




# Transformer-Based Power Optimization for Max-Min Fairness in Cell-Free Massive MIMO

Irched Chafaa , Giacomo Bacci , *Senior Member, IEEE*, Luca Sanguinetti , *Fellow, IEEE*

**Abstract**—Power allocation is an important task in wireless communication networks. Classical optimization algorithms and deep learning methods, while effective in small and static scenarios, become either computationally demanding or unsuitable for large and dynamic networks with varying user loads. This letter explores the potential of transformer-based deep learning models to address these challenges. We propose a transformer neural network to jointly predict optimal uplink and downlink power using only user and access point positions. The max-min fairness problem in cell-free massive MIMO systems is considered. Numerical results show that the trained model provides near-optimal performance and adapts to varying number of users and access points without retraining, additional processing, or updating its neural network architecture. This demonstrates the effectiveness of the proposed model in achieving robust and flexible power allocation for dynamic networks.

**Index Terms**—Cell-free massive MIMO, max-min fairness, power allocation, supervised learning, transformer neural network.

## I. INTRODUCTION

**P**OWER allocation is a crucial step in wireless networks to optimize the communications performance [1]. Depending on the nature of the optimization problem, power allocation can be performed via iterative methods [2] such as in the sum spectral efficiency (SE) problem [3] or using closed-form solution, as reported in [4] for the particular max-min fairness (MMF) problem. Iterative optimization algorithms require multiple iterations to find optimal powers, often failing to converge within channel coherence time for highly dynamic networks. They also impose high computational complexity due to numerous parameters that scale poorly with network size. In addition, if the configuration changes, as commonly happens in wireless networks, additional iterations are required. The closed-form solution [4] still requires substantial real-time (RT) channel information and is computationally expensive, due to matrix inversion and eigenvalue decomposition. Thus, while these methods provide theoretical benchmarks, they struggle with RT, dynamic, and large-scale networks.

Machine learning, including reinforcement and deep learning, has been explored to overcome iterative solution limitations [5]–[7]. While effective for fixed configurations and low-dimensional inputs, these methods are not suited to varying input sizes and dynamic systems, in the absence of retraining and architectural adjustments. In [7], the authors used a transformer-based [8] model for down-link (DL) power

allocation that adapts to varying numbers of user equipments (UEs) via unsupervised learning. However, the proposed method requires post-processing and padding, which increases computational overhead and limits scalability, especially with frequent changes in user load [9], [10]. In addition, it does not address varying numbers of access points (APs). This raises a key question: *How can we design a flexible learning model that can handle different UE loads and active number of APs while maintaining near-optimal performance?*

In this letter, we propose a supervised learning approach to train a transformer neural network (TNN) [8] that leverages UE and AP location information (e.g., spatial coordinates) to jointly predict up-link (UL) and DL powers, maximizing the minimum SE. We focus on the MMF problem in cell-free massive MIMO (mMIMO) systems [3], a key candidate for future 6G networks. Our main contributions are as follows. By leveraging the attention mechanism of the transformer, our model efficiently captures UE-AP relationships in parallel and adapts to the dynamic configuration of the network. Unlike previous works (e.g., [4], [6], [7]), we use only UE and AP coordinates (thus neglecting large-scale fading and channel statistics) while maintaining near-optimal performance. This minimal input reduces overhead, enables RT operation, and decouples power control from data detection, enhancing its applicability across different network layers. Our approach further addresses a key limitation of classical and learning-based power allocation methods, which require redesigning and retraining for variations in the number of UEs ( $K$ ) and APs ( $L$ ). By adapting the transformer with dynamic input and output layers, our solution seamlessly handles varying  $K$  and  $L$  without architectural changes, retraining, or additional processing such as padding. Trained on diverse configurations of  $K$  and  $L$ , the model generalizes effectively to unseen setups during inference. Finally, we validate the performance against the optimal closed-form max-min SE solution [4], achieving comparable results with three distinct advantages: increased flexibility, reduced computational complexity, and reduced input information.

