Heterogeneous Graph Neural Network Methods for Multi-User Fluid Antenna Systems

Geon Kim, Jae Seo Lee, and Hoon Lee, *Member, IEEE*

***Abstract*—This paper proposes a graph neural network (GNN) method for a multi-user fluid antenna system (FAS) where a base station equipped with multiple fluid antennas (FAs) serves multiple users in the downlink. In the FAS, it is necessary to jointly optimize the beamforming vectors and the positions of FAs. To address this nonconvex task, we present heterogeneous GNN (HetGNN) architectures that model the multi-user FAS as heterogeneous graphs comprising user nodes and antenna nodes. To achieve high-level generalization across arbitrary system sizes, the proposed HetGNN is carefully designed to be invariant to the number of FAs and users. Consequently, a sole HetGNN can be readily applied to any FAS with diverse configurations. Numerical results demonstrate the effectiveness of the proposed method over conventional approaches.**

I. Introduction

Over the past few decades, advances in multiple-input multiple-output (MIMO) technologies have substantially increased the capacity of wireless networks [1]–[3]. Nevertheless, with fixed antenna configurations, the wireless propagation environment remains highly uncontrollable and stochastic, leading to deep fading. To overcome this challenge, research on fluid antennas (FAs) [4], [5] employs liquid-metal conductors to realize flexible MIMO configurations. It has been shown that dynamic controls of FA locations can overcome the deep fading issue [6].

The full potential of fluid antenna systems (FASs) can be achieved through a joint optimization of beamforming and FA positions [7]–[9]. A multi-user multiple-input single-output (MISO) FAS was considered in [7] where a base station (BS) equipped with multiple FAs serves multiple single antenna users. The transmit power minimization problem under the minimum signal-to-interference-plus-noise ratio (SINR) constraints was solved using a block coordinate descent (BCD) algorithm. The weighted sum rate maximization task was investigated in the multi-user MISO FAS [8]. For fixed FA locations, efficient beamforming vectors were identified using the weighted minimum mean-squared-error (WMMSE) algorithm [10], whereas the position optimization for given beamforming was addressed via the successive convex approximation method. These approaches have been extended to multi-user MIMO FASs [9] where both the BS and users employ multiple FAs. Similar to [7], [8], the SCA-based BCD optimization technique was developed to determine precoding matrices and transmit-receive FA locations.

Due to the coupled nature of FA locations and beamforming vectors, the optimization techniques in multi-user FASs naturally invoke iterative and alternating algorithms with high computational complexity. This becomes severe when handling large-scale FASs with a number of FAs and users. Therefore, existing optimization-based approaches lack the scalability to the network size.

This limitation can be addressed through the learning-to-optimize framework which exploits neural networks to determine efficient signal processing policies [2], [3], [11]–[14]. Among various candidate architectures, graph neural networks (GNNs) have been regarded as promising models to learn optimal transmission strategies in multi-user networks [3], [11]–[14]. Thanks to the permutation equivariant property, the GNN possesses the generalization capability to the network size, e.g., the number of users. Such a scalability has been investigated for wireless resource management problems in single-antenna systems [11], [12]. They characterized transmitter-receiver pairs as nodes in a graph with their edge connections corresponding to wireless links. This leads to homogeneous GNN (HomGNNs) that models wireless networks by graphs consisting of single-type nodes.

The HomGNN architecture has been recently applied to multi-user MISO FASs for optimizing the FA position and beamforming jointly [14]. Similar to [12], this work models BS-user pairs as single-type nodes in homogeneous graphs. Although it can be universally applicable to FAS with arbitrary user populations, it lacks the generalization capability to the number of FAs. We can overcome such challenges using heterogeneous GNNs (HetGNNs) [3], [13], which define antennas at BSs and users as distinct types of nodes in heterogeneous graphs. By doing so, we can achieve the scalability to both the number of antennas and users. There have been several studies on HetGNN-based beamforming optimization for conventional fixed antenna systems [3], [13]. However, these cannot be straightforwardly extended to FASs with dynamic antenna configurations.

This paper presents a HetGNN approach to multi-user MISO FASs where a FAS-aided BS broadcasts data symbols to multiple users. To maximize the sum rate performance, it is essential to optimize FA positions at the BS and multi-user beamforming vectors jointly. Although this problem can be addressed using existing BCD algorithms, e.g., the WMMSE-based approach [8], their computational complexity become prohibitive as the system size gets larger. To this end, we propose the HetGNN method which can be universally applied to arbitrary FASs with various populations of FAs and users. Inspired by [14], the proposed learning architecture consists of two subsequent HetGNNs, each of which identifies efficient FA locations and beamforming vectors. Numerical results demonstrate the effectiveness of the proposed approach.

The authors are with the Department of Electrical Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan, 44919, South Korea (e-mail: {geoniekim,leejss,hoonlee}@unist.ac.kr).

