



Graph Neural Network Aided Power Control in Partially Connected Cell-Free Massive MIMO

Shashwat Mishra, Lou Salaun, Hong Yang, Chung Shue Chen

► To cite this version:

Shashwat Mishra, Lou Salaun, Hong Yang, Chung Shue Chen. Graph Neural Network Aided Power Control in Partially Connected Cell-Free Massive MIMO. IEEE Transactions on Wireless Communications, 2024, 23 (9), pp.12412-12423. <10.1109/TWC.2024.3392441>. <hal-04933250>

HAL Id: hal-04933250

<https://hal.science/hal-04933250v1>

Submitted on 6 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Graph Neural Network Aided Power Control in Partially Connected Cell-Free Massive MIMO

Shashwat Mishra*, Lou Salaün*, Hong Yang†, and Chung Shue Chen*

*Nokia Bell Labs, 12 Rue Jean Bart, Massy, 91300 France

†Nokia Bell Labs, 600 Mountain Avenue, Murray Hill, NJ 07974 USA

Email: shashwat.mishra@nokia.com, lou.salaun@nokia-bell-labs.com, h.yang@nokia-bell-labs.com, chung_shue.chen@nokia-bell-labs.com

Abstract—Cell-free massive MIMO (CFmMIMO) is a promising paradigm to provide uniform coverage in future wireless networks. However, a fully connected CFmMIMO system where all the access points (APs) serve every user equipment (UE) makes it challenging to deploy and scale in real-time due to high computational complexity and increased signaling overhead. In this work, we study the problem of downlink power allocation in partially connected CFmMIMO (p-CFmMIMO) systems using maximal ratio transmission (MRT). We utilize the underlying geometry of the problem to propose a graph representation of the CFmMIMO system and develop a graph neural network (GNN) based power allocation strategy to maximize the minimum SINR in the system. We demonstrate that the proposed GNN model has excellent generalizability to deployment size, radio propagation morphologies, and per-AP serving density¹. Our GNN can address the power allocation problem in the fully connected case, the partially connected case, and even the cellular case with magnitudes lower computational complexity compared to the conventional numerical solvers. Notably, we show that over a wide range of service scenarios, the model achieves a median spectral efficiency that is within 10% of the optimal second-order cone programming (SOCP) solution while requiring 100 times fewer FLOPS.

Index Terms—Cell-Free Massive MIMO, graph neural network, power control, access point association, max-min SINR, MRT, conjugate beamforming.

I. INTRODUCTION

The increase in user density and the underlying network infrastructure is one of the instrumental factors driving the development of wireless communications. Notably, in the upcoming sixth-generation (6G) communications, an enormous number of wireless devices are envisioned to collaborate across different deployment scenarios and enabling technologies with the goal of near-ubiquitous, high throughput and high-reliability communication [2]. However, this new landscape presents unique challenges of interference management, deployment scalability and low-latency signal processing, which are critically relevant for dense network scenarios. Cell-free massive multiple-input multiple-output (CFmMIMO) is a promising technology that has attracted significant attention in the wireless communication industry. In traditional massive MIMO systems, a single base station with multiple antennas serves a specific geographic area or cell. In contrast, cell-free massive MIMO uses a large number of distributed antennas

that are not confined to a specific cell. The distributed antennas are typically small and low-cost, making it possible to deploy a large number of antennas across a wide area. This allows for better coverage, improved signal quality, and increased capacity. The study in [3] highlights the importance of distributed antennas to reap the full benefits of massive MIMO. Unlike traditional MIMO systems, CFmMIMO has no fixed cell size. Instead, the system can be easily scaled up or down depending on the needs of the network [4]. Cellular massive MIMO leverages channel hardening and spatial interference suppression [5], but the cell-edge UEs can have bad channel conditions hence a uniform level of service cannot be guaranteed. In contrast, CFmMIMO greatly benefits from strong macro diversity but its interference suppression capability critically depends on the deployment design and the division of operations at local processing at the access points (APs) and the global processing at the central processing unit (CPU). A typical CFmMIMO deployment consists of multiple APs distributed across a wide geographical service area. The work in [6] shows that a higher AP density leads to better coverage. The APs are connected to a CPU via fronthaul links allowing centralized processing of the signals to and from the user equipment (UE). This architecture ensures that all the UEs get uniform coverage as there are no cell boundaries [7]. Additionally, all UEs can ideally be served simultaneously by all the APs to ensure diversity in the signal strength. The work in [8] analyses different possible methods of distributing the signal processing load, such as symbol decoding and channel estimation, among the APs and the CPU. The key finding of the study was that cell-free systems outperform small-cell systems, achieving up to three times higher 90% likely spectral efficiency for users when utilizing fully centralized processing, while incurring a small fronthaul overhead. Several challenges related to the practical deployment of CFmMIMO have been progressively addressed in recent years. There are two critical components concerning user access in CFmMIMO systems, namely pilot assignment and AP-UE association. An effective pilot assignment is crucial for UEs to obtain channel state information (CSI) since it is often not possible to assign orthogonal pilots to all UEs in the system due to the scarcity spectrum. There has been significant progress in the design of pilot allocation schemes [9] which significantly outperform simple greedy heuristics and are not considered in the scope of this work.

¹This work is an extension of our conference paper [1].

Recent work from [10] makes a strong case for the use of GNNs in solving wireless problems since they can effectively exploit the graph topology. Furthermore, GNN based solution of optimization problems on an n -node graph, where the nodes may represent UEs, APs, or antennas, have a generalization error and required number of training samples $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$ times lower than the unstructured multi-layer perceptrons respectively. The authors in [11] highlight that the power allocation policy in multi-cell-multi-user systems exhibits an inherent combination of permutation equivariance properties [12] that can be leveraged using GNNs. The authors utilize a heterogeneous GNN structure for learning the policy and demonstrate that the sample and computational complexities for training the GNN are lower than existing deep neural network designs by an order of magnitude.

Our work in [13] proposed a convolutional neural network (CNN)-based solution for the power allocation problem in CFmMIMO systems. We further introduced a GNN-based solution in [1] for fast power control for MRT in CFmMIMO systems that generalizes well to many different deployment morphologies. The proposed solution utilized a GNN to harness the permutational equivariance property naturally inherent in the case of CFmMIMO, hence outperforming the CNN-based solution in [13]. The comparison between the CNN and the GNN-based approaches is shown in [1]. The solution in [1] achieves near-optimal performance assuming that all UEs are connected to all APs. However, such an assumption is not suitable for large systems, since it can cause significant delays and signaling overhead [14]. Practical user access must maintain the signal processing requirements at each AP under limited complexity and resource requirements to make the system scalable [4]. As the number of APs and UEs increases, the communications and control signaling between APs and the CPU in the fronthaul also increases regardless of the level of cooperation between the CPU and the APs. The work in [4] states that a *scalable* CFmMIMO system must fulfill four conditions related to channel estimation, data transmission, fronthaul signaling, and power control. Since the scope of our current work is power control, we consider the paradigm of allocating only a subset of APs to serve each UE, which we refer to as a p-CFmMIMO.

The authors in [15] have considered the uplink of a scalable CFmMIMO system where the UEs are served only by a subset of APs in the network. The proposed solution utilizes the Hungarian Algorithm for UE-AP association based on the knowledge of the position of the APs in the network, which offers a considerably lower backhaul load with a negligible performance loss compared to fully connected cell-free approaches. However, the optimization of the power control policy is not considered.

In this paper, we address the above issues with the following contributions:

- 1) We generalize our work in [1] by enabling a framework for power allocation in the p-CFmMIMO systems as well as cellular massive MIMO systems. We formulate a new downlink max-min SINR optimization problem with the additional constraint corresponding to the UE-AP association that is supplied by the system designer

as an input and develop a GNN to solve it. The GNN accepts any input size, i.e., any number of APs and UEs, and works for any UE-AP association strategy chosen for the deployment. It is highly scalable with the complexity increasing linearly with the graph's number of edges. We show that our model finds near-optimal solutions with practical run times. The GNN runs up to two orders of magnitudes faster than the conventional SOCP solver while achieving similar max-min SINR performance.