## II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a cell-free mMIMO system, where  $K$  single-antenna UEs are served by  $L$  APs with  $N$  antennas each. The APs coordinate via a fronthaul network and a central processing unit (CPU) for joint processing and power allocation. The standard time division duplexing (TDD) protocol of cell-free mMIMO is used [3], where the  $\tau_c$  available channel uses are employed for: UL training phase ( $\tau_p$ ); down-link (DL) payload transmission ( $\tau_d$ ); and UL payload transmission ( $\tau_u$ ). Clearly,  $\tau_c \geq \tau_p + \tau_d + \tau_u$ . We consider a narrowband channel model and assume that the channel remains constant within a

This work was supported by the Italian Ministry of Education and Research (MUR) in the framework of the FoReLab Project (Department of Excellence) and in part by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE00000001 – Program “RESTART”, Structural Project 6GWINET, Cascade Call SPARKS).

I. Chafaa, G. Bacci, L. Sanguinetti are with the Dipartimento di Ingegneria dell’Informazione, University of Pisa, 56122 Pisa, Italy (e-mail: irched.chafaa@ing.unipi.it, {giacomo.bacci, luca.sanguinetti}@unipi.it).

coherence block. We denote the channel vector between the AP  $l$  and UE  $k$  with  $\mathbf{h}_{lk}$ , and model it as [3]:

$$\mathbf{h}_{lk} = \sqrt{\beta_{lk}} \mathbf{R}_{lk}^{1/2} \mathbf{g}_{lk} \quad (1)$$

where  $\beta_{lk}$  is the large-scale fading coefficient, accounting for path loss and shadowing,  $\mathbf{R}_{lk} \in \mathbb{C}^{N \times N}$  is the spatial correlation matrix at AP  $l$ , and  $\mathbf{g}_{lk} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_N)$  is an i.i.d. complex Gaussian vector representing the small-scale fading. We assume that the channels  $\{\mathbf{h}_{lk}; l = 1, \dots, L\}$  are independent and call  $\mathbf{h}_k = [\mathbf{h}_{1k}^T, \dots, \mathbf{h}_{Lk}^T]^T \in \mathbb{C}^{LN}$  the collective channel from all APs to UE  $k$ .

The CPU computes the estimate of  $\mathbf{h}_k$  on the basis of received pilot sequences transmitted during the training phase [3]. The minimum mean square error (MMSE) estimate is  $\hat{\mathbf{h}}_k = [\hat{\mathbf{h}}_{1k}^T, \dots, \hat{\mathbf{h}}_{Lk}^T]^T$  with [3]

$$\hat{\mathbf{h}}_{lk} = \mathbf{R}_{lk} \mathbf{Q}_{lk}^{-1} \left( \mathbf{h}_{lk} + \frac{1}{\tau_p \rho} \mathbf{n}_{lk} \right) \sim \mathcal{N}_C(\mathbf{0}_N, \Phi_{lk}) \quad (2)$$

where  $\rho$  is the UL pilot power of each UE,  $\mathbf{n}_{lk} \sim \mathcal{N}_C(\mathbf{0}_{LN}, \sigma^2 \mathbf{I}_{LN})$  is the thermal noise, and  $\Phi_{lk} = \mathbf{R}_{lk} \mathbf{Q}_{lk}^{-1} \mathbf{R}_{lk}$ , where  $\mathbf{Q}_{lk} = \mathbf{R}_{lk} + \frac{\sigma^2}{\tau_p \rho} \mathbf{I}_{LN}$ . Hence,  $\hat{\mathbf{h}}_k \sim \mathcal{N}_C(\mathbf{0}_{LN}, \Phi_k)$ , with  $\Phi_k = \text{diag}(\Phi_{1k}, \dots, \Phi_{Lk})$ . Note that the method proposed in this letter can be applied to other channel estimation schemes, such as the least-squares one [3].

