\times d_{\text{UE}}}$, and $\mathbf{E} \in \mathbb{R}^{B\times K\times d_{\text{e}}}$, respectively. In our problem, the edge variables are the unknown beamforming matrix $\mathbf{V}$ and the AP-UE association matrix $\mathbf{X}$.

B. GNN Structure Design

The overall structure of the proposed GNN structure is shown in Fig. 2. The EGAT model consists of a preprocessing layer, L update layers and a postprocessing layer. Firstly, the preprocessing layer transforms the initial node features and edge features into real-valued formats suitable for neural

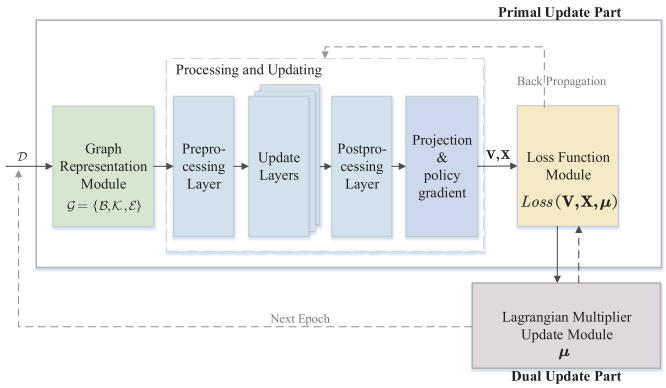


Fig. 2. The EGAT model with primal-dual training.

network processing. We set the maximum transmit power budget $\mathbf{p}^{\max}=[P_1^{\max},\ldots,P_B^{\max}]^T\in\mathbb{R}^B$ and the maximum fronthaul capacity $\mathbf{c}^{\max}=[C_1^{\max},\ldots,C_B^{\max}]^T\in\mathbb{R}^B$ as the initial features of AP nodes. Specifically, the initial feature on AP node b is $\mathbf{f}_{\text{AP},b}^{\text{init}}=[P_b^{\max},C_b^{\max}]^T\in\mathbb{R}^2$. The initial features on the UE nodes are defined as the noise standard deviation $\boldsymbol{\xi}=[\xi_i,\ldots,\xi_K]^T\in\mathbb{R}^K$ and the weight of the UE rate $\mathbf{w}=[w_1,\ldots,w_k]^T\in\mathbb{R}^K$. Thus, the initial feature for UE node k is $f_{\text{UE},k}^{\text{init}}=[\xi_k,w_k]^T\in\mathbb{R}^2$. Moreover, for the (m,k)th edge features, we separate the channel $\mathbf{h}_{bk}$ into the real and imaginary parts, denoted as $\mathbf{e}_{b,k}^{\text{init}}=[\Re\{\mathbf{h}_{bk}\}^T,\Im\{\mathbf{h}_{bk}\}^T]^T\in\mathbb{R}^{2M}$. To prepare these features for processing, we transform the initial features of the AP nodes, the UE nodes, and the edges using a multilayer perceptron (MLP) with one hidden layer. The transformed features, denoted as $\mathbf{f}_{\text{AP},b}^{(0)}$, $\mathbf{f}_{\text{UE},k}^{(0)}$ and $\mathbf{e}_{b,k}^{(0)}$, serve as the input to the subsequent update layers.

Afterwards, the update layers use node and edge attention mechanisms to process these input features. Finally, the post-processing layer transforms the features $\left(\mathbf{F}_{\text{AP}}^{(L)},\mathbf{F}_{\text{UE}}^{(L)},\mathbf{E}^{(L)}\right)$ into the beamforming matrix and the AP-UE association matrix, i.e., $(\mathbf{V},\mathbf{X})$.

C. Node and Edge Attention Mechanisms

In the update layers, we introduce attention mechanisms to separately update node and edge features as shown in Fig. 3.

1) Node Feature Update Module: In the node feature update module, each node aggregates the features of its neighboring nodes and edges based on the computed attention coefficient for feature update. Specifically, in the lth update layer of EGAT, the inputs and outputs are $(\mathbf{F}_{\text{AP}}^{(l-1)},\mathbf{F}_{\text{UE}}^{(l-1)},\mathbf{E}^{(l-1)})$ and $(\mathbf{F}_{\text{AP}}^{(l)},\mathbf{F}_{\text{UE}}^{(l)},\mathbf{E}^{(l)})$. When updating the features of an AP node, we gather all its neighboring UE nodes along with the corresponding edge features. The attention coefficients for these neighbors are then computed using the attention mechanism. Specifically, for AP node b, the attention coefficient $\alpha_{\text{AP},b,k}^{(l)}$ of its neighboring UE node k and edge (b,k) can be calculated as, (17), shown at the bottom of the page,where $\mathbf{W}_{\text{AP}}\in\mathbb{R}^{d'_{\text{AP}}\times d_{\text{AP}}}$, $\mathbf{W}_{\text{UE}}\in\mathbb{R}^{d'_{\text{UE}}\times d_{\text{UE}}}$, $\mathbf{W}_{\text{e}}\in\mathbb{R}^{d'_e\times d_e}$ are the weight matrix with learnable parameters that map the input features of dimensions $d_{\text{AP}},d_{\text{UE}},d_e$ to new features of dimensions $d'_{\text{AP}},d'_{\text{UE}},d'_e$, which provides the network with higher-order learning capabilities. Here $\mathcal{N}_b$ is the set of neighboring UE nodes of AP node b and "||" denotes the splicing operation that concatenates the transformed node and edge features along the feature dimension. The learnable vector $\mathbf{a}_{\text{AP}}\in\mathbb{R}^{d'_{AP}+d'_e+d'_{\text{UE}}}$ is used to calculate the attention coefficient. $\sigma(\cdot)$ denotes the activation function, where we use $\sigma(x)=\text{LeakyReLU}(x)=\max(0.001x,x)$.

Then, we weight and sum the transformed features of the neighboring nodes and edges of the current AP node using the attention coefficients. The aggregated information can be obtained as

$$\mathbf{f}_{\text{AP},b}^{(l)}=\sum_{k\in\mathcal{N}_b}\alpha_{\text{AP},b,k}^{(l)}\left[\mathbf{W}_e\mathbf{e}_{b,k}^{(l-1)}||\mathbf{W}_{\text{UE}}\mathbf{f}_{\text{UE},k}^{(l-1)}\right]. \tag{18}$$

In order to capture information comprehensively, we employ the *multi-head attention mechanism*. Each attention head independently performs operation (18), mapping the input features to different subspaces through distinct linear transformations to compute the attention coefficients separately. We use $\mathbf{f}_{\text{AP},b,c}^{(l)}$ to represent the result of (18) under the cth attention head. Finally, the aggregated results from all C heads are concatenated. The high-dimensional concatenated features are then linearly transformed using an MLP to reduce the dimensionality back to the original dimension. Moreover, the features of the previous layer can be added to the result, which is called *residual connection* to avoid over-smoothing. So the final updated feature for the lth layer of the bth AP-node is

$$\mathbf{f}_{\text{AP},b}^{(l)}=\text{MLP}_{\text{AP}}\left(\|_{c=1}^{C}\mathbf{f}_{\text{AP},b,c}^{(l)}\right)+\mathbf{f}_{\text{AP},b}^{(l-1)}, \tag{19}$$

where $\|_{c=1}^{C}$ denotes the splicing operation of the aggregated output of C attention mechanisms. MLP_{AP} denotes the MLP for updating the AP node features.