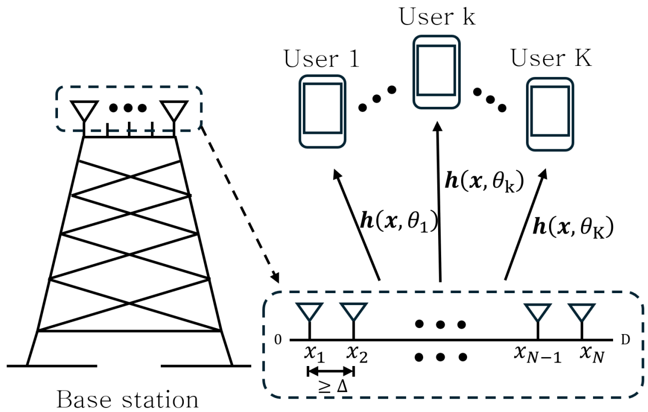

Fig. 1. Schematic diagram of multi-user FAS.

II. System Model

This section describes a problem for optimizing antenna placement and beamforming vectors in multi-user FAS, which is followed by the background on HetGNN.

A. Problem description

As shown in Fig. 1, we consider a multi-user FAS where a BS equipped with N FAs provides downlink communication services to K single-antenna users. Let $\mathcal{N} \triangleq \{1,\cdots,N\}$ and $\mathcal{K} \triangleq \{1,\cdots,K\}$ be index sets of the FAs and users, respectively. It is assumed that the FAs at the BS are arranged in a linear array of length D. The location of FA $n \in \mathcal{N}$ is denoted by $x_n \in [0, D]$. Without loss of generality, we assume that $x_n < x_m$ for $n < m$, $\forall n, m \in \mathcal{N}$. Also, FAs are subject to the minimum antenna spacing constraint Δ expressed by

$$x_n - x_{n-1} \geq \Delta, \forall n \in \mathcal{N}, \tag{1}$$

where we define $x_0 \triangleq -\Delta$.

Let $\theta_k \in [0, \pi]$ be the angle between the BS and user $k \in \mathcal{K}$. Then, the channel vector from the BS to user k can be expressed as [14]

$$\mathbf{h}(\mathbf{x}, \theta_k) = \left[e^{j\frac{2\pi}{\lambda}x_1\cos(\theta_k)}, \cdots, e^{j\frac{2\pi}{\lambda}x_N\cos(\theta_k)}\right]^T, \tag{2}$$

where $\mathbf{x} \triangleq [x_1, \cdots, x_N]^T \in \mathbb{R}^N$ stands for the FA location vector and λ represents the wavelength. Denoting the beamforming vector for user k as $\mathbf{w}_k \in \mathbb{C}^N$, the received signal y_k at user k can be written by

$$y_k = \mathbf{h}(\mathbf{x}, \theta_k)^H \mathbf{w}_k s_k + \sum_{i\in\mathcal{K}\setminus\{k\}} \mathbf{h}(\mathbf{x}, \theta_k)^H \mathbf{w}_i s_i + \eta_k, \tag{3}$$

where $s_k \sim \mathcal{CN}(0, 1)$ accounts for the data symbol for user k and $\eta_k \sim \mathcal{CN}(0, \sigma_k^2)$ indicates the additive Gaussian noise with zero mean and variance σ_k^2. Accordingly, the SINR of user k is given by

$$\gamma_k(\mathbf{x}, \mathbf{W}) = \frac{\left|\mathbf{w}_k^H \mathbf{h}(\mathbf{x}, \theta_k)\right|^2}{\sum_{i\in\mathcal{K}\setminus k} \left|\mathbf{w}_i^H \mathbf{h}(\mathbf{x}, \theta_k)\right|^2 + \sigma_k^2}, \tag{4}$$

where $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_K] \in \mathbb{C}^{N\times K}$.

In this paper, we aim to maximize the sum rate performance of the multi-user FAS by jointly optimizing the beamforming $\mathbf{W}$ and the location $\mathbf{x}$. The sum rate, denoted by $R(\mathbf{x}, \mathbf{W})$ can be calculated as

$$R(\mathbf{x}, \mathbf{W}) = \sum_{k\in\mathcal{K}} \log_2\left(1 + \gamma_k(\mathbf{x}, \mathbf{W})\right). \tag{5}$$

The corresponding optimization task can be formulated as

$$\max_{\mathbf{W},\mathbf{x}} \quad R(\mathbf{x}, \mathbf{W}) \tag{6a}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} \|\mathbf{w}_k\|^2 \leq P_{\mathsf{max}}, \tag{6b}$$

$$x_n \in [0, D], \forall n \in \mathcal{N}, \tag{6c}$$

$$x_n - x_{n-1} \geq \Delta, \ \forall n \in \mathcal{N}, \tag{6d}$$

where P_{max} denotes the transmit power budget at the BS. Notice that at the optimal, we set $x_1 = 0$ without loss of the optimality [14]. Due to the coupled beamforming and FA position vectors in the objective function, problem (6) is generally nonconvex, making it challenging to identify the globally optimal solution.