#### A. Uplink and Downlink Transmissions

To detect the data of UE  $k$  in the UL, the CPU selects an arbitrary receive combining vector  $\mathbf{v}_k \in \mathbb{C}^{LN}$  based on all the collective channel estimates  $\{\hat{\mathbf{h}}_k; k = 1, \dots, K\}$ . An achievable SE of UE  $k$  is given by [3]:

$$\text{SE}_k^{\text{UL}} = \frac{\tau_u}{\tau_c} \log_2(1 + \text{SINR}_k^{\text{UL}}) \quad (3)$$

with the effective signal-to-interference-plus-noise ratio (SINR) defined as

$$\frac{p_k^{\text{UL}} |\mathbb{E}\{\mathbf{v}_k^H \mathbf{h}_k\}|^2}{\sum_{i=1}^K p_i^{\text{UL}} \mathbb{E}\{|\mathbf{v}_k^H \mathbf{h}_i|^2\} - p_k^{\text{UL}} |\mathbb{E}\{\mathbf{v}_k^H \mathbf{h}_k\}|^2 + \sigma^2 \mathbb{E}\{\|\mathbf{v}_k\|^2\}} \quad (4)$$

where  $p_k^{\text{UL}}$  is the UL transmit power of user  $k$ . The expectation is taken with respect to all sources of randomness. Although the bound in (3) is valid for any combining vector, we consider the MMSE combiner, given by [3]:

$$\mathbf{v}_k = \left( \sum_{k=1}^K p_k^{\text{UL}} \hat{\mathbf{h}}_k \hat{\mathbf{h}}_k^H + \mathbf{Z} \right)^{-1} \hat{\mathbf{h}}_k \quad (5)$$

where  $\mathbf{Z} = \sum_{k=1}^K p_k^{\text{UL}} (\mathbf{R}_k - \Phi_k) + \sigma^2 \mathbf{I}_{LN}$ .

In the DL, the CPU coordinates the APs to transmit signals to the UEs. Similarly to UL, an achievable SE of user  $k$  is obtained as:

$$\text{SE}_k^{\text{DL}} = \frac{\tau_d}{\tau_c} \log_2(1 + \text{SINR}_k^{\text{DL}}) \quad (6)$$

with the effective SINR defined as

$$\frac{p_k^{\text{DL}} |\mathbb{E}\{\mathbf{h}_k^H \mathbf{w}_k\}|^2}{\sum_{i=1}^K p_i^{\text{DL}} \mathbb{E}\{|\mathbf{h}_k^H \mathbf{w}_i|^2\} - p_k^{\text{DL}} |\mathbb{E}\{\mathbf{h}_k^H \mathbf{w}_k\}|^2 + \sigma^2} \quad (7)$$

where  $p_k^{\text{DL}}$  is the DL power used by the CPU to serve UE  $k$  and  $\mathbf{w}_k \in \mathbb{C}^{LN}$  is its associated unit-norm precoding vector. The MMSE precoder is used [3], which is given by  $\mathbf{w}_k = \frac{\mathbf{v}_k}{\|\mathbf{v}_k\|}$ .

#### B. Problem Formulation

We aim to develop a flexible, learning-based solution for optimal power allocation in networks with varying  $K$  and  $L$ , using minimal input information. In particular, we consider the max-min optimization problem, commonly used in cell-free mMIMO systems to ensure equal SE across UEs. In the UL, the problem takes the following form [3]:

$$\begin{aligned} & \max_{\{p_k^{\text{UL}} \geq 0\}} \min_k \text{SE}_k^{\text{UL}} \\ & \text{subject to } p_k^{\text{UL}} \leq P_{k,\max}^{\text{UL}} \quad \forall k \end{aligned} \quad (8)$$

where  $P_{k,\max}^{\text{UL}}$  is the maximum UL power for user  $k$ . Similarly, in the DL we have that:

$$\begin{aligned} & \max_{\{p_k^{\text{DL}} \geq 0\}} \min_k \text{SE}_k^{\text{DL}} \\ & \text{subject to } \sum_{k=1}^K p_k^{\text{DL}} \leq \sum_{l=1}^L P_{l,\max}^{\text{DL}} \end{aligned} \quad (9)$$

where  $P_{l,\max}^{\text{DL}}$  is the maximum power per AP. The constraint ensures that the total power allocated to all UEs does not exceed the total power budget across all APs.