For the feature update of UE nodes, it is also necessary to collect all its neighboring AP nodes and the corresponding edge features, use attention mechanism to calculate the attention coefficients of these neighbors, and sum up the weighted transformation features of the neighboring nodes and edges to obtain C heads aggregated information. After concatenation, dimensionality reduction, and residual connection, the final updated features are obtained. Taking UE node k as an example, the specific calculation process is as follows

$$\alpha_{\text{UE},k,b}^{(l)}=\frac{\exp\left(\sigma\left(\mathbf{a}_{\text{UE}}^T\left[\mathbf{W}_{\text{UE}}\mathbf{f}_{\text{UE},k}^{(l-1)}||\mathbf{W}_e\mathbf{e}_{b,k}^{(l-1)}||\mathbf{W}_{\text{AP}}\mathbf{f}_{\text{AP},b}^{(l-1)}\right]\right)\right)}{\sum_{b'\in\mathcal{N}_k}\exp\left(\sigma\left(\mathbf{a}_{\text{UE}}^T\left[\mathbf{W}_{\text{UE}}\mathbf{f}_{\text{UE},k}^{(l-1)}||\mathbf{W}_e\mathbf{e}_{b',k}^{(l-1)}||\mathbf{W}_{\text{AP}}\mathbf{f}_{\text{AP},b'}^{(l-1)}\right]\right)\right)}, \tag{20}$$

$$\alpha_{\text{AP},b,k}^{(l)}=\frac{\exp\left(\sigma\left(\mathbf{a}_{\text{AP}}^T\left[\mathbf{W}_{\text{AP}}\mathbf{f}_{\text{AP},b}^{(l-1)}||\mathbf{W}_e\mathbf{e}_{b,k}^{(l-1)}||\mathbf{W}_{\text{UE}}\mathbf{f}_{\text{UE},k}^{(l-1)}\right]\right)\right)}{\sum_{k'\in\mathcal{N}_b}\exp\left(\sigma\left(\mathbf{a}_{\text{AP}}^T\left[\mathbf{W}_{\text{AP}}\mathbf{f}_{\text{AP},b}^{(l-1)}||\mathbf{W}_e\mathbf{e}_{b,k'}^{(l-1)}||\mathbf{W}_{\text{UE}}\mathbf{f}_{\text{UE},k'}^{(l-1)}\right]\right)\right)}, \tag{17}$$

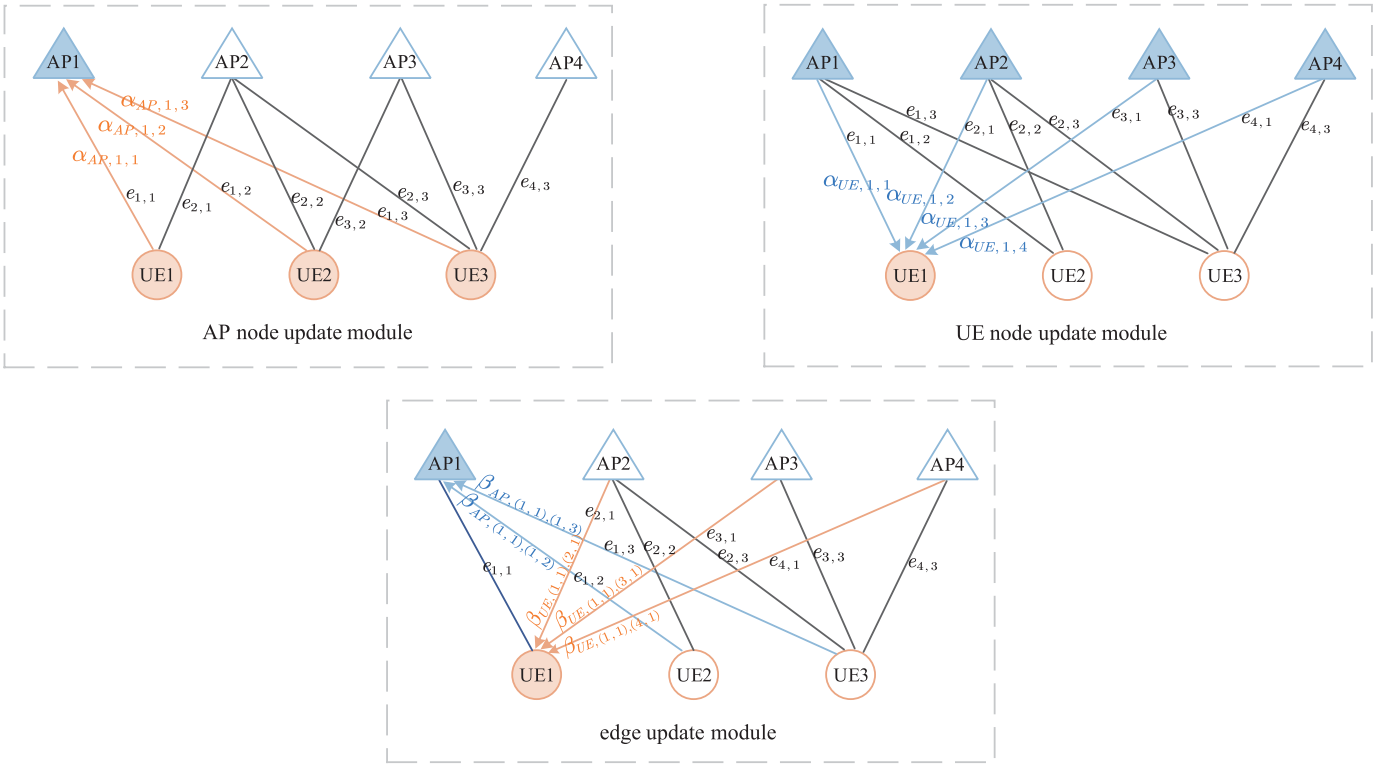


Fig. 3. Node and edge attention mechanisms.

$$\mathbf{f}_{\mathrm{UE},k}^{(l)} = \sum_{b\in\mathcal{N}_k} \alpha_{\mathrm{UE},k,b}^{(l)} \left[\mathbf{W}_e \mathbf{e}_{b,k}^{(l-1)} || \mathbf{W}_{\mathrm{AP}} \mathbf{f}_{\mathrm{AP},b}^{(l-1)}\right], \tag{21}$$

$$\mathbf{f}_{\mathrm{UE},k}^{(l)} = \mathrm{MLP}_{\mathrm{UE}}\left(\|_{c=1}^{C} \mathbf{f}_{\mathrm{UE},k,c}^{(l)}\right) + \mathbf{f}_{\mathrm{UE},k}^{(l-1)}, \tag{22}$$

where $\mathcal{N}_k$ denote the set of AP nodes connected to UE node k and $\mathbf{a}_{\mathrm{UE}} \in \mathbb{R}^{d'_{UE}+d'_e+d'_{\mathrm{AP}}}$ is a learnable vector to calculate the attention coefficient.

2) Edge Feature Update Module: In this module, we also use the calculated attention coefficients to aggregate the features of neighboring nodes and edges during edge feature updates. The difference is that each edge is connected both to an AP node and a UE node. Therefore, its neighboring edges are divided into two categories, one is the neighboring edge that is connected to the corresponding AP node, and the other is the neighboring edge that is connected to the corresponding UE node. We combine the aggregated information of these two types of neighbors to obtain our final update feature.

In the lth update layer, we incorporate the influence of neighboring edges on the target edge by aggregating information from other UE connections at the same AP node. Specifically, considering the target edge (b,k), let $\mathcal{N}_b$ denote the set of UE nodes that are connected to AP node b. Edge (b,m) is a neighboring edge of (b,k), where $m \in \mathcal{N}_b \setminus k$, and it contributes to the representation of edge (b,k) through an attention coefficient $\beta_{\mathrm{AP},(b,k),(b,m)}^{(l)}$ that can be computed as

$$\beta_{\mathrm{AP},(b,k),(b,m)}^{(l)} = \frac{\exp\left(\sigma\left(\mathbf{a}_e^T\left[\mathbf{W}_e \mathbf{e}_{b,m}^{(l-1)} || \mathbf{W}_{\mathrm{AP}} \mathbf{f}_{\mathrm{AP},b}^{(l-1)} || \mathbf{W}_e \mathbf{e}_{b,k}^{(l-1)}\right]\right)\right)}{\sum_{m'\in\mathcal{N}_b\setminus k} \exp\left(\sigma\left(\mathbf{a}_e^T\left[\mathbf{W}_e \mathbf{e}_{b,m'}^{(l-1)} || \mathbf{W}_{\mathrm{AP}} \mathbf{f}_{\mathrm{AP},b}^{(l-1)} || \mathbf{W}_e \mathbf{e}_{b,k}^{(l-1)}\right]\right)\right)}, \tag{23}$$

where $\mathbf{a}_e \in \mathbb{R}^{d'_{\mathrm{AP}}+2d'_e}$ is a learnable vector to calculate the attention coefficient. The softmax normalization ensures that the attention coefficients over all neighboring edges connected to node b (excluding edge (b,k) itself) sum to one.