B. Preliminaries

We briefly introduce the GNN architecture. According to the heterogeneity of graphs, the GNN can be classified into two categories: HomGNN and HetGNN. The HomGNN is developed for homogeneous graphs consisting of single-type nodes. Let $\mathcal{G}_{\mathsf{hom}} = (\mathcal{K}, \mathcal{E})$ be a homogeneous graph consisting of K nodes $\mathcal{K} \triangleq \{1, \cdots, K\}$ and their edge connections $\mathcal{E} \triangleq \{(k, n) : \forall k, n \in \mathcal{K}\}$. The HomGNN updates hidden node features through a layered architecture. Let $\mathbf{v}_k^{(l)} \in \mathbb{R}^{V_l}$ be the feature vector of node $k \in \mathcal{K}$ at the l-th HomGNN layer. We denote $\mathbf{e}_{kn} \in \mathbb{R}^E$ be the feature vector of edge $(k, n) \in \mathcal{E}$. The l-th HomGNN layer consists of pooling and combining steps. These can be expressed as [11], [12]

$$\mathbf{a}_k^{(l)} = \mathsf{PL}^{(l)}(\{\mathsf{NN}_{\mathsf{PL}}^{(l)}(\mathbf{v}_n^{(l-1)}, \mathbf{e}_{kn}) : \forall n \in \mathcal{K}(k)\}), \tag{7a}$$

$$\mathbf{v}_k^{(l)} = \mathsf{NN}_{\mathsf{CB}}^{(l)}(\mathbf{v}_k^{(l-1)}, \mathbf{a}_k^{(l)}), \tag{7b}$$

where $\mathbf{a}_k^{(l)}$ is the aggregated feature of node k, $\mathsf{PL}^{(l)}(\cdot)$ equals the pooling operator, $\mathcal{K}(k)$ accounts for neighbors of node k, and $\mathsf{NN}_{\mathsf{PL}}^{(l)}(\cdot)$ and $\mathsf{NN}_{\mathsf{CB}}^{(l)}(\cdot)$ respectively denote neural network modules dedicated to the pooling and combining, respectively.

In contrast, the HetGNN handles heterogeneous graphs with T different node types. Unlike HomGNN, which applies a single set of MLPs to all nodes, the HetGNN employs distinct MLPs for different node types. This enables type-aware feature update mechanisms. For brevity, we focus on a special case of the HetGNN with $T = 2$ node types [3]. Let $\mathcal{N}$ and $\mathcal{K}$ be the node sets with two different types. Also, the set of edges is denoted by $\mathcal{E} = \{(n, k) : \forall n \in \mathcal{N}, k \in \mathcal{V}\}$. Then, the heterogeneous graph $\mathcal{G}_{\mathsf{het}}$ can be expressed as $\mathcal{G}_{\mathsf{het}} = (\mathcal{N}, \mathcal{K}, \mathcal{E})$. Notice that the inference of the HetGNN can be designed similarly to (7).

Both the HomGNN and HetGNN can be employed for the optimization of the multi-user FAS. According to [14], the FAS can be characterized as a homogeneous graph where each node

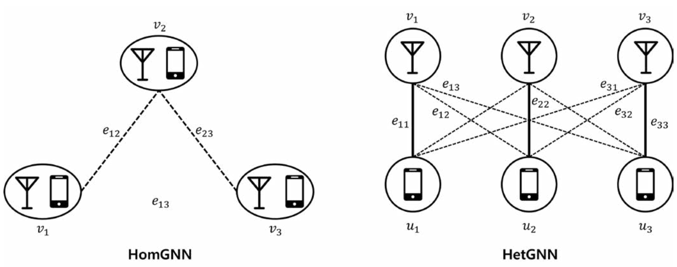

Fig. 2. HomGNN and HetGNN for multi-user FASs.

represents a BS-user pair (see Fig. 2). The beamforming vector $\mathbf{w}_k$ of user k can be determined using the node feature $\mathbf{v}_k^{(l)}$. Although this guarantees the scalability to the user population K, the overall inference becomes rigid in terms of the number of FAs N. Thus, it loses the generalization capability for arbitrary configurations (N, K).

This issue can be addressed by using the HetGNN approach, where FAs and users are modeled by two different types of nodes in a heterogeneous graph. Edge (n, k) connecting antenna $n \in \mathcal{N}$ and user $k \in \mathcal{K}$ interprets a scalar fading channel $[\mathbf{h}(\mathbf{x}, \theta_k)]_n$. The HetGNN has proven to satisfy the permutation equivariance property, ensuring invariance to node reordering and scalability to varying graph sizes (N, K) [13]. As a result, the HetGNN is a perfect candidate for establishing scalable FAS optimizers.

III. Proposed Method

This section proposes a HetGNN approach for solving (6). As shown in Fig. 3, the proposed learning architecture comprises two subsequent stages. In the first stage, the HetGNN $\mathsf{G}_1(\cdot; \varphi_1)$ with parameter φ_1 determines the antenna position, whereas the HetGNN $\mathsf{G}_2(\cdot; \varphi_2)$ with parameter φ_2 in the second stage optimizes the beamforming vectors based on the optimized antenna position.