- 2) Our graph neural network design utilizes edge attributes to encode the information about whether a UE is connected to an AP or not, which is provided as input by the system designer. The UE-AP association may be chosen based on implementational complexity, deployment specifications or to meet certain quality-of-service (QoS) requirements. Results show the suitability of our proposed GNN design for power allocation strategy under various UE-AP association schemes, such as many-to-many stable matching and greedy association for CFmMIMO, and a distance-based rule for cellular networks.
- 3) Additionally, we show that the proposed GNN-based solution has an excellent generalization to different graph sizes, corresponding to different numbers of UEs and APs, as well as a wide range of cell-free and cellular radio propagation morphologies and UE-AP association strategies despite being trained on a limited dataset of cell-free scenarios. The performance evaluation reveals that our proposed model has spectral efficiency close to the optimal while executing significantly faster on a variety of morphologies and network sizes.

We re-iterate that the GNN structure in [1] cannot adapt to partially connected cell-free and cellular scenarios, which are of immense importance for practical deployment. In our current work, this generalization is achieved using the novel graph structure that utilizes edge attributes for encoding the UE-AP association that is not present in any existing work so far to the best of our knowledge.

II. SYSTEM MODEL

A. Cell-Free Massive MIMO

We consider the downlink of a CFmMIMO system with M APs deployed throughout the coverage area. All APs are connected to a CPU for precoding and decoding processing, and for power control coordination. Without loss of generality we assume that each AP has one service antenna, and the system is to serve K single-antenna UEs, where M is much larger than K . Each UE can be served by multiple distributed APs simultaneously. We assume that each AP serves no more than $K' \leq K$ UEs and each UE is allocated to at most $M' \leq M$ APs. In practice, K' and M' are chosen to meet constraints on the system's precoding and decoding capabilities, the signaling overhead, and delays requirements.

We denote the channel matrix between the M AP antennas and the K user antennas as:

$$\mathbf{G} = (\mathbf{g}_1 \quad \cdots \quad \mathbf{g}_K) = \begin{pmatrix} \bar{\mathbf{g}}_1^T \\ \vdots \\ \bar{\mathbf{g}}_M^T \end{pmatrix} \in \mathbb{C}^{M \times K},$$

where $\mathbf{g}_k \in \mathbb{C}^M$ is the channel vector between the k -th user and the M AP antennas, and $\bar{\mathbf{g}}_m \in \mathbb{C}^K$ is the channel vector between the m -th AP antenna and the K users, where the superscript T denotes the transpose. The downlink data channel is modeled as :

$$\mathbf{x} = \mathbf{G}^{\text{T}}(\sqrt{\rho_d}\mathbf{s}) + \mathbf{w} \quad (1)$$

where $\mathbf{x} \in \mathbb{C}^K$ is the received signal vector at the K user terminals, ρ_d is the downlink signal to noise ratio (SNR) for each AP, $\mathbf{s} \in \mathbb{C}^M$ is M precoded inputs to the M antenna ports at the M APs, and $\mathbf{w} \in \mathbb{C}^K$ is a circularly-symmetric Gaussian noise vector. Each AP has a downlink power constraint, which can be specified as:

$$\|\mathbb{E}(\mathbf{s}^{\text{T}} \odot \mathbf{s})\|_{\infty} \leq 1. \quad (2)$$

Here, $\mathbb{E}(\cdot)$ is the expectation, the superscript $*$ denotes the complex conjugate transpose and $\odot$ denotes the element-wise multiplication.

B. Downlink Max-Min SINR with MRT

The channel between the m -th service antenna and the k -th user is denoted by:

$$g_{m,k} = \sqrt{\beta_{m,k}} h_{m,k}, \quad (3)$$

where $\beta_{m,k}$ models the large-scale fading that accounts for geometric attenuation and shadow fading, and $h_{m,k}$ models the small-scale fading that accounts for random scattering. In a rich scattering propagation environment, the magnitude of the signal typically varies randomly according to the Rayleigh distribution, thus the small-scaling fading $h_{m,k}$ are modeled as circularly symmetric complex Gaussian, independent and identically distributed random variables.

Under these assumptions, and with MMSE (minimum mean square error) channel estimation based on orthogonal uplink pilot sequence, the ergodic downlink effective SINR for the CFmMIMO system with MRT precoding is given by [16]:

$$\text{SINR}_k^{\text{cell-free}} = \frac{\rho_d \left(\sum_{m=1}^M \sqrt{\alpha_{m,k}} \eta_{m,k} \right)^2}{1 + \rho_d \sum_{m=1}^M \beta_{m,k} \sum_{k'=1}^K \eta_{m,k'}}, \quad (4)$$

where

$$\alpha_{m,k} = \frac{\rho_u \tau \beta_{m,k}^2}{1 + \rho_u \tau \beta_{m,k}}. \quad (5)$$

$\alpha_{m,k}$ is the mean-square of the channel estimate, and ρ_d and ρ_u are the normalized downlink and uplink SNR respectively. τ is the length of the uplink pilot sequence that is used for channel estimation. $\boldsymbol{\eta} = \{\eta_{m,k}\}$ is the downlink power control which is subject to per-AP power constraint:

$$\sum_{k'=1}^K \eta_{m,k'} \leq 1, \quad \forall m. \quad (6)$$

III. PROBLEM FORMULATION

The partially connected power control optimization problem can be formulated as:

$$\begin{aligned} & \max_{\boldsymbol{\eta}} \quad \min_k \quad \text{SINR}_k^{\text{cell-free}}, \\ & \text{subject to} \quad C1 : \sum_{k=1}^{K'} \eta_{m,k} \leq 1, \quad \forall m, \\ & \quad \quad \quad C2 : \eta_{m,k} \geq 0, \quad \forall m, k, \\ & \quad \quad \quad C3 : \eta_{m,k} = 0, \quad \forall (m, k) \in \bar{\Omega}. \end{aligned} \quad (\mathcal{P})$$

The above problem $\mathcal{P}$ is a generalization of the max-min problem formulated in [13] with the addition of constraint $C3$ which account for the partial connectivity between the APs and the UEs. In $C3$, $\bar{\Omega}$ is the set of inactive channels between all UE-AP pairings. An inactive channel implies that AP m does not serve UE k and consequently, the power allocation coefficient between them is set to zero. The set $\bar{\Omega}$ is an input to the problem which is specified by the system designer using the UE-AP association strategy. Furthermore, the set of all active channels between APs and UEs is represented as Ω whereas $\bar{\Omega}$ is its complement, i.e., $\Omega = \{1, \dots, M\} \times \{1, \dots, K\} \setminus \bar{\Omega}$.

An optimal solution to problem $\mathcal{P}$ can be obtained with SOCP feasibility bisection search [16]–[18]. However, SOCP's computational complexity becomes impractical as M and K increase, which would pose a serious challenge for the deployment of real-time future networks.

Our goal is to train a GNN to find a sub-optimal power control that will deliver similar performance as SOCP in terms of spectral efficiency but have a substantially lower runtime, although the power control solutions may be different. This work considers downlink power control under moderate mobility. We assume that the large-scale fading remains substantially the same for the duration of power allocation cycles [8], [19]. Only the large-scale fading coefficients $\beta_{m,k}$ between each pair of AP-UE [20] are used for the design of power control which implies that the fronthaul requirement for the SOCP solution and our proposed GNN solution is the same.

The subject of fronthaul signalling and synchronization for CFmMIMO systems is addressed in detail in section 3.2 of [21]. In our model, the impairments arising due to synchronization errors and limitations of fronthaul signalling affect both the GNN and SOCP solutions in the same manner since we solve the same optimization, and the input (the large scale fading coefficients) is obtained in the same way. The fronthaul signalling makes an effect only on the exchange of power coefficients and the channel gain, not the power control optimization itself.

IV. UE-AP ASSOCIATION

The canonical CFmMIMO framework [16], [17] assumes that all APs in the system serve each UE. However, such a framework is not scalable as the system expands and more APs are added to cover a greater service area. Note that UE-AP association is not the focus of this work and we assume that the system designer provides the association Ω which is most suitable to specific deployment constraint or mobility management requirement. Interested readers may refer to [9],

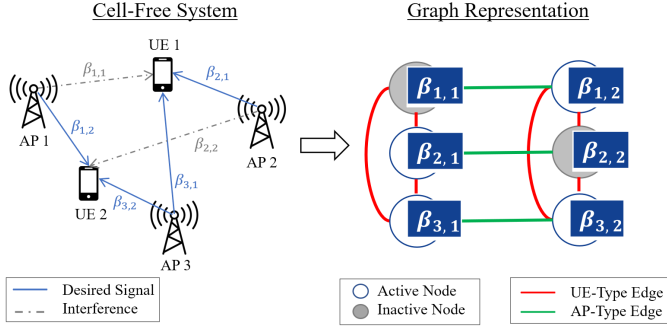


Fig. 1. An overview of the system model and its graphical representation, the edge attribute information is suppressed for the sake of brevity.