Both optimization problems can be solved using the closed-form solution in [4], online iterative solvers [2], [3], or traditional DL models [5]–[7]. However, as discussed earlier, all these methods may be demanding for RT application in dynamic situations. To address this, we propose a flexible, data-driven alternative: a supervised learning framework with a trained TNN that jointly predicts the optimal powers, while handling varying numbers of UEs and APs.

### III. TRANSFORMER-BASED POWER ALLOCATION

Unlike other deep learning architectures [5]–[7], transformers can handle size-varying inputs and outputs without the need for additional processing steps [8]. The parallel processing capability of transformers further enhances their efficiency, making them ideal for RT applications. In addition, the self-attention mechanism enables them to capture complex relationships between UEs, APs, and channels, enabling more accurate and scalable power allocation predictions. By training on data spanning different numbers of UE and AP configurations, transformers can generalize well across a wide range of scenarios, eliminating the need to retrain or reconfigure the model's architecture for each new scenario. In the following, we explain in detail the basic parts of our proposed solution.

#### A. Training Data

For a given AP placement and path loss model, we generate multiple UE configurations to evaluate large-scale fading coefficients. Optimal UL and DL powers are then computed offline by solving the max-min SE optimization problems in (8) and (9), using the closed-form solution in [4]. The training dataset consists of input-output pairs  $(\mathbf{Z}, \mathbf{p}^*)$ , where  $\mathbf{Z} \in \mathbb{R}^{K \times (2L+2)}$  captures normalized  $x$ - $y$  coordinates of  $K$  UEs and  $L$  APs, and  $\mathbf{p}^* \in \mathbb{R}^{2K}$  contains the UL and DL

optimal power values. Min-max normalization [6] ensures consistent scaling for stable training. The dataset is split 80-20 into training and testing sets. To enhance generalization, the dataset covers diverse configurations, varying UE and AP counts, distributions, and channel realizations. Multiple samples per  $(K, L)$  combination enable learning robust power allocation across cell-free mMIMO scenarios.

### B. Model Architecture

The proposed TNN model predicts UL and DL powers by capturing relationships between input features. It comprises three key components: a *dynamic* input layer, a multi-layer transformer encoder, and an output layer, enabling adaptability to varying UEs and APs. Fig. 1 illustrates the processing steps.

- 1) *Dynamic input layer*: For each batch of size  $B$ , the input tensor  $\mathbf{X} \in \mathbb{R}^{B \times K \times (2L+2)}$  encodes spatial information for  $K$  UEs and  $L$  APs, including only their  $x$ - $y$  coordinates. A fully connected input layer maps features to a  $d_{\text{mod}}$ -dimensional space using [8], [11]:

$$\mathbf{H} = \text{ReLU}(\mathbf{X}\mathbf{W}_{\text{input}} + \mathbf{b}_{\text{input}}) \quad (10)$$

where ReLU is an activation function [12],  $\mathbf{W}_{\text{input}} \in \mathbb{R}^{(2L+2) \times d_{\text{mod}}}$  and  $\mathbf{b}_{\text{input}} \in \mathbb{R}^{d_{\text{mod}}}$  are learnable parameters. The resulting tensor  $\mathbf{H} \in \mathbb{R}^{B \times K \times d_{\text{mod}}}$  is fed into the transformer encoder. Unlike previous methods [5]–[7], our model dynamically adjusts the input tensor size  $\mathbf{X}$  based on  $K$  and  $L$  in each batch. This enables seamless handling of varying  $K$  and  $L$  during both training and inference, eliminating the need to add more processing steps. Note that, while the model does not explicitly receive channel information as an input, its effect is accounted for indirectly through the training labels  $\mathbf{p}^*$ , which are computed offline using a channel-aware solution for each UE/AP position.