A. FA position optimization

In this subsection, we develop the first HetGNN $\mathsf{G}_1(\cdot; \varphi_1)$ which identifies the FA position vector $\mathbf{x}$. It is nontrivial to learn the minimum antenna spacing constraint in (6d) using neural networks. To this end, we introduce an auxiliary variable $s_n \triangleq x_n - x_{n-1} - \Delta \geq 0$, which satisfies $s_1 = x_1 - x_0 - \Delta = x_1$ and

$$\sum_{n \in \mathcal{N}} s_n \leq D - (N-1)\Delta \triangleq s_{\mathsf{max}}. \tag{8}$$

It is not difficult to show that the FA position x_n can be equivalently retried from $\mathbf{s} \triangleq [s_1, \cdots, s_N]^T$, as

$$x_n = (n-1)\Delta + \sum_{i=1}^{n-1} s_i. \tag{9}$$

To yield the auxiliary variable $\mathbf{s}$ satisfying (8), we alternatively create intermediate features $\boldsymbol{\xi} \triangleq [\xi_1, \cdots, \xi_N, \xi_{N+1}]^T$ with $\xi_1 = 0$ such that [14]

$$s_n = \frac{s_{\mathsf{max}} \xi_n}{\sum_{i \in \mathcal{N}} \xi_i} \mathsf{sigmoid}(\xi_{N+1}), \tag{10}$$

where $\mathsf{sigmoid}(q) \triangleq (1 + e^{-q})^{-1}$ represent the sigmoid function. Consequently, the FA position vector $\mathbf{x}$ can be constructed using the activation function $\mathsf{X}(\cdot)$.

$$\mathbf{x} = \mathsf{X}(\boldsymbol{\xi}), \tag{11}$$

where $\mathsf{X}(\cdot)$, when receives $\boldsymbol{\xi}$ as input, applies (9) and (10) to each ξ_n and outputs $\mathbf{x}$.

Based on this, we can design the HetGNN $\mathsf{G}_1(\cdot; \varphi_1)$ which learns an efficient FA location vector $\mathbf{x}$. As shown in Fig. 3, it consists of two distinct node sets $\bar{\mathcal{N}} \triangleq \{1, \cdots, N, N+1\}$ and $\mathcal{K}$, each of which models antenna position features ξ_n, $\forall n \in \bar{\mathcal{N}}$, and user $k \in \mathcal{K}$. Then, the corresponding heterogeneous graph $\mathcal{G}_{1,\mathsf{het}}$ is expressed as $\mathcal{G}_{1,\mathsf{het}} = (\bar{\mathcal{N}}, \mathcal{K}, \bar{\mathcal{E}})$, where $\bar{\mathcal{E}} \triangleq \{(n, k) : \forall n \in \bar{\mathcal{N}}, k \in \mathcal{K}\}$. Since there are no physical relationships among $\bar{\mathcal{N}}$ and $\mathcal{K}$, edge features are ignored in constructing $\mathsf{G}_1(\cdot; \varphi_1)$.

Let $\mathbf{v}_n^{(l)} \in \mathbb{R}^{V_l}$ and $\mathbf{u}_k^{(l)}$ be the feature vectors of nodes $n \in \bar{\mathcal{N}}$ and $k \in \mathcal{K}$ at the l-th layer of $\mathsf{G}_1(\cdot; \varphi_1)$ $(l = 1, \cdots, L_1)$. With slight abuse of notations, $\mathbf{v}_n^{(0)}$ and $\mathbf{u}_k^{(0)}$ denote scalar node features at an input layer, which are respectively set to

$$\mathbf{v}_n^{(0)} = (n-1)\Delta \text{ and } \mathbf{u}_k^{(0)} = \theta_k. \tag{12}$$

In the proposed HetGNN, we employ the pooling operator $\mathsf{PL}^{(l)}(\cdot)$ with the graph attention (GAT) mechanism [15], yielding weighted aggregations of neighboring node features. As a result, the update rule of the node feature $\mathbf{v}_n^{(l)}$ is written by

$$\mathbf{v}_n^{(l)} = \mathsf{ReLU}\Big(\alpha_{nn}^{(l)} \mathbf{W}_V^{(l)} \mathbf{v}_n^{(l-1)} + \sum_{k \in \mathcal{K}} \alpha_{nk}^{(l)} \mathbf{W}_U^{(l)} \mathbf{u}_k^{(l-1)}\Big), \tag{13}$$

where $\mathsf{ReLU}(\cdot)$ is the rectified linear unit (ReLU) activation, $\mathbf{W}_V^{(l)}$ and $\mathbf{W}_U^{(l)}$ indicate learnable weight matrices, and $\alpha_{nj}^{(l)}$ stands for the attention weight which represents the importance of node $j \in \mathcal{K} \bigcup \{n\}$ measured at node n. It is calculated as

$$\alpha_{nj}^{(l)} = \frac{\mathsf{LeakyReLU}(c_{nj}^{(l)})}{\sum_{i \in \mathcal{K} \bigcup \{n\}} \mathsf{LeakyReLU}(c_{ni}^{(l)})}, \tag{14}$$

where $\mathsf{LeakyReLU}(\cdot)$ is the leaky ReLU activation and $c_{nj}^{(l)}$ denotes the attention coefficient obtained as

$$c_{nj}^{(l)} = \mathbf{a}_V^{(l),T} \mathbf{W}_V^{(l)} \mathbf{v}_n^{(l-1)} + \mathbf{a}_U^{(l),T} \mathbf{W}_U^{(l)} \mathbf{u}_j^{(l-1)} \tag{15}$$

with $\mathbf{a}_V^{(l)}$ and $\mathbf{a}_U^{(l)}$ being the learnable weights. The feature vector $\mathbf{u}_k^{(l)}$ of user node k is attained similarly to (13)-(15).