To update the target edge (b,k), we also need to consider the influence of other edges connected to UE node k. Edge (n,k) is a neighboring edge of (b,k), where $n \in \mathcal{N}_k \setminus b$, and it contributes to the representation of (b,k) through an attention coefficient $\beta_{\mathrm{UE},(b,k),(n,k)}^{(l)}$ that can be computed as

$$\beta_{\mathrm{UE},(b,k),(n,k)}^{(l)} = \frac{\exp\left(\sigma\left(\mathbf{a}_e^T\left[\mathbf{W}_e \mathbf{e}_{n,k}^{(l-1)} || \mathbf{W}_{\mathrm{UE}} \mathbf{f}_{\mathrm{UE},k}^{(l-1)} || \mathbf{W}_e \mathbf{e}_{b,k}^{(l-1)}\right]\right)\right)}{\sum_{n'\in\mathcal{N}_k\setminus b} \exp\left(\sigma\left(\mathbf{a}_e^T\left[\mathbf{W}_e \mathbf{e}_{n',k}^{(l-1)} || \mathbf{W}_{\mathrm{UE}} \mathbf{f}_{\mathrm{UE},k}^{(l-1)} || \mathbf{W}_e \mathbf{e}_{b,k}^{(l-1)}\right]\right)\right)}. \tag{24}$$

Then we use these two types of attention coefficients to calculate the weighted sum of the transformation features of the two types of neighboring nodes and edges of the target edge, and the final aggregated information can be obtained as

$$\begin{aligned}\mathbf{e}_{b,k}^{(l)} = &\sum_{m\in\mathcal{N}_b\setminus k} \beta_{\mathrm{AP},(b,k),(b,m)}^{(l)} \left[\mathbf{W}_e \mathbf{e}_{b,m}^{(l-1)} || \mathbf{W}_{\mathrm{AP}} \mathbf{f}_{\mathrm{AP},b}^{(l-1)}\right] \\ &+ \sum_{n\in\mathcal{N}_k\setminus b} \beta_{\mathrm{UE},(b,k),(n,k)}^{(l)} \left[\mathbf{W}_e \mathbf{e}_{n,k}^{(l-1)} || \mathbf{W}_{\mathrm{UE}} \mathbf{f}_{\mathrm{UE},k}^{(l-1)}\right].\end{aligned} \tag{25}$$

Using the same operation as the node update module, the neighbor aggregation information of the target edge is obtained through a multi-head attention mechanism. Let $\mathbf{e}_{b,k,c}^{(l)}$ represent the result of (25) under the cth attention head. Then, the aggregation results under the C heads are concatenated,

and the concatenated high-dimensional features are linearly transformed through an MLP to obtain the final updated feature as follows

$$\mathbf{e}_{b,k}^{(l)} = \mathrm{MLP}_e\left(\|_{c=1}^{C}\mathbf{e}_{b,k,c}^{(l)}\right) + \mathbf{e}_{b,k}^{(l-1)}. \tag{26}$$

Here MLP_e denotes the MLP for updating the edge features.

D. Postprocessing Layer

In this section, we explain how to map the output of the neural network to the final required beamforming and association matrix.

Firstly, we input the final edge features $\mathbf{e}_{b,k}^{(L)}$ into an MLP and perform linear transformation to reduce their dimensionality to the desired level, i.e.,

$$\begin{aligned}[\mathbf{v}_{bk}^{\mathrm{Re}}, \mathbf{v}_{bk}^{\mathrm{Im}}] &= \mathrm{MLP}_{\mathrm{v}}(\mathbf{e}_{b,k}^{(L)}),\\ \pi_{bk} &= \mathrm{MLP}_x(\mathbf{e}_{b,k}^{(L)}),\end{aligned} \tag{27}$$

where $\mathrm{MLP}_{\mathrm{v}}$ is an MLP with a single linear layer and MLP_x is a single-layer MLP with a sigmoid activation function.

Then we need to consider the constraints of the optimization problem (3). Below, we process our output results for the three constraints of problem (3) and design the loss function to train our neural network.

1) Lagrangian Duality: To handle the constraints, we rewrite the original optimization problem (3) in terms of the EGAT network parameters $\boldsymbol{\theta}$ after neural network parameterization. With the beamforming vectors and association indicators now generated by the EGAT, the original formulation becomes

$$\max_{\boldsymbol{\theta}} \quad \sum_{k=1}^{K} w_k \log\left(1+\mathrm{SINR}_k(\boldsymbol{\theta})\right), \tag{28a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} x_{bk}(\boldsymbol{\theta}) \log\left(1+\mathrm{SINR}_k(\boldsymbol{\theta})\right) \le C_b^{\max}, \ \forall b \in \mathcal{B}, \tag{28b}$$

$$\|\mathbf{V}_b(\boldsymbol{\theta})\|_F^2 \le P_b^{\max}, \ \forall b \in \mathcal{B}, \tag{28c}$$

$$x_{bk}(\boldsymbol{\theta}) \in \{0,1\}, \ \forall b \in \mathcal{B}, k \in \mathcal{K}, \tag{28d}$$

To deal with the fronthaul constraint (28b), we introduce non-negative Lagrange multipliers $\boldsymbol{\mu} \in \mathbb{R}_+^B$, and form the partial Lagrangian as follows

$$\begin{aligned}\mathcal{L}_2(\boldsymbol{\theta}, \boldsymbol{\mu}) = &\sum_{k=1}^{K} w_k \log\left(1+\mathrm{SINR}_k(\boldsymbol{\theta})\right)\\ &- \sum_{b=1}^{B}\mu_b\left(\sum_{k=1}^{K} x_{bk}(\boldsymbol{\theta})\log\left(1+\mathrm{SINR}_k(\boldsymbol{\theta})\right) - C_b^{\max}\right).\end{aligned} \tag{29}$$

The corresponding dual problem can be formulated as a min-max optimization problem as follows

$$\min_{\boldsymbol{\mu}\ge 0}\max_{\boldsymbol{\theta}} \quad \mathcal{L}_2(\boldsymbol{\theta}, \boldsymbol{\mu}), \tag{30a}$$

$$\text{s.t.} \quad (28c), (28d). \tag{30b}$$

Such a problem can be solved by alternately optimizing $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ using primal–dual updates [39]. Following the analysis in [39], the primal–dual updates converge to a stationary point of the Lagrangian, and the use of near-universal GNN parameterizations ensures a small duality gap, yielding solutions that are close to the point satisfying the Karush-Kuhn-Tucker (KKT) conditions.

Note that the dual problem also needs to include the constraints (28c) and (28d). However, we can enforce these two constraints by choosing proper output functions in neural networks (e.g., sigmoid, softmax, and ReLU functions) or using proper processing (e.g., normalization operations, which we will discuss later). This eliminates the need to introduce additional dual parameters for the dual problem.