[15], [22] for a more rigorous treatment of this aspect. Yet we highlight two simple strategies to demonstrate the effectiveness of the GNN in different deployment scenarios.

- Greedy matching is a simple heuristic for UE-AP association based on the channel gain strength between the APs and the UEs. Since the large-scale fading coefficients for all UE-AP pairs, $\beta = \{\beta_{m,k}\}$, are available before the beginning of power allocation, we use this information to sort the APs in the decreasing order of large-scale fading coefficients to each UE and subsequently match the M' best APs to each UE. This strategy is better than distance-based matching since it inherently takes into account shadow fading and path loss. Additionally, it is feasible even for large systems with a complexity of $\mathcal{O}(M \log M)$.
- Stable matching is a generalization of the one-to-one matching where UEs and APs are associated considering their large-scale fading coefficients as their preferences. The details of this UE-AP association scheme can be found in [22]. Stable matching is Pareto optimal in channel gain, independent of the power control scheme, and completely deterministic with a practical computational complexity of $\mathcal{O}(M^2)$.

V. GRAPH NEURAL NETWORK-BASED POWER CONTROL

A. Heterogeneous Graph Representation

We generate datasets composed of optimal $(\mathbf{B}, \boldsymbol{\eta})$ pairs, which are used to train and evaluate the neural network. Each dataset corresponds to a different simulation scenario characterized by a triplet (M, K, mor) , where $mor \in \{\text{urban}, \text{suburban}, \text{rural}\}$ denotes the deployment and radio propagation morphology. For each example of the dataset, we generate $\mathbf{B}$ with $M \times K$ coefficients, as described in [18] and following the ITU-R specifications [23]. The corresponding optimal power control $\boldsymbol{\eta}$ is obtained by SOCP.

We note that, if the AP indices $\{1, \dots, M\}$ are permuted and/or the UE indices $\{1, \dots, K\}$ are permuted at the input $\mathbf{B}$, then the optimal output would be $\boldsymbol{\eta}$ permuted in the same order. In other words, problem $\mathcal{P}$ does not depend on the choice of indices for the UEs and APs. Note that it is important that an optimal solution to problem $\mathcal{P}$ must satisfy this requirement. This property is called *permutation equivariance*, and GNNs are known to inherently satisfy permutation invariance and equivariance [12] which enables a highly expressive graph representation for our problem.

Our proposed solution is centralized since this GNN is constructed at a central controller, such as a CPU, and the power allocation coefficients are calculated at the central controller before being transmitted to each of the APs. We represent our data as graphs to apply the GNN based solution, as shown in Fig. 1. Instead of directly operating on the links between the APs and the UEs, we first obtain a line graph where the channel between each AP and UE is represented as a node. This representation is more expressive of the power allocation problem and faithfully captures the channel activity between related UEs and APs. We convert each data sample to a directed graph $\mathcal{G} = (V, E)$, where V is the set of nodes and E is the set of directed edges. We create one node for each pair $(m, k) \in \{1, \dots, M\} \times \{1, \dots, K\}$, for a total of MK nodes. The one-to-one mapping between the (m, k) pair and the node index $i \in V = \{1, \dots, MK\}$ is denoted by $\pi(m, k) = i$. Conversely, we have $\pi^{-1}(i) = (m, k)$. Each node $i \in V$ is associated with a tensor h_i called *node feature*. The initial node features are the input $\mathbf{B}$ of the problem: for all $i \in V$, $h_i(0) = \beta_{m,k}$, where $\pi(m, k) = i$. Essentially, each node represents the channel between the AP and the UE. The GNN successively updates the node features through T iterations, i.e., $h_i(t)$, for $t = 1, \dots, T$. The goal is to approximate the optimal output of the problem $\boldsymbol{\eta}$ with the final features: for all $i \in V$, $h_i(T) \approx \eta_{m,k}$, where $\pi(m, k) = i$. The intermediate features, for $t = 1, \dots, T - 1$, are called *hidden features* or *hidden layers*. We define $\mathbf{h}(t)$ as the tensor containing all the node features $h_i(t), i \in V$ at iteration t .

Since the GNN performs local computations on each node and their neighbors connected by an edge, we choose to create one edge between two nodes if they share the same AP or the same UE. There is no self-loop, i.e., $\forall i \in V, (i, i) \notin E$. Let $e \in E$, we define $\text{type}(e) \in \{\text{AP}, \text{UE}\}$ to be the edge type. Each edge is either of type AP if they share the same AP, or of type UE if they share the same UE. Hence, the edge construction is summarized by the following two statements, for all $m, m' \in \{1, \dots, M\}$, $m \neq m'$, and $k, k' \in \{1, \dots, K\}$, $k \neq k'$:

- 1) $e = (\pi(m, k), \pi(m, k')) \in E$ and $\text{type}(e) = \text{AP}$,
- 2) $e = (\pi(m, k), \pi(m', k)) \in E$ and $\text{type}(e) = \text{UE}$.

We consider heterogeneous edge types so the GNN can distinguish between these relationships and apply different operations on each type of edge. Let $i \in V$, define its set of neighbors as:

$$\mathcal{N}(i) = \{j \in V : (i, j) \in E\}.$$

Similarly, we define the set of neighbors of type UE and type AP as:

$$\begin{aligned} \mathcal{N}_{\text{UE}}(i) &= \{j \in V : (i, j) \in E \text{ and } \text{type}(i, j) = \text{UE}\}, \\ \mathcal{N}_{\text{AP}}(i) &= \{j \in V : (i, j) \in E \text{ and } \text{type}(i, j) = \text{AP}\}. \end{aligned}$$

Fig. 2 depicts the neighbors of a typical active node and its associated neighbors and edge attributes.

For $i \in V$, we have:

$$\begin{aligned} \mathcal{N}_{\text{UE}}(i) \cup \mathcal{N}_{\text{AP}}(i) &= \mathcal{N}(i), \\ \mathcal{N}_{\text{UE}}(i) \cap \mathcal{N}_{\text{AP}}(i) &= \emptyset. \end{aligned}$$

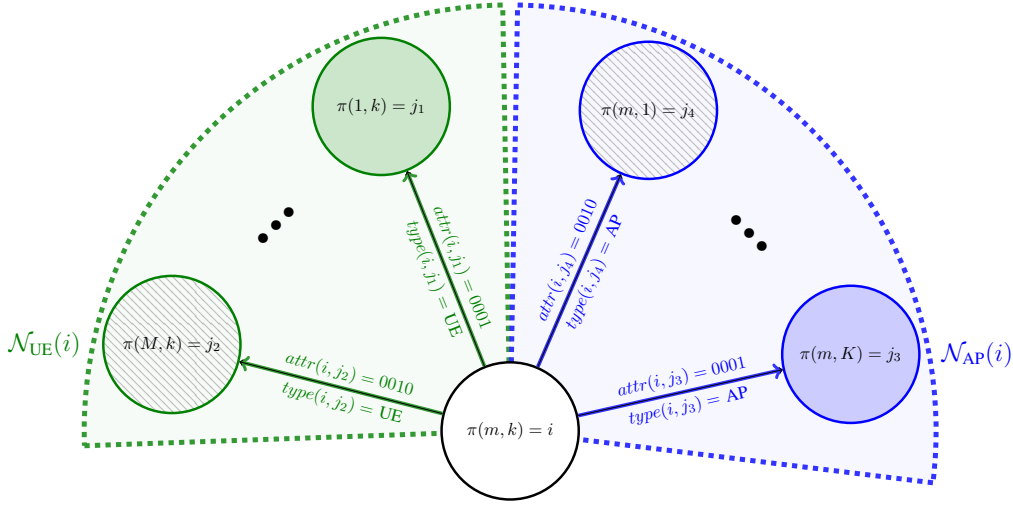


Fig. 2. An active node $\pi(m, k) = i \in V$ where UE k is served by AP m . The two types of neighbors of this node and their corresponding edge attributes are shown here. The nodes in gray with hatched fill represent an inactive node whereas the nodes with solid fill represent an active node.

Source node i	Destination node j	$\text{attr}(i, j)$
Active	Active	0001
Active	Inactive	0010
Inactive	Active	0100
Inactive	Inactive	1000

TABLE I
ONE-HOT ENCODING FOR THE EDGE ATTRIBUTES.