To produce the FA position feature ξ_n, the node feature $\mathbf{v}_n^{(L_1)}$ at the last layer is processed by a multi-layer perceptron (ML) $\mathsf{MLP}_1(\cdot)$ as

$$\xi_n = \mathsf{MLP}_1(\mathbf{v}_n^{(L_1)}). \tag{16}$$

The FA position vector $\mathbf{x}$ is then retrieved using the activation function $\mathsf{X}(\cdot)$ in (11). Note that the inputs to the HetGNN $\mathsf{G}_1(\cdot; \varphi_1)$ in (12) can be expressed as two vectors $\mathbf{v} \triangleq [0, \Delta, \cdots, N\Delta]^T \in \mathbb{R}^{N+1}$ and $\boldsymbol{\theta} \triangleq [\theta_1, \cdots, \theta_K]^T$. Consequently, its inference process can be written by

$$\mathbf{x} = \mathsf{G}_1(\mathbf{v}, \boldsymbol{\theta}; \varphi_1). \tag{17}$$

With the optimized the FA position $\mathbf{x}$, we can readily determine the channel vectors $\mathbf{h}(\mathbf{x}, \theta_k)$, $\forall k \in \mathcal{K}$, which will be utilized in the beamforming optimization stage.

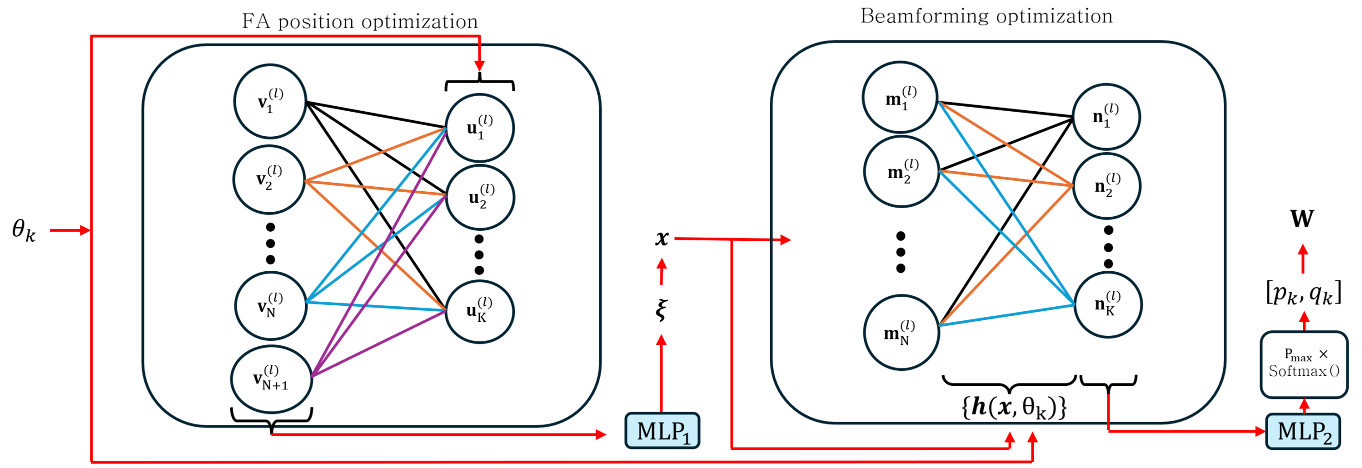

Fig. 3. Proposed HetGNN architecture.

B. Beamforming optimization

Next, we consider the HetGNN $\mathsf{G}_2(\cdot;\varphi_2)$ in the second stage which optimizes the beamforming matrix $\mathbf{W}$. To this end, we exploit the beam feature learning strategy [2], [3]. Instead of generating high-dimensional beamforming vectors, this method allows to learn its low-dimensional sufficient statistics, thereby leading to efficient learning structures. According to the uplink-downlink duality, the optimal beamforming vector can be parameterized through the downlink power allocation vector $\mathbf{p} \triangleq [p_1, \cdots, p_K]^T$ and its dual uplink power allocation vector $\mathbf{q} \triangleq [q_1, \cdots, q_K]^T$ [1]. For given $\mathbf{p}$ and $\mathbf{q}$, the beamforming vector $\mathbf{w}_k$ can be computed using the activation function $\mathsf{W}(\cdot)$ as

$$\begin{aligned}\mathbf{w}_k &= \mathsf{W}(\mathbf{p}, \mathbf{q}) \\ &\triangleq \sqrt{p_k}\frac{(\mathbf{I} + \sum_{i\in\mathcal{K}} q_i \mathbf{h}(\mathbf{x}, \theta_i)\mathbf{h}(\mathbf{x}, \theta_i)^H)^{-1}\mathbf{h}(\mathbf{x}, \theta_k)}{\left\|(\mathbf{I} + \sum_{i\in\mathcal{K}} q_i \mathbf{h}(\mathbf{x}, \theta_i)\mathbf{h}(\mathbf{x}, \theta_i)^H)^{-1}\mathbf{h}(\mathbf{x}, \theta_k)\right\|}. \quad (18)\end{aligned}$$