2) Association Matrix Design: One key challenge for the design of algorithms for the association matrix in (30) is how to manage the binary decision variables x_{bk}. Neural networks are not able to directly enforce binary constraints for the variables. To overcome this issue, we adopt a policy gradient-based approach [39] to indirectly generate binary outputs. Specifically, for each AP-UE pair (b,k), the network outputs a value $\pi_{bk} \in (0,1)$ according to (27) representing the probability that AP b serves UE k. These values are produced by applying a sigmoid activation function to the network output and collectively form an association probability matrix $\boldsymbol{\Pi} = \{\pi_{bk}\}_{b\in\mathcal{B},k\in\mathcal{K}}$. Each binary decision x_{bk} is then sampled from a Bernoulli distribution, i.e., $x_{bk} \sim \mathrm{Bernoulli}(\pi_{bk})$.

Let $\mathbf{X}$ denote the sampled binary association matrix. The joint probability of obtaining $\mathbf{X}$ under the association probability matrix $\boldsymbol{\Pi}$ is given by

$$P(\mathbf{X}|\boldsymbol{\Pi}) = \prod_{b=1}^{B}\prod_{k=1}^{K} \pi_{bk}^{x_{bk}}\left(1-\pi_{bk}\right)^{(1-x_{bk})}. \tag{31}$$

To evaluate the quality of each sampled matrix $\mathbf{X}$, we define a reward function $Q(\mathbf{X})$ that reflects the performance of the association scheme. We adopt the partial Lagrangian (29) as the reward, ensuring that both data rate and fronthaul feasibility are jointly considered. The reward function is given by

$$\begin{aligned}Q(\mathbf{X}) = &\left[\sum_{k=1}^{K} w_k \log\left(1+\mathrm{SINR}_k\right)\right.\\ &\left.- \sum_{b=1}^{B}\mu_b\left(\sum_{k=1}^{K} x_{bk}\log\left(1+\mathrm{SINR}_k\right) - C_b^{\max}\right)\right].\end{aligned} \tag{32}$$

Based on each sampled association matrix $\mathbf{X} \sim \boldsymbol{\Pi}$, we evaluate both the dual objective (29) and the policy gradient term using the REINFORCE principle. Specifically, the policy gradient term is given by the product $Q(\mathbf{X})\log P(\mathbf{X}|\boldsymbol{\Pi})$, serving as an unbiased estimator of the gradient of the expected reward. These two terms are combined to construct the loss function. To handle the stochastic nature of sampling, the final loss is computed by averaging the contributions from multiple independently sampled association matrices, i.e.,

$$Loss(c, \boldsymbol{\mu}) = -\mathbb{E}\left[\mathcal{L}_2(\boldsymbol{\theta}, \boldsymbol{\mu}) + Q(\mathbf{X})\log P(\mathbf{X}|\boldsymbol{\Pi})\right]. \tag{33}$$

Algorithm 2 EGAT With Primal-Dual Training

1: **Initialization**: Initialize EGAT parameters $\boldsymbol{\theta}^{\mathbf{V}}, \boldsymbol{\theta}^{\mathbf{\Pi}}$, dual variables $\boldsymbol{\mu} \geq \mathbf{0}$, learning rate and training dataset with N_{ta} samples. Set iteration index $t = 0$.
2: **repeat**
3: Set $t = t + 1$;
4: Perform forward pass of EGAT network to obtain beamforming matrix $\mathbf{V}$ and association probabilities $\mathbf{\Pi}$;
5: Sample binary association matrix $\mathbf{X}$ according to $\mathbf{\Pi}$;
6: Calculate loss function using (33);
7: Update primal and dual variables as follows:
Update $\boldsymbol{\theta}_t^{\mathbf{V}}$ via (38);
Update $\boldsymbol{\theta}_t^{\mathbf{\Pi}}$ via (34);
$\mu_t^b = \left[\mu_{t-1}^b - \left(\sum_{k=1}^{K} x_{bk} \log\left(1 + \mathrm{SINR}_k\right) - C_b^{\max}\right)\right]_+$, $\forall b \in \mathcal{B}$.
8: **until** maximum number of iteration is reached
9: **Output**: Optimized parameters $\boldsymbol{\theta}^{\mathbf{V}}, \boldsymbol{\theta}^{\mathbf{\Pi}}$, and dual variables $\boldsymbol{\mu}$.

Accordingly, the parameters $\boldsymbol{\theta}^{\mathbf{\Pi}}$ of the EGAT that outputs the association probability matrix $\mathbf{\Pi}$ are updated using the policy gradient method as follows

$$\boldsymbol{\theta}_{t+1}^{\mathbf{\Pi}} = \boldsymbol{\theta}_t^{\mathbf{\Pi}} + \eta_{\mathbf{\Pi}} \mathbb{E}\left[Q(\mathbf{X}) \nabla_{\boldsymbol{\theta}^{\mathbf{\Pi}}} \log P(\mathbf{X}|\mathbf{\Pi})\right], \tag{34}$$

where $\eta_{\mathbf{\Pi}}$ denotes the learning rate.

3) Beamforming Design: Following the conversion of the postprocessing layer in (27), we splice the obtained real and imaginary parts of beamforming vectors and restore them to complex numbers as follows

$$\mathbf{v}_{bk} = \mathbf{v}_{bk}^{\mathrm{Re}} + j\mathbf{v}_{bk}^{\mathrm{Im}}. \tag{35}$$

Then, we multiply the association matrix with the beamforming matrix. This step aims to remove the irrelevant elements in the beamforming matrix. It also serves as a prerequisite for power normalization in the subsequent steps. The formula for this step is

$$\mathbf{v}_{bk} = \mathbf{v}_{bk} x_{bk}, \forall b \in \mathcal{B}, k \in \mathcal{K}. \tag{36}$$

Afterwards, we perform power normalization on the beamforming vector to meet the power constraint as follows

$$\mathbf{V}_b = \frac{\mathbf{V}_b}{\|\mathbf{V}_b\|_F} \sqrt{P_b^{\max}}, \forall b \in \mathcal{B}. \tag{37}$$

The beamforming matrix $\mathbf{V}$ is generated by the EGAT network with parameters $\boldsymbol{\theta}^{\mathbf{V}}$. Since elements of $\mathbf{V}$ are continuous-valued outputs, $\boldsymbol{\theta}^{\mathbf{V}}$ are optimized via standard gradient descent on the overall loss function as follows

$$\boldsymbol{\theta}_t^{\mathbf{V}} = \boldsymbol{\theta}_{t-1}^{\mathbf{V}} + \eta_{\mathbf{V}} \nabla_{\boldsymbol{\theta}^{\mathbf{V}}} \mathrm{Loss}(\boldsymbol{\theta}, \boldsymbol{\mu}), \tag{38}$$

where $\eta_{\mathbf{V}}$ denotes the learning rate.

To conclude, we summarize the learning-based EGAT approach with primal-dual training in Algorithm 2.

TABLE I
SIMULATION PARAMETERS

Symbols	Physical Meaning	Values
M	The number of BS antennas	2
K	The number of UEs	4
B	The number of the APs	8
w_k	The weight for UEs data rate	1
$P_b^{\max}$	Max transmit power of AP b	30 dBm
ξ_k^2	UE noise power	-85 dBm
BW	Bandwidth	10MHz

V. PERFORMANCE EVALUATION

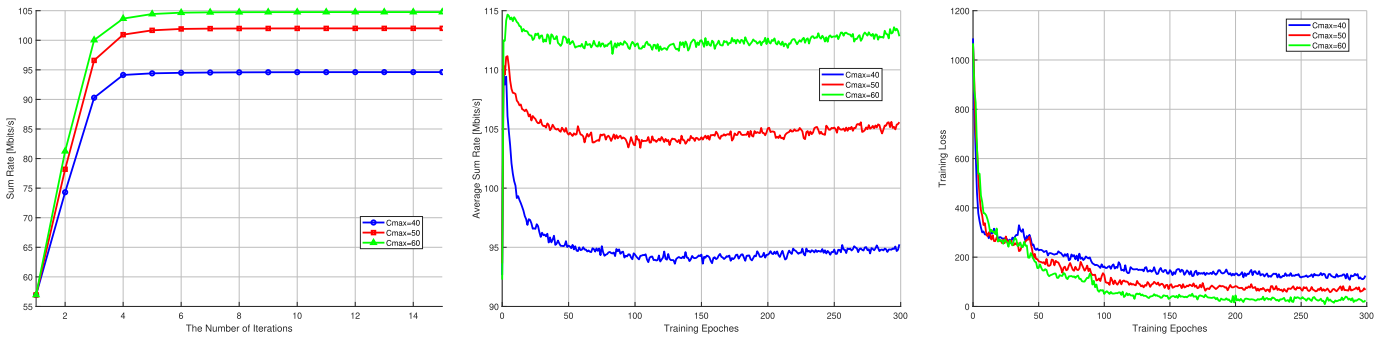
In this section, we present numerical results to demonstrate the effectiveness of the proposed approaches for cell-free communication systems. We compare the overall performance of the proposed algorithms with state-of-the-art model-based and learning-based algorithms. We also conduct detailed tests on various performance indicators for the proposed methods, including generalization performance with respect to the number of APs and UEs, and performance under different fronthaul capacity constraints.