Each $\mathcal{N}_{\text{UE}}(i)$ has $M - 1$ elements, each $\mathcal{N}_{\text{AP}}(i)$ has $K - 1$ elements, and each $\mathcal{N}(i)$ has $M + K - 2$ elements.

This node-edge structure associates nodes with first-order dependence as neighbors while segregating the nodes that share UEs and nodes that share APs. However, to support scalable operations, we must take into account that each AP may serve only a subset of UEs at any given time. This adds another dimension to the edges since now we have several possibilities based on the edge type and whether the channel represented by the edge (m, k) is active or inactive. Additionally, we utilize the edge attributes of the GNN to add this information to the graph through a one-hot encoding corresponding to the channel activity. An active node $\pi(m, k)$ corresponds to UE k being served by AP m , whereas, in an inactive node, the UE is not served by the AP in that node.

We define the function edge attribute as $\text{attr}(e)$ to characterize the edge encoding based on the channel activity of the source nodes and destination nodes. Recall that the set of active channels is given by the set Ω . Thus, we incorporate the channel activity information between UEs and APs, specified as the UE-AP association input given by the designer, using these edge attributes into the GNN. The function is elaborated in Table I.

The combination of {edge type, edge attribute} completely characterizes an edge. Using edge attributes provides more versatility for learning compared to other solutions like scalar edge weights. Additionally, we choose one-hot encoding over the compressed 2-bit representation for edge attributes since one-hot encoding provides each combination with its weight leading to a more generalizable model. The edge attributes are elaborated in Table I.

B. Data Preprocessing

The graph data preprocessing is done similarly to [13]. We first apply a $\log_2$ transformation to all the input and output features. This way, the features are within the same order of magnitude, and the GNN can extract useful information. As an example, if $\beta_{m,k}$ takes values in $[10^{-15}, 10^{-5}]$, then $-50 < \log_2 \beta_{m,k} < -16$. Then, we normalize the resulting values so that in the input features and output features, each has zero mean and unit standard deviation over all examples of the dataset.

C. The Structure of the Neural Network

A broad overview of our proposed network is shown in Fig. 3 which we shall elaborate on in this sub-section. The notation $\mathcal{L}$ denotes the linear operation that takes the tensor x of size n as input and outputs a tensor $\mathcal{L}(x)$ of size m according to the following relation:

$$\mathcal{L}(x) = Wx + b,$$

where $W \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ are trainable parameters referred to as *weight* and *bias*. The linear layers utilized in (7) and (8) do not share any trainable parameter in general, therefore we denote $\mathcal{L}$ with different subscripts and superscripts to remove any ambiguity. The aggregation functions $\text{aggr}_{\bullet}$, for $\bullet \in \{\text{AP}, \text{UE}\}$, are based on the graph transformer in [24, Section 3.1] and they act identically on both edge types. With the above considerations, a single attention layer can be characterized by the following operations:

$$\text{aggr}_{\bullet}(i) = \mathcal{L}_{\bullet}^1(h_i) + \sum_{j \in \mathcal{N}_{\bullet}(i)} MP_{i,j}, \quad (7)$$

The message-passing function between the nodes is defined as follows:

$$MP_{i,j} = \alpha_{\bullet}(i, j) \times (\mathcal{L}_{\bullet}^2(h_j) + \mathcal{L}_{\bullet}^3(\text{attr}(i, j))),$$

where $\alpha_{\bullet}(i, j)$ is the attention coefficient between source node i and destination node j , such that:

$$\alpha_{\bullet}(i, j) = \frac{\langle \mathcal{L}_{\bullet}^4(h_i), (\mathcal{L}_{\bullet}^5(h_j) + \mathcal{L}_{\bullet}^3(\text{attr}(i, j))) \rangle}{\sum_{u \in \mathcal{N}_{\bullet}(i)} \langle \mathcal{L}_{\bullet}^4(h_i), (\mathcal{L}_{\bullet}^5(h_j) + \mathcal{L}_{\bullet}^3(\text{attr}(i, j))) \rangle}, \quad (8)$$

where $\langle x, y \rangle = \exp\left(\frac{x^T y}{\sqrt{d}}\right)$, is the exponential scale dot-product [25] and d is the tensor size of each head. Furthermore, $\text{attr}(i, j)$ is the edge attribute for the edge between source node i and destination node j , defined as a one-hot encoded vector as elaborated in Table I. The edge attribute represents the UE-AP association, indicating the active nodes and which APs serve each of the UEs.

We leverage the notion of multi-headed attention mechanism such that each node can focus specifically on a subset of its neighbors that are of interest rather than imparting equal importance to all its neighbors. We implement each of these functions with $C = 2$ attention heads. We extend the aggregation function defined in (7) to account for this multi-headed attention mechanism. For an iteration t , the aggregation function can now be characterized as $\text{aggr}_{\bullet, c, t}(i)$, for all nodes $i \in V$. The two attention heads corresponding to $c = 1$ and $c = 2$ are processed separately following (7) and then concatenated into a single tensor. This concatenation operation is shown in Fig. 3 at the addition junction. The resultant intermediate node tensor $h'_i(t)$ is denoted as:

$$h'_i(t) = \sum_{\bullet \in \{\text{AP}, \text{UE}\}} \bigoplus_{c=1}^C \text{aggr}_{\bullet, c, t}(i),$$

where $\bigoplus$ denotes the concatenation operation. If the output of each attention head is a tensor of size d , then $h'_i(t)$ is a tensor of size $2d$. The tensor containing all the intermediate node tensors is denoted as $\mathbf{h}'(t)$.

The attention coefficient controls which feature $h_j(t)$ will contribute to $h_i(t+1)$ through (7): the feature is discarded if $\alpha_{\bullet}(i, j) \approx 0$, and kept otherwise. The level of dependence between two APs (or UEs) varies due to factors such as their relative geographic location and channel conditions. We use attention to efficiently learn the complex relationship between nodes. Furthermore, with multiple heads, different attention levels can be assigned to the various features of the same neighbor. Here, we implement $C = 2$ attention heads. A further increase in the number of heads does not significantly improve the performance of this problem. During our study, we observe that the attention mechanism greatly improves the GNN performance and generalizability to a large number of APs and UEs. The stark impact of having an attention mechanism as opposed to no attention is highlighted in the results in Section VI in Table IV and Table V.

The node tensor for the next iteration is obtained after applying ReLU activation followed by layer normalization as:

$$\mathbf{h}(t+1) = \text{Norm}(\text{ReLU}(\mathbf{h}'(t))). \quad (9)$$

Edge features are added to the keys after linear transformation, that is, before computing the attention dot product. They are also added to final values after the same linear

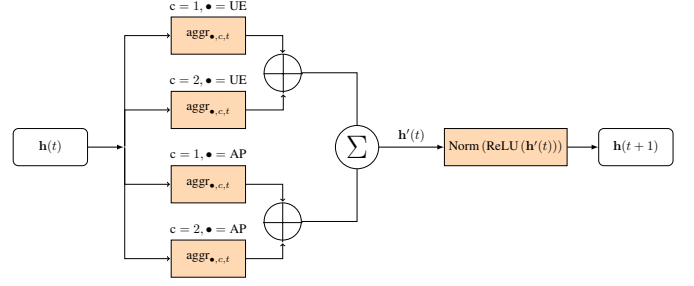


Fig. 3. An overview of one iteration over the input tensor $\mathbf{h}(t)$. $\bigoplus$ and Σ denote respectively the concatenation and summation operators.

transformation. The edge attributes encapsulate the activity status of the nodes on either side of a type UE or type AP edge. Additionally, edge attributes enable the trained model to be versatile and more generalizable compared to other methods such as masking since each combination of {edge type, activity} has its weight that can be tuned for better learning.

We can see that whenever the input is permuted, the π mapping is permuted, but the neighborhood of each $\pi(m, k)$ node remains the same. Furthermore, the operations defined above do not depend on any ordering of the neighbors, since the neighboring features $h_j(t)$, for $j \in \mathcal{N}(i)$, are summed in (7). As a consequence, the GNN guarantees that the output is also permuted equivariantly. Thus, the GNN satisfies the permutation equivariance property of our problem $\mathcal{P}$.