Here, the power allocation vectors $\mathbf{p}$ and $\mathbf{q}$ should satisfy the total power constraints given by

$$\sum_{k\in\mathcal{K}} p_k = \sum_{k\in\mathcal{K}} q_k = P_{\mathsf{max}}. \quad (19)$$

As depicted in Fig. 3, the heterogeneous graph $\mathcal{G}_{2,\mathsf{het}}$ in the second stage contains antennas and users as distinct node sets $\mathcal{N}$ and $\mathcal{K}$. Thus, it can be expressed as $\mathcal{G}_{2,\mathsf{het}} = (\mathcal{N}, \mathcal{K}, \mathcal{E})$, where $\mathcal{E} \triangleq \{(n,k) : \forall n \in \mathcal{N}, k \in \mathcal{K}\}$. Let $\mathbf{m}_n^{(l)} \in \mathbb{R}^{M_l}$ and $\mathbf{n}_l^{(b)} \in \mathbb{R}^{N_l}$ respectively be hidden feature vectors of antenna node $n \in \mathcal{N}$ and user node $k \in \mathcal{K}$ at the l-th layer of $\mathsf{G}_2(\cdot;\varphi_2)$ $(l = 1, \cdots, L_2)$. Scalar input features $\mathbf{m}_n^{(0)}$ and $\mathbf{n}_k^{(0)}$ are respectively given by

$$\mathbf{m}_n^{(0)} = x_n \text{ and } \mathbf{n}_k^{(0)} = \theta_k. \quad (20)$$

In addition, we set the feature vector $\mathbf{e}_{nk} \in \mathbb{R}^2$ of edge $(n,k) \in \mathcal{E}$ to the associated channel coefficient as

$$\mathbf{e}_{nk} = \Big[\Re\{[\mathbf{h}(\mathbf{x}, \theta_k)]_n\}, \Im\{\mathbf{h}(\mathbf{x}, \theta_k)]_n\}\Big]^T, \quad (21)$$

where $\Re\{Z\}$ and $\Im\{Z\}$ account for real and imaginary parts of a complex number Z.

To incorporate both the user node feature $\mathbf{n}_k^{(l-1)}$ and the edge feature $\mathbf{e}_{nk}$, at the l-th layer of $\mathsf{G}_2(\cdot;\varphi_2)$, the feature vector $\mathbf{m}_n^{(l)}$ of antenna node n is updated as

$$\mathbf{m}_n^{(l)} = \mathsf{ReLU}\Big(\beta_{nn}^{(l)}\mathbf{Q}_M^{(l)}\mathbf{m}_n^{(l-1)} + \sum_{k\in\mathcal{K}} \beta_{nk}^{(l)}\mathbf{Q}_N^{(l)}\mathbf{n}_k^{(l-1)}\Big), \quad (22)$$

where $\mathbf{Q}_M$ and $\mathbf{Q}_N$ are learnable parameters and the attention weight $\beta_{nj}^{(l)}$ can be attained as

$$\beta_{nj}^{(l)} = \frac{\mathsf{LeakyReLU}(d_{nj}^{(l)})}{\sum_{i\in\mathcal{K}\bigcup\{n\}} \mathsf{LeakyReLU}(d_{ni}^{(l)})}. \quad (23)$$

Here, the attention coefficient $d_{nk}^{(l)}$ is computed as

$$\begin{aligned}d_{nj}^{(l)} &= \mathbf{b}_M^{(l),T}\mathbf{Q}_M^{(l)}\mathbf{m}_n^{(l-1)} + \mathbf{b}_N^{(l),T}\mathbf{Q}_N^{(l)}\mathbf{n}_j^{(l-1)} \\ &\quad + \mathbf{b}_E^{(l),T}\mathbf{Q}_E^{(l)}\mathbf{e}_{nj}, \quad (24)\end{aligned}$$

where $\mathbf{b}_\mathsf{Z}$, $\forall \mathsf{Z} \in \{M, N, E\}$, are learnable parameters.

We leverage an MLP $\mathsf{MLP}_2(\cdot)$ to convert the user node feature $\mathbf{n}_k^{(L_2)} \in \mathbb{R}^2$ at the last layer into the corresponding beam feature $[p_k, q_k]^T \in \mathbb{R}^2$. To fulfill the sum power constraint (19), the scaled softmax activation function is employed as

$$[p_k, q_k]^T = P_{\mathsf{max}}\frac{\exp(\mathsf{MLP}_2(\mathbf{n}_k^{(L_2)}))}{\sum_{i\in\mathcal{K}}\exp(\mathsf{MLP}_2(\mathbf{n}_i^{(L_2)}))}. \quad (25)$$

Then, we recover the beamforming vector $\mathbf{w}_k$ using the activation function $\mathsf{W}(\cdot)$ in (18). The input features of the HetGNN $\mathsf{G}_2(\cdot;\varphi_2)$ in (20) and (21) are consolidated into the optimized FA position vector $\mathbf{x}$ and the angle vector $\boldsymbol{\theta}$. Combining (20)-(25) leads to the inference of $\mathsf{G}_2(\cdot;\varphi_2)$ as

$$\mathbf{W} = \mathsf{G}_2(\mathbf{x}, \boldsymbol{\theta}; \varphi_2) = \mathsf{G}_2(\mathsf{G}_1(\mathbf{v}, \boldsymbol{\theta}; \varphi_1), \boldsymbol{\theta}; \varphi_2). \quad (26)$$