A. Simulation Settings

Our simulation parameters are summarized in Table I. The network structure and channel models are adopted from [7]. We consider a downlink wireless network in a 500m $\times$ 500m area, where the APs and UEs are uniformly distributed. The wrap-around technique is employed to eliminate boundary effects. The AP antenna height is 15 m; the user antenna height is 1.65 m; and the carrier frequency is set as 1.9 GHz. The channel between AP b and user k is expressed as $\mathbf{h}_{bk} = \sqrt{\rho_{bk}}\mathbf{g}_{bk}$, where $\mathbf{g}_{bk}$ denotes Rayleigh small-scale fading effects. The large-scale fading coefficient $\rho_{bk} = PL_{bk} \cdot 10^{\frac{\sigma_{\mathrm{sh}} z_{bk}}{10}}$ integrates path loss PL_{bk} and log-normal shadowing. The path loss PL_{bk} follows the COST231-Hata model with the three-slope distance correction. The standard deviation of shadow fading is $\sigma_{\mathrm{sh}} = 8$ dB. The shadowing correlation of z_{bk} is modeled by an exponential decay function with 0.5 km correlation distance, applied separately across APs and users.

For the proposed EGAT structure, all the MLPs in (19), (22) and (26) are implemented by 3 linear layers, followed by a ReLU activation function and a batch normalization layer. In the training process, the number of epochs is set to 300. Each epoch consists of 100 mini-batches of training samples with a batch size of 256. For each training sample, the APs' and UEs' locations, and the channel coefficients are randomly generated. A learning rate $\eta = 10^{-3}$ is adopted to update the trainable parameters of neural networks using the Adam solver in an unsupervised manner. After training, we test the average performance of 200 samples. For comparison, the following schemes are considered:

- *AO-SCA Scheme*: Our proposed model-based approach in Section III.
- *EGAT Scheme*: Our proposed learning-based approach in Section IV.



(a) Average sum rate of AO-SCA (b) Average sum rate of EGAT (c) Training loss of EGAT

Fig. 4. Convergence behavior of the proposed algorithms.

- *ENGNN Scheme* [33]: ENGNN employs an edge-update mechanism to handle both node and edge variables without attention. Fronthaul constraints and AP-UE associations are addressed using the method in Section IV-D.
- *BGNN Scheme* [32]: BGNN interprets the cell-free system as a weighted bipartite graph where APs and UEs form disjoint vertex sets connected via edges weighted by channel coefficients. It employs a distinct message-passing and update mechanism.
- *DNN Scheme* [40]: A fully-connected dense neural network (DNN) model directly learns the mapping from channel coefficients to the beamforming vectors and AP-UE association matrices, which does not utilize the topology of the network.
- *Full Connection Approach Using Zero-Forcing (FC-ZF) Scheme*: Each UE is served by all APs, with beamforming vectors computed via the ZF algorithm [15].
- *Partial Connection Using Weighted Minimum Mean-Square Error (PC-WMMSE) Scheme*: All UEs are initially connected to all APs. Beamforming vectors are computed via WMMSE. AP fronthaul loads are checked iteratively, disconnecting the UE with the weakest channel from the overloaded AP until all fronthaul constraints are satisfied.

To ensure a fair comparison under stringent fronthaul constraints, we employ a *heuristic scaling approach* to ensure the fronthaul constraints are satisfied by the learning-based algorithms,i.e., EGAT, ENGNN, BGNN, and DNN. This approach iteratively scales down the beamforming gain of the weakest UE associated with each overloaded AP until all fronthaul constraints are met. We verified that this adjustment has negligible impact on the final performance of our methods, where the average reduction in the weighted sum rate remains below 3%.

B. Convergence Behavior

Fig. 4 demonstrates the convergence performance of the proposed AO-SCA and EGAT algorithms. Fig. 4a illustrates the convergence behavior of the AO-SCA algorithm under different fronthaul capacity constraints $C^{\max}$. The weighted sum rate increases monotonically and converges within 8–10 iterations, with larger $C^{\max}$ leading to higher final performance. This confirms the effectiveness of the alternating optimization with SCA.

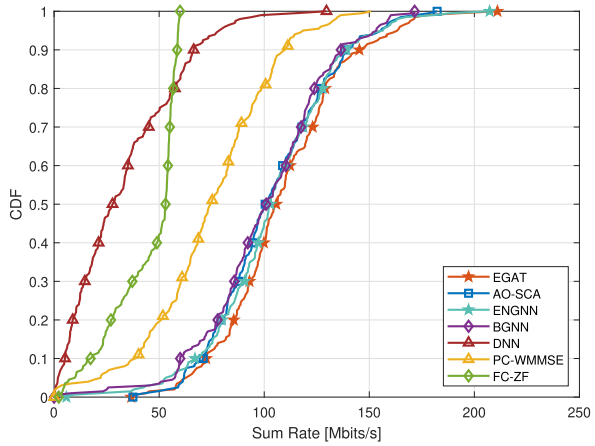
The training dynamics of EGAT, shown in Figs. 4b- 4c, are more complex. Fig. 4b shows that the average weighted sum rate initially reaches a high value and then gradually decreases as the Lagrangian multipliers enforce the fronthaul constraints to be satisfied. It eventually stabilizes after 250 epochs. Fig. 4c presents the training loss, which exhibits a step-like pattern due to learning rate adjustments and the evolution of Lagrangian multipliers. Similar to Fig. 4b, the training loss stabilizes after 250 epochs, demonstrating the effectiveness of the primal-dual training framework of EGAT.

C. Model Effectiveness

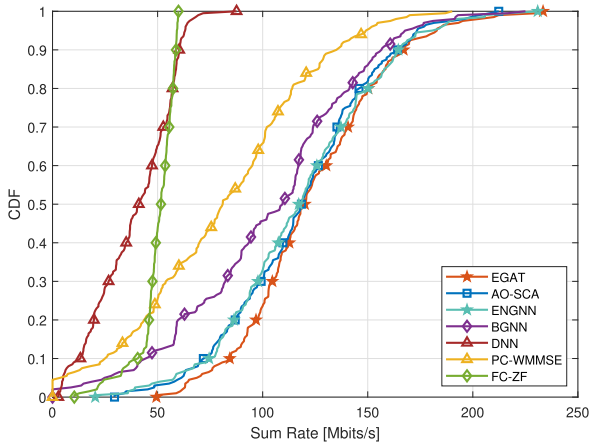
Fig. 5a and Fig. 5b show the empirical cumulative distribution function (CDF) of weighted sum rates for all algorithms under different number of APs. The proposed EGAT algorithm consistently outperforms all baselines. Its advantage over ENGNN is attributed to the attention-based aggregation mechanism, which effectively captures the latent interference patterns. Although BGNN adopts a graph-structured design, its limited edge-feature updating capability leads to inferior performance compared with both EGAT and ENGNN, with the performance gap increasing evidently as the number of APs increases. The DNN baseline performs the worst among the learning-based algorithms because it fails to exploit the underlying graph topology. Beyond the learning-based algorithms, EGAT also surpasses the model-based AO-SCA, as its policy-gradient association module yields higher-quality solutions and mitigates premature convergence. Compared to AO-SCA and EGAT, PC-WMMSE performs worse due to its inferior UE association design. FC-ZF performs the worst due to its sensitivity to noise and reliance on power scaling.