Through experiments, we optimized the structure of the GNN to achieve near-optimal performance with reasonable complexity. In particular, we found the least number of hidden layers and the smallest size of each of the layers such that the spectral efficiency performance is acceptable over a wide range of scenarios while keeping the model parameters limited. This is important for a practical deployment. Our GNN model contains 11 hidden layers, i.e., $T = 12$, with the following node feature tensor sizes:

$$\text{Layer sizes: (in=1, 8, 10, 16, 24, 36, 64, 36, 24, 16, 10, 8, out=1)}.$$

As expected, the input and output both have a single value per node representing respectively the large-scale fading coefficient, and the power control coefficient for each channel. Each hidden layer, for $t = 1, \dots, 12$, is obtained from the previous layer by applying the multi-head attention neural network defined in (9). The final output tensor is obtained by applying a simple linear activation of the form $h_i(T) = \mathcal{L}_{\text{out}}(h_i(T-1))$, for all node $i \in V$. To guarantee that the power constraints in $\mathcal{P}$ are satisfied, we apply the following two operations on the output tensor: (i) If any power coefficient is negative, we set it to zero. (ii) If the power budget constraint is violated for an AP m , then we re-normalize its transmit powers. That is, if $\sum_{k'=1}^K \eta_{m, k'} > 1$, then for all UE k , we assign $\eta_{m, k} \leftarrow \eta_{m, k} / \sum_{k'=1}^K \eta_{m, k'}$.

The choice of layer sizes and the number of attention heads are subject to deployment-dependent hyperparameter tuning. For instance, in our experiments we found that using fewer attention heads result in worse generalization capabilities and

M	K	M'	K'	Number of samples
32	6	32	6	8000
32	9	32	9	8000
64	9	64	9	8000
64	18	64	18	8000
32	6	22	4	12000
32	9	11	3	12000
32	9	15	4	12000
32	9	25	7	12000
64	9	29	4	12000
64	9	36	5	12000
64	9	50	7	12000
64	18	15	4	12000
64	18	25	7	12000
64	18	40	11	12000
64	18	54	15	12000

TABLE II
THE TRAINING DATASET USED FOR TRAINING. ALL THE DATA
CORRESPONDS TO AN URBAN CELL-FREE DEPLOYMENT.

using too many attention heads may be computationally prohibitive without appreciable generalization gains. In an actual deployment, the system designer will have to make pragmatic choices based on the available training data, the processing capabilities of the APs and the desired quality of service [26].

D. Training and SINR Loss Function

The training dataset is composed of 15 scenarios as shown in Table II, consisting of 164,000 training samples. The first 4 cases correspond to a fully connected cell-free network, while the rest of the cases correspond to a p-CFmMIMO. During our experiments, we found that training over only the fully connected cell-free cases does not yield SINR performance sufficiently close to the optimal SOCP solution, even if we utilize the edge attributes to specify the active nodes. The chosen set of partially connected cases allows the model to learn the characteristics of the network over a wide range of per-AP serving capacity without necessarily having to learn each case. A high per-AP serving capacity signifies that almost all UEs are being served by all APs and conversely a low per-AP serving capacity implies each UE is served by a small number of APs. These training datasets have been obtained using *Stable Matching* for UE-AP association for the partially connected cell-free cases.

In comparison, the training dataset in [1] uses 80,000 raw data, considering only the four fully connected cases. The loss function used for training is formulated as:

$$\frac{1}{K} \sum_{k=1}^K (\text{SINR}_k - \text{SINR}'_k)^2 + \frac{c}{K} \sum_{k=1}^K (\eta_k - \eta'_k)^2,$$

where SINR_k is the optimal SINR of user k obtained by SOCP, and SINR'_k is the SINR computed from the GNN predicted power coefficients η' . Furthermore, η_k is the power coefficients of user k obtained by SOCP, η'_k is the GNN predicted power coefficients and c is a tunable parameter. The first term in this expression is the SINR loss for user k in the mean squared error (MSE) sense, while the second term is the MSE signal loss for user k scaled by c . The proposed loss function is differentiable and varies continuously with each user's SINR. We use the Adam optimizer [27] with a learning

rate of 7×10^{-4} to train the GNN model. The batch size is 64, and the training is executed for 100 epochs.

VI. NUMERICAL RESULTS

In this section, we show the performance of our GNN in terms of spectral efficiency, computational complexity, and generalizability for different system sizes and deployment morphologies. We compare it to the results obtained using the optimal SOCP benchmark. To evaluate each algorithm's complexity, we count the number of floating point operations (FLOPs) during execution. The total number of APs in our simulations ranges from $M = 24$ to 128. In partially connected scenarios, the number of APs serving simultaneously the same UE goes from $M' = 11$ to 72. Similarly, the total number of UEs varies from $K = 5$ to 32, while the number of UEs served by each AP in the partially connected case ranges from $K' = 3$ to 18. We consider that the APs and UEs are deployed in a circular disk of a radius of 500 meters for the urban scenario, 1 km for the suburban, and 4 km for the rural following a uniform random distribution, respectively. We utilize the "NLoS" propagation model from [23], and the path loss parameters used are the same as in [13].

The workflow for obtaining the UE-AP association and power allocation is summarized in Fig. 4. The inputs are the large-scale fading coefficients β and the UE-AP association, Ω , encoding the per-AP serving capacity. Note that when we consider a partially connected cell-free system, there is at most $M' < M$ APs available for connection to any user, and equivalently each AP serves at most $K' < K$ users, i.e.,

$$\sum_{m'=1}^M \mathbb{1}(\eta_{m',k}) = M', \quad \forall k, \\ \sum_{k'=1}^K \mathbb{1}(\eta_{m,k'}) = K', \quad \forall m,$$

where $\mathbb{1}(\cdot)$ is the indicator function which takes a value of one only if the input is strictly positive and it is zero otherwise. We obtain the UE-AP association using *Stable Matching* as described in Section IV, unless stated otherwise, and use it as an edge attribute for our GNN model. This process is computationally efficient and can be done on the fly before the inference of power coefficients using the GNN.

The figures in this section show the spectral efficiency cumulative distribution function (CDF) achieved by the different algorithms for various simulation scenarios. We generate 1,000 large-scale fading realizations for each simulation scenario. The *performance loss at median* refers to the relative difference in spectral efficiency between the GNN approach and the optimal solution obtained by SOCP, calculated at the median of the CDF. Furthermore, the *95%-likely performance* refers to the spectral efficiency at the bottom 5-th percentile, which can be inferred as the coverage quality for 95% of the users. This is a crucial metric for evaluating the CFmMIMO deployment since we can provide more uniform coverage to all the users within the area of deployment.

The simulations and algorithms presented in this work are implemented in Python 3. The SOCP problem $\mathcal{P}$ is solved

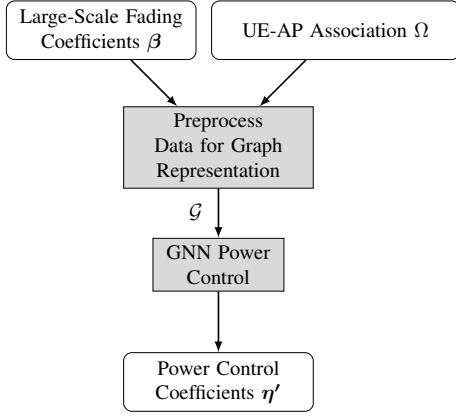


Fig. 4. Workflow for the proposed GNN enhanced power allocation in p-CFmMIMO. The per-AP serving capacity is inferred from the matching Ω .

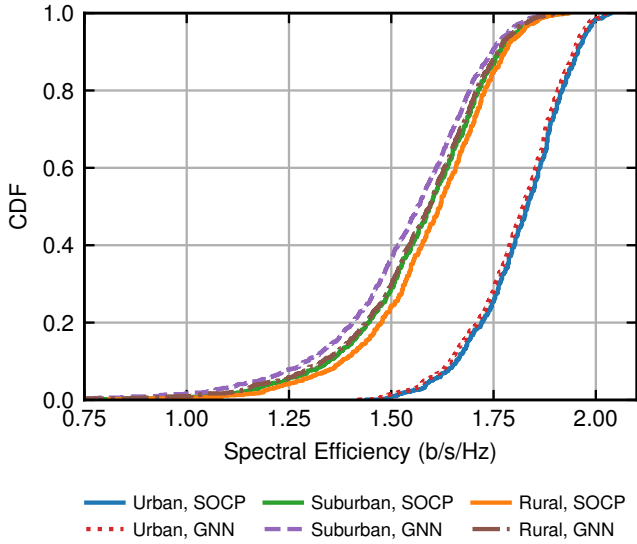


Fig. 5. Performance of the scalable GNN model for different radio propagation morphologies with the same UE density. Here, $M = 32$, $K = 9$ and $K' = 4$.

using Mosek solver [28] combined with a bisection search. The bisection search terminates when the termination threshold for of $\epsilon = 0.01$ in the SINR is reached. The GNN is realized using the PyTorch Geometric library [29]. We evaluate the inference on an Intel Core i7 CPU with 16 GB of RAM running 64 bit Windows 10. In the following sections, we demonstrate the capability of our GNN to generalize, not only to changes in the total number of APs and UEs but also in scenarios where the per-AP serving capacity is different. Note that the performance loss at median for all the cases in the training dataset shown in Table II is less than 2.3%.