C. Training and implementation

We provide a training algorithm that optimizes two HetGNNs $\mathsf{G}_1(\cdot;\varphi_1)$ and $\mathsf{G}_2(\cdot;\varphi_2)$ jointly in an end-to-end manner. Let $\mathsf{F}(\cdot;\psi)$ with $\psi \triangleq \{\varphi_1, \varphi_2\}$ be the inference procedure of the proposed two-stage HetGNN in (17) and (26). Its input-output relationships can be written by

$$(\mathbf{x}, \mathbf{W}) = \mathsf{F}(\mathbf{v}, \boldsymbol{\theta}; \psi). \tag{27}$$

The training objective function $J(\psi)$ is set to the average sum rate as

$$J(\psi) = \frac{1}{|\Theta|} \sum_{\boldsymbol{\theta} \in \Theta} R(\mathsf{F}(\mathbf{v}, \boldsymbol{\theta}; \psi)), \tag{28}$$

where Θ with the cardinality $|\Theta|$ indicates the training dataset containing numerous instances of angle vectors $\boldsymbol{\theta}$.

The proposed two-stage HetGNN is trained to maximize the average sum rate performance via the mini-batch stochastic gradient descent (SGD) algorithm. The update strategy of the parameter ψ is given by

$$\psi \leftarrow \psi + \eta \frac{1}{|\mathcal{B}|} \sum_{\boldsymbol{\theta} \in \mathcal{B}} \nabla_{\psi} R(\mathsf{F}(\mathbf{v}, \boldsymbol{\theta}; \psi)), \tag{29}$$

where $\eta > 0$ is the learning rate, $\mathcal{B} \subset \Theta$ with the cardinality $|\mathcal{B}|$ represents the mini-batch set, and ∇_Z accounts for the gradient operator with respect to Z.

After the offline training, the optimized HetGNNs $\mathsf{G}_1(\cdot; \varphi_1)$ and $\mathsf{G}_2(\cdot; \varphi_2)$ are implemented at the BS to produce efficient FA position and beamforming vectors in an online manner. Since the inference calculations of the HetGNNs are independent of the number of antennas and users, they can be directly applied to diverse FAS configurations with different N and K. Notice that the combining strategies in (14) and (23) can be realized in parallel over the sets of nodes $\bar{\mathcal{N}}$, $\mathcal{N}$, and $\mathcal{K}$. Therefore, the time complexity of the proposed two-stage HetGNN is also independent of the network size.

IV. Simulation Results

This section demonstrates the effectiveness of the proposed HetGNN approach through numerical simulations. The signal-to-noise ratio $P_{\max}/\sigma_k^2$ and wavelength λ are set to 10 dB and $\lambda = 0.167$ m, respectively. Also, the minimum antenna spacing Δ and the length D of FA arrays are fixed as $\Delta = \lambda/2$ and 10λ, respectively [14]. The angles $\theta_k \in [0, \pi]$, $\forall k \in \mathcal{K}$, are uniformly distributed.

We employ $L_1 = L_2 = 4$ HetGNN layers both for the first and second stages. To improve the expressive power, multi-head extensions are adopted for the GAT operations in (14) and (23) with 4 heads. Thus, the dimension of the node feature vectors at the l-th layer is given by 4^l. The MLPs $\mathsf{MLP}_1(\cdot)$ and $\mathsf{MLP}_2(\cdot)$ consist of three hidden layers with dimensions 256, 100, and 50. For the training, we adopt the Adam optimizer with learning rate $\eta = 2 \times 10^{-5}$ and mini-batch sizes 1024. To verify the scalability, the proposed HetGNN with a fixed FAS configuration $(N, K) = (8, 8)$ is applied to various systems with unseen numbers of FAs and users.

For comparison, we consider the following benchmarks.

- *Conventional [14]:* The conventional HomGNN is adopted which lacks the generalization ability to the number of FAs. Thus, this method needs to be trained for each given N.
- *WMMSE [10]:* Locally optimal beamforming vectors are obtained using the WMMSE algorithm under the fixed antenna configuration $x_n = (n-1)D/(N-1)$.

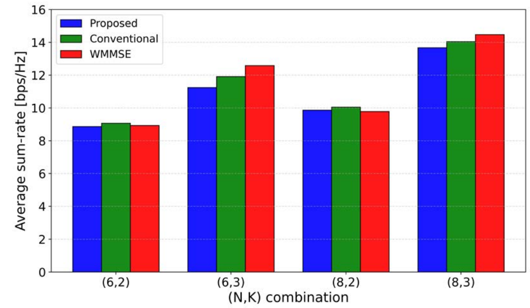

Fig. 4. Average sum-rate performance for various FAS configurations (N, K).

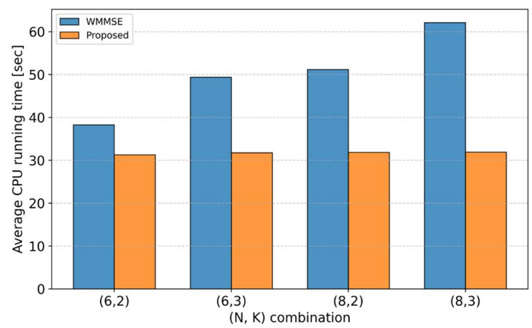

Fig. 5. Average CPU running time for various FAS configurations (N, K).