D. Performance Under Fronthaul Constraints

Fig. 6 presents the performance comparison of all the algorithms under varying fronthaul capacity constraints. As observed from the figure, the proposed EGAT algorithm outperforms the other algorithms across all fronthaul capacity constraints. ENGNN and AO-SCA achieve similar weighted sum rates in our simulation settings. DNN exhibits the poorest performance across all capacity settings, which can be attributed to its failure to exploit the topological information of wireless networks.



(a) Number of APs = 8.



(b) Number of APs = 12.

Fig. 5. The empirical CDF of sum rate of all schemes.

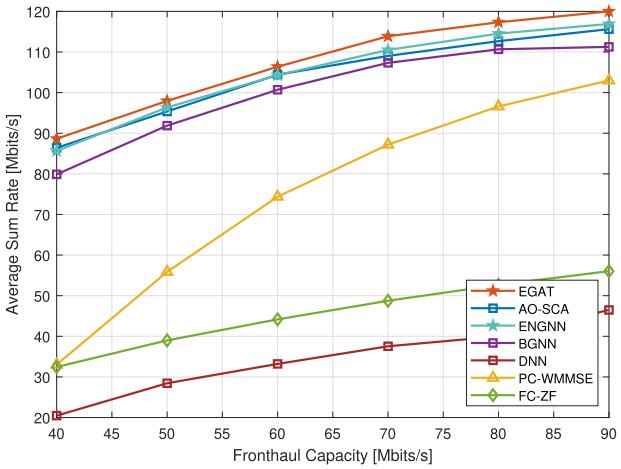


Fig. 6. Comparative analysis of algorithm performance under different fronthaul capacities.

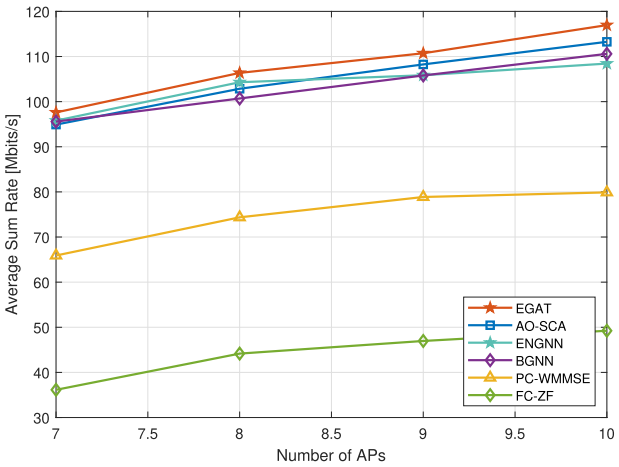


Fig. 7. Comparative analysis of generalization performance for different number of APs.

TABLE II

FEASIBILITY PROBABILITY UNDER DIFFERENT FRONTHAUL CAPACITIES (W/O HEURISTIC SCALING TO ENSURE FRONTHAUL FEASIBILITY)

Alg. \ Cap.	40 (Mbits/s)	50 (Mbits/s)	60 (Mbits/s)	70 (Mbits/s)	80 (Mbits/s)	90 (Mbits/s)
EGAT	**0.912**	**0.946**	**0.949**	0.965	0.983	0.991
ENGNN	0.901	0.928	0.945	**0.981**	**0.990**	**0.992**
BGNN	0.876	0.892	0.908	0.947	0.966	0.967

Table II summarizes the feasibility performance of all GNN-based schemes under different fronthaul capacity constraints without the heuristic scaling approach in Section V-A. Although primal-dual training cannot guarantee the fronthaul constraints to be strictly satisfied by the learning-based approaches, we still observe high feasibility probabilities especially for EGAT and ENGNN, which are both above 90% under all capacity limits. The feasibility of EGAT is higher at low to medium fronthaul capacity limits, while ENGNN has higher feasibility at high capacity limits.

E. Generalization Performance Comparison

Neural networks are trained under fixed AP and UE configurations. However, practical deployments often involve different network sizes, making generalization ability a key metric for learning-based approaches. Models with stronger generalization ability ensure better adaptability in real scenarios. Here we train our model using a base scenario with 4 UEs and 8 APs and verify the network's generalization performance across different numbers of UEs and APs.

Fig. 7 presents the generalization performance of the evaluated schemes under varying numbers of APs. The results indicate that the EGAT model consistently achieves the best overall performance across the considered settings. Although its advantage over AO-SCA narrows as the test network diverges from the training configuration, EGAT remains the most robust among learning-based approaches. ENGNN exhibits limited scalability, with modest rate improvement as AP count increases and noticeable degradation under configuration shifts. Although BGNN has the lowest average sum rate in the 8-AP scenario, it surpasses ENGNN when the AP number increases to 10.

Table III provides the feasibility probability of each scheme under varying AP numbers without the heuristic scaling approach in Section V-A. It can be observed that EGAT maintains the highest feasibility probability consistently for all the considered number of APs.

TABLE III

FEASIBILITY PROBABILITY WITH DIFFERENT NUMBER OF APS (W/O HEURISTIC SCALING TO ENSURE FRONTHAUL FEASIBILITY)

Alg. \ AP	7	8	9	10
EGAT	**0.946**	**0.949**	**0.952**	**0.942**
ENGNN	0.944	0.945	0.943	0.942
BGNN	0.917	0.908	0.931	0.927

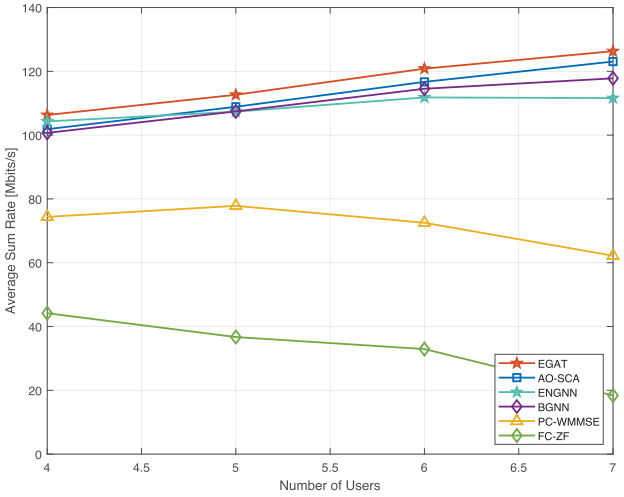


Fig. 8. Comparative analysis of generalization performance for the number of UEs.

TABLE IV

FEASIBILITY PROBABILITY WITH DIFFERENT NUMBER OF UES (W/O HEURISTIC SCALING TO ENSURE FRONTHAUL FEASIBILITY)

Alg. \ UE	4	5	6	7
EGAT	**0.949**	0.959	0.958	0.967
ENGNN	0.945	**0.962**	**0.967**	**0.974**
BGNN	0.908	0.943	0.948	0.946

Fig. 8 shows the generalization performance of all schemes with respect to the number of UEs. Consistent with Fig. 7, EGAT achieves the highest average rate, which shows it can effectively handle the increased interference and resource competition. In contrast, ENGNN experiences severe performance degradation when the number of UEs increase to 7.

Table IV provides the feasibility probability for different number of UEs without the heuristic scaling approach in Section V-A. While ENGNN's feasibility probability exceeds that of EGAT when the number of UEs increases, this is achieved at the cost of sacrificing average sum rate performance. All the considered GNN-based learning approaches achieve high feasibility probability, which are all above 90%.