A. Performance on Cell-Free Scenarios

In this section, we showcase the generalizability of the GNN on untrained cell-free scenarios, i.e., scenarios that have not been used for training. Fig. 5 shows the performance of the GNN on different radio propagation morphologies. For this evaluation, we choose a partially connected cell-free system with 32 APs and 9 UEs and $K' = 4$, i.e., each AP served at most 4 UEs and the association Ω is

obtained using *Stable Matching*. We observe that the GNN adapts well to the varying morphology of deployment, with less than 2% spectral efficiency loss at median over the rural and suburban morphologies. Note that during its training the GNN only utilized the urban morphology, yet it can perform well on other morphologies. This is because the underlying adjacency structure between the UEs and APs remains the same upon a change in morphology and therefore the permutation equivariance property, as highlighted earlier, still holds. The only change is in the scale of the SINR due to the difference in the service area size and propagation conditions of the morphologies, and the GNN has shown its robustness to these scaling changes.

In general, the spectral efficiency decreases with the decrease in the number of service APs per UE, M' , in the partially connected case when compared to the fully connected case. This loss can be attributed to the reduction in diversity with the decreasing number of APs. The results in Fig. 6 support this analysis, where we see a marked decrease in the spectral efficiency as M' decreases and we move farther away from a fully connected system. For a given deployment, the ratio of M' and M must be carefully chosen to achieve the best trade-off between computational complexity and coverage. A good value of M' in the system may be decided based on UE arrival statistics in the network.

Furthermore, we note that the GNN can equally well handle unseen node densities, despite training on only a limited set of per-AP serving capacity as shown in Fig. 6. Here, the number of service APs per UE is varied for a fixed morphology and network size. In our previous work [1], we established that the GNN performs efficient power allocation irrespective of the size of the input since it learns the essential neighborhood relationships between the nodes. In the case of p-CFmMIMO networks, this strength comes into play again as the attention mechanism allows the GNN to learn the general node relational characteristics of a certain active node distribution rather than relying on the exact number of active nodes present in the network.

In Fig. 7, we evaluate the performance of the GNN on scenarios when each AP serves a fixed number of UEs K' and varying network size and morphology. The GNN has excellent performance over the different network sizes with less than 4% performance loss at median and 5% loss at the green 5-th percentile. A comprehensive evaluation of the GNN performance under various per-AP serving capacity in different morphologies, in terms of the SINR loss compared to the SOCP solution, is presented in Table IV. On all the tested scenarios, the median loss and 95%-likely loss remains respectively under 4% and 9%. It is worth highlighting that without the attention mechanism, we see drastically poor results even though all the other parameters are kept the same, as shown in the last column of Table IV. This demonstrates the importance of the attention mechanism and its efficiency at adapting to the changing network size and morphology.

Furthermore, we explore the impact of a different UE-AP association scheme than the one used for training the GNN. Table III shows the results for our GNN solution, without any re-training for the greedy UE-AP association. We see that

TABLE III
PERFORMANCE OF GNN WITH GREEDY MATCHING

Scenario				GNN SE Loss(%)	
<i>mor</i>	M	K	K'	Median	95%-likely
urban	32	9	4	4.66	8.63
urban	60	10	5	4.31	7.26
sub-urban	55	15	11	5.23	9.18
rural	32	9	4	3.92	7.81

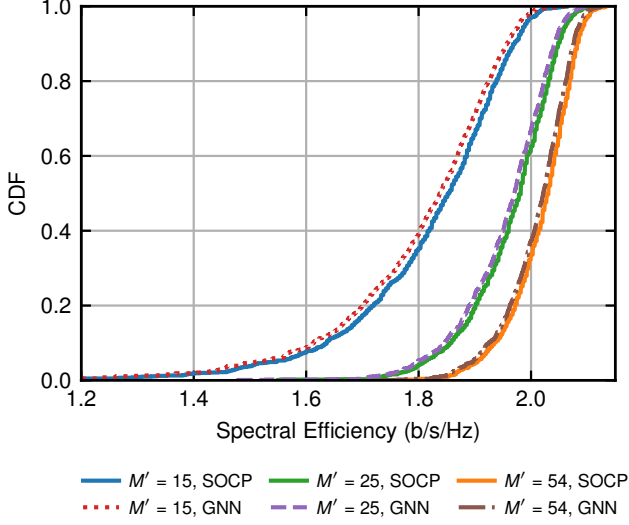


Fig. 6. Performance of the scalable GNN model on urban $M = 64$, $K = 18$ scenario with a varying per-AP serving capacity.

the GNN still maintains an SE-loss under 10% even on new untrained scenarios with Greedy matching. Note that the design of UE-AP association algorithms is not in the scope of our current work. We may train a GNN with different UE-AP association schemes to get even better performance, but these UE-AP association schemes are not optimized by the GNN itself.

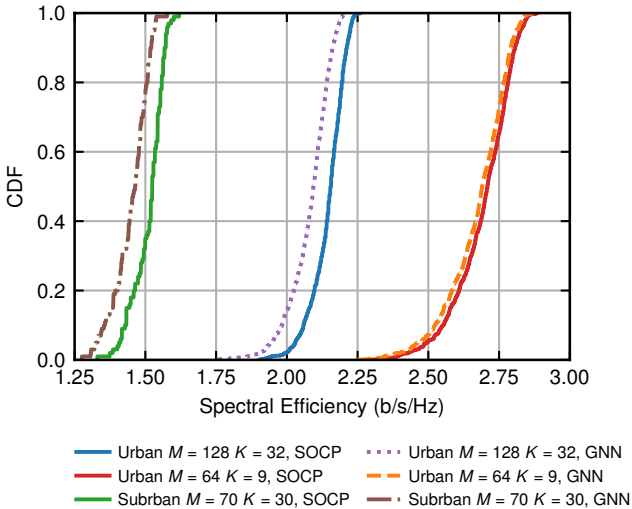


Fig. 7. Performance of the scalable GNN model on different scenario with the same number of UEs served per-AP, $K' = 8$.

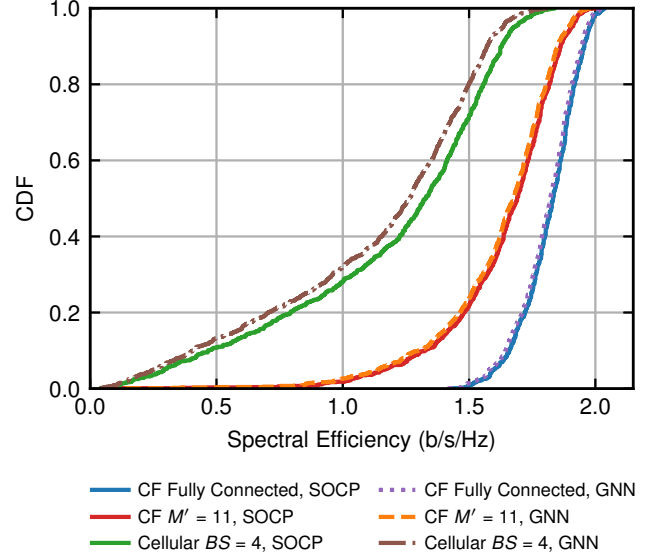


Fig. 8. Performance of the GNN model on cell-free (CF) and cellular deployments in urban $M = 32$, $K = 9$ morphology.