Fig. 4 plots the average sum rate performance for various FAS configurations (N, K) with $N \in \{6, 8\}$ and $K \in \{2, 3\}$. Both the proposed and conventional GNN methods show only slight performance loss to the locally optimal WMMSE algorithm. The proposed HetGNN approach, which is scalable to both the number of FAs N and the number of users K, exhibits performance comparable to the conventional HomGNN that requires dedicated training for each given N. This demonstrates the scalability of the proposed method over the existing scheme [14]. The proposed method shows a slight performance loss compared to the WMMSE algorithm.

Fig. 5 evaluates the average CPU running time of the proposed HetGNN and WMMSE methods. Due to the parallel computation capability, the time complexity of the proposed method remains unchanged over any N and K. In contrast, the WMMSE generally requires more computations as the system size gets larger, thereby increasing the CPU running time for large N and K. This confirms the scalability of the proposed approach in terms of online inference time.

V. Conclusions

This paper has presented a HetGNN approach to maximize the sum rate of the multi-user FAS. The proposed learning architecture comprises two sequential HetGNNs, each of which optimizes FA locations and beamforming vectors. Compared to existing HomGNN methods which have rigid inference structures in terms of the number of FAs, the proposed

HetGNN can be straightforwardly generalized to various FASs with arbitrary numbers of FAs and users. With the parallel computation, we can further secure the scalability in terms of the time complexity. Numerical results have confirmed the superiority of the proposed scheme to conventional methods.

ACKNOWLEDGEMENT

This work was supported in part by a National Research Foundation of Korea grant funded by the Korea Government (MSIT) under Grant 2021R1I1A3054575 and Grant RS-2022-NR070834.

REFERENCES

[1] E. Björnson, M. Bengtsson, and B. Ottersten, "Optimal multiuser transmit beamforming: A difficult problem with a simple solution structure [lecture notes]," *IEEE Signal Process. Mag.*, vol. 31, no. 4, pp. 142–148, 2014.

[2] J. Kim, H. Lee, S.-E. Hong, and S.-H. Park, "Deep learning methods for universal MISO beamforming," *IEEE Wireless Commun. Lett.*, vol. 9, no. 11, pp. 1894–1898, Nov. 2020.

[3] ——, "A bipartite graph neural network approach for scalable beamforming optimization," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 333–347, Jan. 2023.

[4] G. J. Hayes, J.-H. So, A. Qusba, M. D. Dickey, and G. Lazzi, "Flexible liquid metal alloy (egain) microstrip patch antenna," *IEEE Trans. Antennas Propag.*, vol. 60, no. 5, pp. 2151–2156, Jun. 2012.

[5] A. M. Morishita, C. K. Kitamura, A. T. Ohta, and W. A. Shiroma, "A liquid-metal monopole array with tunable frequency, gain, and beam steering," *IEEE Antennas Wireless Propag. Lett.*, vol. 12, pp. 1388–1391, Oct. 2013.

[6] K.-K. Wong, A. Shojaeifard, K.-F. Tong, and Y. Zhang, "Fluid antenna systems," *IEEE Trans.n Wireless Commun.*, vol. 20, no. 3, pp. 1950–1962, Nov. 2020.

[7] H. Qin, W. Chen, Z. Li, Q. Wu, N. Cheng, and F. Chen, "Antenna positioning and beamforming design for fluid antenna-assisted multi-user downlink communications," *IEEE Wireless Commun. Lett.*, vol. 13, no. 4, pp. 1073–1077, Apr. 2024.

[8] B. Feng, Y. Wu, X.-G. Xia, and C. Xiao, "Weighted sum-rate maximization for movable antenna-enhanced wireless networks," *IEEE Wireless Commun. Lett.*, vol. 13, no. 6, pp. 1770–1774, Apr. 2024.

[9] T. Liao, W. Guo, H. He, S. Song, J. Zhang, and K. B. Letaief, "Joint beamforming and antenna position optimization for fluid antenna-assisted MU-MIMO networks," *IEEE J. Sel. Areas Commun.*, Sep. 2025, to be published.

[10] S. S. Christensen, R. Agarwal, E. De Carvalho, and J. M. Cioffi, "Weighted sum-rate maximization using weighted mmse for MIMO-BC beamforming design," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4792–4799, Dec. 2008.

[11] H. Lee, S. H. Lee, and T. Q. S. Quek, "Learning autonomy in management of wireless random networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 8039–8053, Dec. 2021.

[12] 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. 2020.

[13] Y. Peng, J. Guo, and C. Yang, "Learning resource allocation policy: Vertex-GNN or edge-GNN?" *IEEE Trans. Machine Lear. Commun. Netw.*, vol. 2, pp. 190–209, Jan. 2024.

[14] C. He, Y. Lu, W. Chen, B. Ai, K.-K. Wong, and D. Niyato, "Graph neural network enabled fluid antenna systems: A two-stage approach," *IEEE Trans. Veh. Technol.*, Oct. 2025.

[15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Fev. 2018.