Fig. 9 shows the average number of serving APs per UE under different UE densities. The proposed EGAT yields the most stable and well-structured association pattern, maintaining approximately two serving APs per UE. As the number of UEs increases, its number of connected APs gradually decreases. This phenomenon is due to the fronthaul capacity limit. The average APs connected per UE for AO–SCA is similar to that of EGAT, while it attains a lower average sum-rate than EGAT. Such behavior shows that EGAT is able to infer near-optimal AP–UE associations under varying user densities. ENGNN tends to over-allocate APs compared to EGAT without increasing the average sum-rate. Moreover, BGNN fails to allocate enough APs to UEs, resulting in less than one serving AP per UE on average and thus leaves many UEs being unable to be served. PC-WMMSE and FC-ZF result in the densest connections due to the lack of effective association strategies.

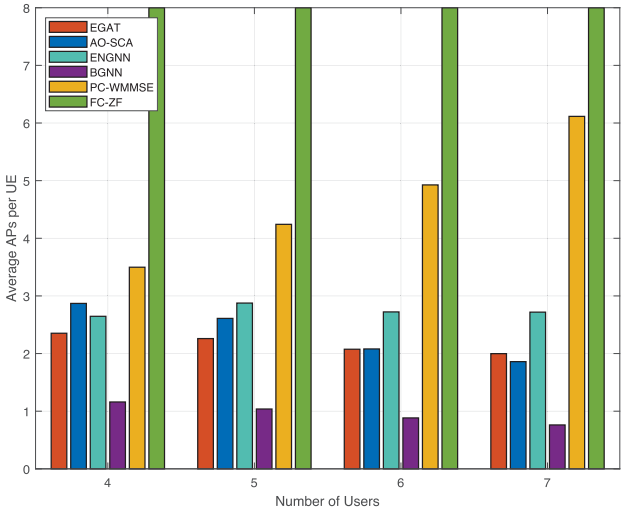


Fig. 9. Comparative analysis of the average number of APs connected to each UE for the number of UEs.

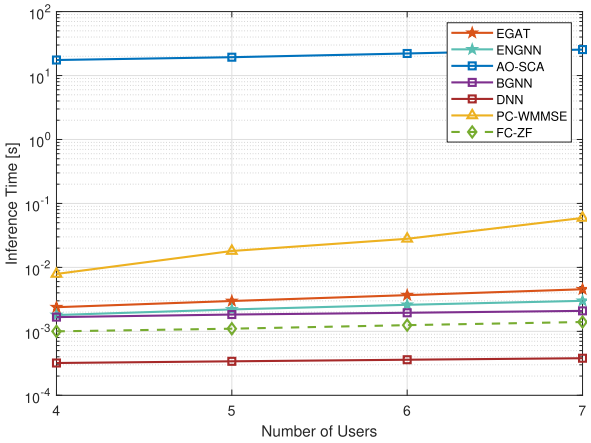


Fig. 10. The average inference time required for different schemes.

F. Complexity Comparisons

The computational complexity of EGAT primarily arises from the update layers. In the forward pass, updating node features requires aggregating information from neighbors, giving a cost of $O(BK)$. The edge-feature updates are more expensive because each edge (b,k) aggregates information from the other $(K-1)$ edges incident to the same AP and the other $(B-1)$ edges incident to the same UE, so the per-edge cost is $O(B+K)$. Since there are BK edges in total, the overall complexity of the edge updates is $O(BK(B+K))$. Consequently, with L update layers, the total cost of the EGAT is $O(LBK(B+K))$, where the feature dimensions and the number of attention heads are treated as constants.

Fig. 10 compares the inference time of different schemes as the number of UEs varies. The experiments are performed

TABLE V
REQUIRED FLOPS FOR ENGNN AND EGAT

	Alg.	ENGNN	EGAT
$K=3$	$B=4$	2.5×10^6	2.7×10^6
	$B=8$	4.6×10^6	5.1×10^6
	$B=10$	5.7×10^6	6.4×10^6
$K=6$	$B=4$	4.5×10^6	5.0×10^6
	$B=8$	8.5×10^6	9.7×10^6
	$B=10$	1.1×10^7	1.2×10^7

on an Intel Core i5-10210U CPU with 16 GB of memory. Training the EGAT for 300 epochs on a dataset of $25,600$ samples requires approximately 10–12 hours. This training cost is incurred only offline and does not impact the latency of online inference. The inference of GNN-based algorithms runs significantly faster than iterative PC-WMMSE and AO-SCA, with AO-SCA exhibiting the longest runtime due to its reliance on convex optimization solvers. In contrast, GNNs maintain stable and efficient inference, highlighting their practical deployment advantage.

A comparison of the required floating-point operations (FLOPs) between EGAT and ENGNN is presented in Table V. EGAT requires more FLOPs due to its attention mechanism, which dynamically computes coefficients for neighboring nodes using additional matrix multiplications, nonlinear activations, and normalization. As shown in Fig. 10, EGAT's inference time is roughly 1.5 times that of ENGNN, which is caused by frequent access to node features and re-computation of attention weights in each forward pass. In contrast, ENGNN uses simpler linear message passing with fixed weights, which reduces computational and memory overhead. Despite this overhead, EGAT's absolute inference time remains low, and its performance gain in weighted sum rate justifies this additional cost.

VI. CONCLUSION

In this paper, we investigate the joint optimization of AP-UE association strategies and downlink beamforming design in a cell-free network, subject to constraints on limited fronthaul capacity and transmit power. We first propose a model-based optimization algorithm named AO-SCA. To improve algorithm performance in practical deployment, we propose a learning-based algorithm named EGAT, which solves the considered problem more efficiently. We employ a primal-dual learning framework to tackle the complicated fronthaul constraints in neural network, and use a policy gradient scheme to solve the binary constraints for AP-UE association. Our simulation results show that the proposed EGAT algorithm achieves superior performance and inference speed balance than state-of-the-art benchmark algorithms. Moreover, the proposed EGAT algorithm shows stronger generalization capability across network sizes than conventional GNN-based algorithms.

REFERENCES

[1] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6G networks: Use cases and technologies," *IEEE Commun. Mag.*, vol. 58, no. 3, pp. 55–61, Mar. 2020.

[2] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6G wireless communication systems: Applications, requirements, technologies, challenges, and research directions," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 957–975, 2020.

[3] R. Madan, J. Borran, A. Sampath, N. Bhushan, A. Khandekar, and T. Ji, "Cell association and interference coordination in heterogeneous LTE–A cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 9, pp. 1479–1489, Dec. 2010.

[4] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.

[5] W. Yu, T. Kwon, and C. Shin, "Multicell coordination via joint scheduling, beamforming, and power spectrum adaptation," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 1–14, Jul. 2013.

[6] Y.-F. Liu and Y.-H. Dai, "On the complexity of joint subcarrier and power allocation for multi-user OFDMA systems," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 583–596, Feb. 2014.

[7] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[8] J. Zhang, E. Björnson, M. Matthaiou, D. W. K. Ng, H. Yang, and D. J. Love, "Prospective multiple antenna technologies for beyond 5G," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1637–1660, Aug. 2020.

[9] E. Björnson and L. Sanguinetti, "Making cell-free massive MIMO competitive with MMSE processing and centralized implementation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 77–90, Jan. 2020.

[10] S. Buzzi and C. D'Andrea, "Cell-free massive MIMO: User-centric approach," *IEEE Wireless Commun. Lett.*, vol. 6, no. 6, pp. 706–709, Dec. 2017.

[11] Ö. Özdogan, E. Björnson, and J. Zhang, "Performance of cell-free massive MIMO with Rician fading and phase shifts," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5299–5315, Nov. 2019.

[12] M. Bashar, K. Cumanan, A. G. Burr, M. Debbah, and H. Q. Ngo, "On the uplink max–min SINR of cell-free massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2021–2036, Apr. 2019.

[13] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, "Cell-free massive MIMO for wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6377–6392, Oct. 2020.