- 2) *Multi-layer transformer encoder*: The core of the model is a transformer encoder that processes the tensor  $\mathbf{H}$ . It comprises  $M$  layers, each featuring a multi-head attention (MHA) mechanism and a feed-forward network (FFN) [8]. The self-attention mechanism effectively captures dependencies by computing attention scores between all input element pairs:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_K}}\right) \mathbf{V} \quad (11)$$

where Softmax is an activation function [12],  $\mathbf{Q}$ ,  $\mathbf{K}$ , and  $\mathbf{V}$  are query, key, and value matrices derived from  $\mathbf{H}$  with  $D_K$  being a dimension of the key matrix [8], [11]. After that, each UE's feature is processed through the FFN:

$$\text{FFN}(\mathbf{h}) = \text{ReLU}(\mathbf{h}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \quad (12)$$

where  $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$  are trainable parameters. As a result, the transformer encoder output,  $\mathbf{H}_{\text{out}} \in \mathbb{R}^{B \times K \times d_{\text{mod}}}$ , encodes the learned relationships between all UEs and APs, effectively capturing their interactions.

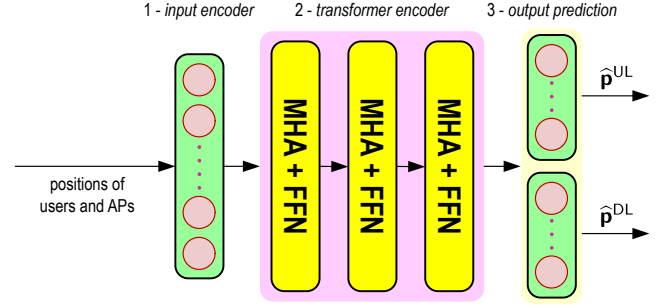


Fig. 1: Architecture diagram of the proposed transformer-based model to predict jointly UL and DL powers leveraging spatial information at the input.

- 3) *Output layer*: To predict the UL and DL powers, the output of the transformer encoder  $\mathbf{H}_{\text{out}}$  is passed through two separate fully-connected layers yielding:

$$\hat{p}_k^{\text{UL}} = \text{Sigmoid}(\mathbf{H}_{\text{out}} \mathbf{W}_{\text{UL}} + \mathbf{b}_{\text{UL}}) \cdot P_{k, \text{max}}^{\text{UL}} \quad (13)$$

$$\hat{p}_k^{\text{DL}} = \text{Sigmoid}(\mathbf{H}_{\text{out}} \mathbf{W}_{\text{DL}} + \mathbf{b}_{\text{DL}}) \cdot \frac{\sum_{l=1}^L P_{l, \text{max}}^{\text{DL}}}{\sum_{k=1}^K \hat{p}_k^{\text{DL}}} \quad (14)$$

where  $\mathbf{W}_{\text{UL}}, \mathbf{W}_{\text{DL}}, \mathbf{b}_{\text{UL}}, \mathbf{b}_{\text{DL}}$  are trainable parameters, and Sigmoid is an activation function ensuring an output in the range  $[0, 1]$  [12]. The predicted UL and DL powers for each user are concatenated to form the final output tensor of powers  $\hat{\mathbf{p}} \in \mathbb{R}^{B \times K \times 2}$  for all UEs in the batch.