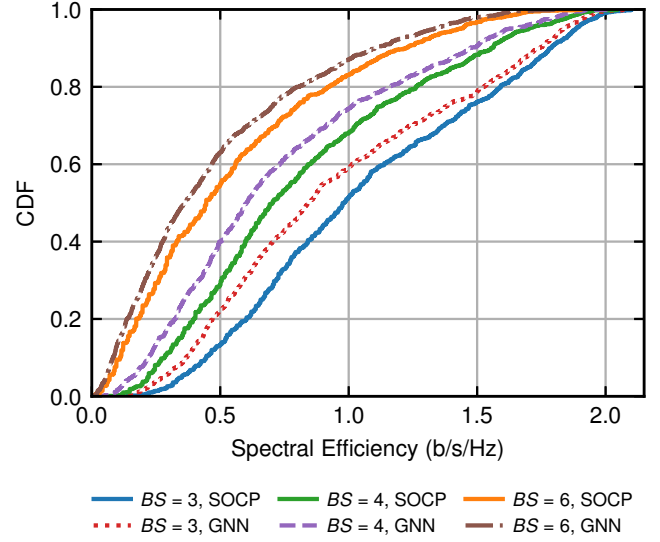


Fig. 9. Performance of the GNN model on different cellular deployments in urban morphology with $M = 48$ and $K = 12$.

B. Performance on Cellular Scenarios

Even though the GNN is trained on cell-free data, it still handles cellular scenarios with ease. This is because CFm-MIMO may be seen as a general case of the cellular system, and as such, the underlying graph structure remains relevant for power control. Since the only differentiating factor here is the UE-AP association, which is given as an input to the GNN, we can faithfully infer the power control coefficients using the same GNN model. For the cellular massive MIMO deployment, we consider $BS \in \{3, 4, 6\}$ base stations, each with $\lceil M/BS \rceil$ antennas, serving $\lceil K/BS \rceil$ users nearest to them, where $\lceil x \rceil$ represents the smallest successive integer

of x . The UE-AP association here follows the Greedy matching approach. These multi-antenna base stations are deployed in the same cell size as the cell-free deployment and distributed such that each of them serves the same fraction of the total geographical service area. More precisely, with 4 base stations, each one serves a quarter of the deployment area. This methodology of simulating cells in the same geographical area as a cell-free system is often adopted in the literature for a fair comparison between the cellular and the cell-free deployments [4]. Fig. 8 presents a comparison between the spectral efficiency achieved by the cell-free and cellular systems. It re-affirms the advantage of cell-free deployment, where the spectral efficiency is higher due to enhanced diversity. The same GNN model can perform real-time power allocation on a wide range of deployments ranging from fully and p-CFmMIMO systems to cellular systems. The performance loss at the median for the GNN solution in the cellular case is about 5%.

Fig. 9 shows the effectiveness of GNN when applied to cellular scenarios with different number of base stations. The GNN performance on cellular scenarios is also summarized in Table V. Note again that without the attention mechanism, we see drastically poor results here as well, as evident from the last column of Table V. We see that even without any additional training the GNN has a maximum performance loss of about 10% while running substantially faster than the SOCP solution. Here we avoid comparing the 95%-likely loss since uniform coverage is not in the design scope of the compared cellular deployment. These results highlight the generalizability of the GNN on cellular scenarios since we did not re-train the GNN for the cellular case. However it must be noted that a performance loss of well under 4% can be achieved with the same GNN model by fine-tuning the training dataset to include some cellular data.

C. Computational Complexity Analysis

We evaluate the computational complexity of the proposed GNN model in terms of FLOPs versus the number of edges of its graph representation. This edge count corresponds to the size of the network and more precisely it is equal to $MK \times (M + K - 2)$. Additionally, each multiplication or addition operation on the processor counts as one FLOP. Recall that the asymptotic computational complexity of the GNN solution in our case is $O(MK(M + K))$ [1]. On the other hand, the asymptotic computational complexity of the SOCP solution is $O(n^{3.5})$ [30], [31], where n is the number of cones. From [18], we see that the SOCP problem has K cones to constrain the K minimum required SINR in (6) and M cones to constrain the M per-antenna power limit in (7). Thus in our case $n = M + K$. On top of this, we have a multiplicative factor of $\log(\epsilon^{-1})$ due to the bisection search where ϵ is the termination threshold. Thus, the total asymptotic computational complexity of the SOCP solver is $O(\log(\epsilon^{-1})(M + K)^{3.5})$, which is considerably higher than the GNN solution.

In Fig. 10, we show the FLOP count of the GNN compared to the FLOP count of the SOCP solution. The GNN outperforms SOCP in all cases with about 10 times fewer FLOPs

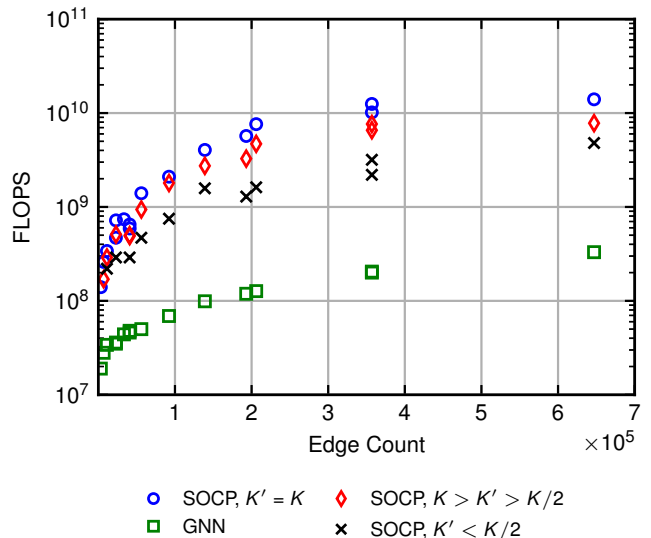


Fig. 10. Comparison of FLOPs for SOCP and GNN. The FLOPs depend on the number of edges in the graph.

at 10,000 edges and close to 100 times fewer FLOPs as the number of edges increase. This demonstrates that the GNN solution is highly preferable over the SOCP method for large network sizes. Furthermore, the computational complexity of the GNN does not depend on the per-AP serving capacity in the system but only depends on the number of edges in the graph. This is in contrast to the SOCP solution where the computational complexity increases with the per-AP serving capacity K' . Notably, the GNN-based solution presented in this work performs power allocation over any per-AP serving capacity at near-constant computational complexity, which is of immense importance for practical deployments. As an example, the GNN takes less than 100 ms to evaluate all of the presented cases with the hardware specifications corresponding to a commodity machine as elaborated in Section VI, which makes it a strong candidate for real-time power allocation in scalable CFmMIMO deployments.

VII. CONCLUSION

We present a GNN-based solution for supporting p-CFmMIMO networks beyond 5G wireless systems. We utilize graph representation of the large-scale fading coefficients between the UEs and the APs to design a robust GNN that performs fast max-min power control for MRT, the corresponding UE-AP association is encoded as edge attributes. We demonstrate the generalizability of our model through extensive simulations over a wide range of morphologies, graph sizes, and number of UEs served by each AP. The proposed GNN model is significantly faster than the conventional SOCP solver while achieving similar spectral efficiency. We observe that the GNN model achieves a median spectral efficiency that is within 4% of the optimal SOCP solution while requiring 10 to 100 times fewer FLOPs under a wide range of scenarios. Additionally, the GNN can perform efficient power allocation even in cellular scenarios without any additional training.

TABLE IV
PERFORMANCE OF GNN AND SOCP FOR DIFFERENT CELL-FREE
SCENARIOS

Scenario					GNN SE Loss(%)		
<i>mor</i>	<i>M</i>	<i>K</i>	<i>K'</i>	<i>K/K'</i>	Median	95%-likely	Median No Attention
urban	24	5	5	1	0.73	1.42	73.08
urban	32	6	4	1.45	0.05	0.03	60.52
urban	32	9	4	2.25	0.79	5.9	63.34
urban	48	12	12	1	0.77	1.51	90.56
urban	64	9	8	1.12	0.67	1.84	87.33
urban	64	18	4	4.26	1.64	4.11	70.18
urban	64	18	8	2.25	0.55	1.19	85.70
urban	64	18	16	1.12	0.45	0.76	94.04
urban	96	30	11	2.66	2.46	2.81	88.00
urban	128	32	4	8	2.33	8.34	73.13
urban	128	32	8	4	2.88	4.54	85.53
urban	128	32	32	1	2.05	2.94	94.64
suburban	32	9	4	2.25	1.34	8.42	73.19
suburban	35	14	11	1.25	3.34	3.60	93.99
suburban	55	15	6	2.50	1.39	7.69	78.95
suburban	70	30	8	3.75	3.93	4.83	90.91
suburban	77	25	25	1	1.91	4.31	98.84
suburban	77	25	7	3.50	1.47	3.99	91.49
rural	32	9	4	2.25	1.57	7.81	64.14
rural	44	10	10	1	3.92	5.12	92.52