[14] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4247–4261, Jul. 2020.

[15] E. Nayebi, A. Ashikhmin, T. L. Marzetta, H. Yang, and B. D. Rao, "Precoding and power optimization in cell-free massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4445–4459, Jul. 2017.

[16] H. Q. Ngo, L.-N. Tran, T. Q. Duong, M. Matthaiou, and E. G. Larsson, "On the total energy efficiency of cell-free massive MIMO," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 1, pp. 25–39, Mar. 2018.

[17] R. Pinto Antonioli, I. M. Braga, G. Fodor, Y. C. B. Silva, A. L. F. de Almeida, and W. C. Freitas, "On the energy efficiency of cell-free systems with limited fronthauls: Is coherent transmission always the best alternative?," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8729–8743, Oct. 2022.

[18] J. Zhao, T. Q. S. Quek, and Z. Lei, "Heterogeneous cellular networks using wireless backhaul: Fast admission control and large system analysis," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2128–2143, Oct. 2015.

[19] J. Yao, J. Xu, W. Xu, D. W. K. Ng, C. Yuen, and X. You, "Robust beamforming design for RIS-aided cell-free systems with CSI uncertainties and capacity-limited backhaul," *IEEE Trans. Commun.*, vol. 71, no. 8, pp. 4636–4649, Aug. 2023.

[20] S. Geng, Z. Wei, J. Zhao, F. Shen, J. Joung, and S. Sun, "Joint deployment and resource allocation for service provision in multi-UAV-assisted wireless networks," *IEEE Internet Things J.*, vol. 11, no. 22, pp. 37269–37286, Nov. 2024.

[21] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 101–115, Jan. 2021.

[22] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 2977–2991, 2020.

[23] M. Lee, G. Yu, and G. Y. Li, "Graph embedding-based wireless link scheduling with few training samples," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2282–2294, Apr. 2021.

[24] A. Chowdhury, G. Verma, C. Rao, A. Swami, and S. Segarra, "Unfolding WMMSE using graph neural networks for efficient power allocation," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 6004–6017, Sep. 2021.
[25] I. Nikoloska and O. Simeone, "Modular meta-learning for power control via random edge graph neural networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 457–470, Jan. 2023.
[26] S. He, S. Xiong, W. Zhang, Y. Yang, J. Ren, and Y. Huang, "GBLinks: GNN-based beam selection and link activation for ultra-dense D2D mmWave networks," *IEEE Trans. Commun.*, vol. 70, no. 5, pp. 3451–3466, May 2022.
[27] Y. Li, Y. Lu, B. Ai, O. A. Dobre, Z. Ding, and D. Niyato, "GNN-based beamforming for sum-rate maximization in MU-MISO networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 9251–9264, Aug. 2024.
[28] Y. Zhang, J. Yang, Q. Liu, Y. Liu, and T. Zhang, "Unsupervised learning-based coordinated hybrid precoding for mmWave massive MIMO-enabled HetNets," *IEEE Trans. Wireless Commun.*, vol. 23, no. 7, pp. 7200–7213, Jul. 2024.
[29] S. Mishra, L. Salaun, H. Yang, and C. S. Chen, "Graph neural network aided power control in partially connected cell-free massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 23, no. 9, pp. 12412–12423, Sep. 2024.
[30] J. Guo and C. Yang, "Learning power allocation for multi-cell-multi-user systems with heterogeneous graph neural networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 884–897, Feb. 2022.
[31] X. Zhang, H. Zhao, J. Xiong, X. Liu, L. Zhou, and J. Wei, "Scalable power control/beamforming in heterogeneous wireless networks with graph neural networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2021, pp. 01–06.
[32] J. Kim, H. Lee, S.-E. Hong, and S.-H. Park, "A bipartite graph neural network approach for scalable beamforming optimization," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 333–347, Jan. 2023.
[33] Y. Wang, Y. Li, Q. Shi, and Y.-C. Wu, "ENGNN: A general edge-update empowered GNN architecture for radio resource management in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 6, pp. 5330–5344, Jun. 2024.
[34] N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "Learning resilient radio resource management policies with graph neural networks," *IEEE Trans. Signal Process.*, vol. 71, pp. 995–1009, 2023.
[35] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. V. Srinivas, "User-centric cell-free massive MIMO networks: A survey of opportunities, challenges and solutions," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 611–652, 1st Quart., 2022.
[36] K. Shen and W. Yu, "Fractional programming for communication systems—Part I: Power control and beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, May 2018.
[37] M. Grant and S. Boyd. (2020). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.2*. [Online]. Available: https://cvxr.com/cvx
[38] J. Zhao, T. Q. S. Quek, and Z. Lei, "Coordinated multipoint transmission with limited backhaul data transfer," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2762–2775, Jun. 2013.
[39] M. Eisen, C. Zhang, L. F. O. Chamon, D. D. Lee, and A. Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2775–2790, May 2019.
[40] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.

Xiaoyu Liu received the B.S. degree from the School of Information Science and Engineering, Shandong University, China, in 2020. She is currently pursuing the M.S. degree with the School of Electronic Science and Engineering, Nanjing University, China. Her research interests include machine learning for wireless communications and 6G communication networks.

Jian Zhao (Senior Member, IEEE) received the B.S. degree from Nanjing University, Nanjing, China, and the Dr. Sc. degree from Swiss Federal Institute of Technology (ETH) Zürich, Switzerland. From 2010 to 2015, he was a Research Scientist with the Institute for Infocomm Research, A*STAR, Singapore. He is currently an Associate Professor with the School of Electronic Science and Engineering, Nanjing University. He has published over 70 papers in leading journals and conferences, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. His research interests include machine learning, wireless communications and networks, and mathematical optimization techniques. He was honored with the Best Paper Awards from IEEE GLOBECOM and ICCC, and Chinese Government Award for Outstanding Self-Financed Students Abroad. He was a co-recipient of Jiangsu Computer Society Science and Technology Award in 2022.

Furao Shen (Member, IEEE) received the B.Sc. and M.Sc. degrees in mathematics from Nanjing University, Nanjing, China, in 1995 and 1998, respectively, and the Ph.D. degree from Tokyo Institute of Technology, Tokyo, Japan, in 2006. He is currently a Full Professor with the School of Artificial Intelligence, Nanjing University. His research interests include neural computing and robotic intelligence.

Kun Yang (Fellow, IEEE) received the Ph.D. degree from the Department of Electronic and Electrical Engineering, University College London (UCL), U.K. He is currently a Chair Professor with Nanjing University and an affiliated Professor with the University of Essex. He has published more than 600 articles and filed 50 patents. His main research interests include wireless networks and communications, communication-computing cooperation, and AI (artificial intelligence) for wireless. He is a member of Academia Europaea (MAE), a fellow of IET, and a Distinguished Member of ACM. He has been a Judge of GSMA GLOMO Award at World Mobile Congress–Barcelona since 2019. He was a Distinguished Lecturer of IEEE ComSoc, a Recipient of the 2024 IET Achievement Medals, and a Recipient of the 2024 IEEE CommSoft TC's Technical Achievement Award. He serves on the Editorial Board of a few IEEE journals (e.g., IEEE WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE TRANSACTIONS ON NANOBIOSCIENCE). He is a Deputy Editor-in-Chief of *IET Smart Cities* journal.

Sumei Sun (Fellow, IEEE) is currently the Executive Director of the Institute for Infocomm Research (I^2R), Agency for Science, Technology, and Research (A*STAR), Singapore. She also holds a joint appointment with Singapore Institute of Technology, and an adjunct appointment with the National University of Singapore, both as a Full Professor. Her current research interests include next-generation wireless communications, joint communication-sensing-computing-control design, the industrial Internet of Things, and applied artificial intelligence. She is a member of the IEEE Vehicular Technology Society Board of Governors (2022–2027) and a fellow of the Academy of Engineering Singapore.