TABLE V
PERFORMANCE OF GNN AND SOCP FOR DIFFERENT CELLULAR
SCENARIOS

Scenario					GNN SE Loss(%)	
<i>mor</i>	<i>M</i>	<i>K</i>	<i>K'</i>	BS	Median	Median No Attention
cellular-urban	32	9	3	4	2.30	85.27
cellular-urban	48	12	4	3	8.13	96.67
cellular-urban	48	12	3	4	3.46	64.10
cellular-urban	48	12	2	6	10.08	94.91
cellular-urban	64	9	3	4	4.78	41.73

REFERENCES

- [1] L. Salatin, H. Yang, S. Mishra, and C. S. Chen, "A GNN approach for cell-free massive MIMO," in *IEEE Global Communications Conference*, 2022, pp. 3053–3058.
- [2] H. He, X. Yu, J. Zhang, S. Song, and K. B. Letaief, "Cell-free massive MIMO for 6G wireless communication networks," *Journal of Communications and Information Networks*, vol. 6, no. 4, pp. 321–335, 2021.
- [3] Z. Liu and L. Dai, "A comparative study of downlink MIMO cellular networks with co-located and distributed base-station antennas," *IEEE Transactions on Wireless Communications*, vol. 13, no. 11, pp. 6259–6274, 2014.
- [4] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4247–4261, 2020.
- [5] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, *Fundamentals of Massive MIMO*. Cambridge University Press, 2016.
- [6] A. Papazafeiropoulos, P. Kourtessis, M. D. Renzo, S. Chatzinotas, and J. M. Senior, "Performance analysis of cell-free massive MIMO systems: A stochastic geometry approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3523–3537, 2020.
- [7] E. Nayebi, A. Ashikhmin, T. L. Marzetta, and H. Yang, "Cell-free massive MIMO systems," in *Asilomar Conference on Signals, Systems and Computers*, 2015, pp. 695–699.
- [8] E. Björnson and L. Sanguinetti, "Making cell-free massive MIMO competitive with mmse processing and centralized implementation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 77–90, 2020.
- [9] W. Zeng, Y. He, B. Li, and S. Wang, "Pilot assignment for cell free massive MIMO systems using a weighted graph framework," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6190–6194, 2021.
- [10] Y. Shen, J. Zhang, S. H. Song, and K. B. Letaief, "Graph neural networks for wireless communications: From theory to practice," *IEEE Transactions on Wireless Communications*, vol. 22, no. 5, pp. 3554–3569, 2023.
- [11] J. Guo and C. Yang, "Learning power allocation for multi-cell-multi-user systems with heterogeneous graph neural networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 884–897, 2022.
- [12] N. Keriven and G. Peyré, "Universal invariant and equivariant graph neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [13] L. Salatin and H. Yang, "Deep learning based power control for cell-free massive MIMO with MRT," in *IEEE Globecom*, 2021.
- [14] S. Buzzi and C. D'Andrea, "Cell-free massive MIMO: User-centric approach," *IEEE Wireless Communications Letters*, vol. 6, no. 6, pp. 706–709, 2017.
- [15] C. D'Andrea and E. G. Larsson, "User association in scalable cell-free massive MIMO systems," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 826–830.
- [16] E. Nayebi, A. Ashikhmin, T. L. Marzetta, H. Yang, and B. D. Rao, "Pre-coding and power optimization in cell-free massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4445–4459, 2017.
- [17] 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, 2017.
- [18] H. Yang and T. L. Marzetta, "Energy efficiency of massive MIMO: cell-free vs. cellular," in *IEEE 87th Veh. Technol. Conf.*, 2018.
- [19] E. Björnson, J. Hoydis, and L. Sanguinetti, *Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency*. Foundations and Trends in Signal Processing, 2017, vol. 11, no. 3–4.
- [20] H. Yang and T. L. Marzetta, "A macro cellular wireless network with uniformly high user throughputs," in *IEEE 80th Vehicular Technology Conference*, 2014, pp. 1–5.
- [21] G. Interdonato, E. Björnson, H. Q. Ngo, P. Frenger, and E. G. Larsson, "Ubiquitous cell-free massive MIMO communications," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, aug 2019.
- [22] H. Yang, "How to associate users with access points in a small cell network?" in *IEEE 91st Vehicular Technology Conference*, 2020, pp. 1–5.
- [23] M. Series, "Guidelines for evaluation of radio interface technologies for IMT-Advanced," *Report ITU M.2135-1*, 2009.
- [24] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," *arXiv preprint arXiv:2009.03509*, 2020.
- [25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [26] U. Challita, H. Ryden, and H. Tullberg, "When machine learning meets wireless cellular networks: Deployment, challenges, and applications," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 12–18, 2020.
- [27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [28] E. D. Andersen, C. Roos, and T. Terlaky, "On implementing a primal-dual interior-point method for conic quadratic optimization," *Mathematical Programming*, vol. 95, no. 2, pp. 249–277, 2003.
- [29] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [30] S. J. Wright, *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics, 1997. [Online]. Available: <https://epubs.siam.org/doi/abs/10.1137/1.9781611971453>
- [31] R. Monteiro and T. Tsuchiya, "Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions," *Mathematical Programming, Series B*, vol. 88, pp. 61–83, 06 2000.



Shashwat Mishra obtained his M.S. degree in 2021 on connection density maximization in NB-IoT networks from the Indian Institute of Technology Madras. He is pursuing his Ph.D., working jointly with Nokia Bell Labs France and INRIA Center, CITI Laboratory at INSA Lyon. His research interests include radio resource allocation, signal processing, system optimization, and geometric deep learning.



Lou Salaün obtained his Ph.D. degree in 2020, on resource allocation and optimization for the non-orthogonal multiple access from Institut Polytechnique de Paris, jointly working with Telecom Paris and Nokia Bell Labs. He received the IDIA Best Thesis Award from the Department of Computer Science, Data and Artificial Intelligence at Institut Polytechnique de Paris. He also received the PGM thesis first prize delivered by the Mathematical Foundation Jacques Hadamard. He is currently working as a research scientist at Nokia Bell Labs,

AI Research Lab. His research interests include algorithmics, graph theory, optimization, deep learning, telecommunications, resource allocation, and distributed computing.



Hong Yang (Senior Member, IEEE) received the PhD degree in applied mathematics from Princeton University, Princeton, NJ, USA. He worked in academia and for a start-up network technology company before joining Lucent Technologies and Alcatel-Lucent, where he was with the Wireless Design Center, the Systems Engineering Department, and the Bell Labs Research. He is currently a Research Scientist with the Department of Mathematics and Algorithms Research, Nokia Bell Labs, Murray Hill, NJ, USA, where he conducts research in com-

munications networks and AI applications. He has co-authored many research papers in wireless communications, applied mathematics, control theory, and financial economics. He co-invented many U.S. and international patents. He co-authored the book *Fundamentals of Massive MIMO* (Cambridge University Press, 2016).



Chung Shue Chen (Senior Member, IEEE) received the B.Eng., M.Phil., and Ph.D. degrees in information engineering from The Chinese University of Hong Kong (CUHK) in 1999, 2001, and 2005, respectively. He is a DMTS at Nokia Bell Labs. Prior to joining Bell Labs, he worked at INRIA, in the research group on Network Theory and Communications (TREC, INRIA-ENS). He was an Assistant Professor at CUHK. He was an ERCIM Fellow at the Norwegian University of Science and Technology (NTNU), Norway, and the National Centre for

Mathematics and Computer Science (CWI), The Netherlands. He worked with CNRS in Lorraine on Real-Time and Embedded Systems. His research interests include wireless communications and networking, optimization and algorithms, 5G/6G, IoT, machine learning, and intelligent systems. He has served as a TPC in international conferences, including IEEE ICC, Globecom, WCNC, PIMRC, VTC, CCNC, and WiOpt (TPC Vice Chair). He is an Editor of the Transactions on Emerging Telecommunications Technologies (ETT) and Associate Editor of the Springer Nature Telecommunication Systems. He is a Permanent Member of the Laboratory of Information, Networking and Communication Sciences (LINCS), Paris. He was a recipient of the Sir Edward Youde Memorial Fellowship and the ERCIM Fellowship.