The model is trained to minimize the mean square error (MSE) between the predicted powers  $\hat{\mathbf{p}}$  and the optimal powers  $\mathbf{p}^*$  from offline optimization. By doing so, the model implicitly encodes the effects of the channel propagation environment during training, and learns to approximate the function, mapping the positions of UEs and APs to optimal powers that maximize the minimum SE:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{B} \sum_{i=1}^B \|\mathbf{p}_i^* - \hat{\mathbf{p}}_i\|^2. \quad (15)$$

### C. Computational Complexity

The computational complexity of the proposed model differs between the training and inference phases. During training, the model performs both a forward and backward pass, with the main cost arising from the self-attention mechanism in the transformer encoder. The overall training complexity [8], [11] is in the order of:

$$\mathcal{O}(2MBKd_{\text{mod}}(d_{\text{mod}} + K) + 2BK(2L + 2)d_{\text{mod}}). \quad (16)$$

In contrast, inference involves only a forward pass, which is faster and more efficient since it lacks back-propagation and gradient updates and only uses  $x$ - $y$  coordinates as input. The inference complexity per sample [8], [11] is in the order of:

$$\mathcal{O}(M(Kd_{\text{mod}}^2 + K^2d_{\text{mod}}) + K(2L + 2)d_{\text{mod}}). \quad (17)$$

For high numbers of UEs ( $K$ ), the complexity for predicting UL and DL powers jointly is dominated by  $\mathcal{O}(Md_{\text{mod}}K^2)$ . In contrast, the closed-form solution in [4] has a higher

complexity of  $\mathcal{O}(K^3)$  for each power prediction (UL or DL separately). Iterative optimization methods such as [2], scales with  $\mathcal{O}(QLK^2)$ , where  $Q$  represents the number of iterations required to converge, which can exceed hundreds [7]. Moreover, the optimal powers are computed based on a substantial amount of information, including channel statistics (such as covariance matrices, channel estimates, and large-scale fading) as well as combining and precoding vectors. These elements must be updated regularly according to the current network configuration and UE load. As a result, the optimal powers must be recalculated each time the parameters  $K$  or  $L$  change. All these points establish the computational advantage of our method compared to existing approaches.

To further demonstrate the efficiency of the proposed model, we compare the computational time required to predict both UL and DL powers with  $K = 40$  and  $L = 16$ : using the same computational facilities, the proposed method has a runtime of 18.5 ms, around 1700 times faster than the one required by the closed-form solution (31.6 s).

#### IV. NUMERICAL RESULTS

In this section, we present numerical results to illustrate the performance of the proposed solution for a cell-free mMIMO system, as described in Sect. II.

##### A. Cell-free mMIMO parameters

We consider a network with a coverage area of  $500\text{ m} \times 500\text{ m}$ , with  $N = 4$  antennas per AP. The APs are uniformly deployed within the squared coverage area. The maximum UL transmit power for each user is 100 mW, whereas the maximum DL transmit power for each AP is 200 mW. We assume  $\tau_c = 200$  and set  $\tau_p = K$ ,  $\tau_u = \lfloor (\tau_c - \tau_p) / 2 \rfloor$  and  $\tau_d = \tau_c - \tau_p - \tau_u$ . Large-scale fading coefficients are computed following the 3GPP path-loss model adopted in [4, Sect. III-D] for a 2-GHz carrier frequency, a pathloss exponent of 3.67, a UE-AP height difference of 10 m and a shadow fading  $F_{kl} \sim \mathcal{CN}(0, \alpha^2)$ , with  $\alpha^2 = 4\text{ dB}$ . The shadow fading terms are spatially correlated as in [4, Sect. III-D] to account for the fact that closely located UEs experience similar shadow fading effects. The noise power is  $\sigma^2 = -94\text{ dBm}$  [4] with a noise figure  $\eta = 7\text{ dB}$  and a bandwidth  $B = 20\text{ MHz}$ .

A total number of 800 training samples are generated for each value of  $K \in \{2, 4, 6, 8, 10\}$  and  $L \in \{9, 16\}$ . Each sample consists of UE and AP positions, and optimal powers. Additionally, a testing dataset of 200 samples for  $K = 2, 3, \dots, 100$  and  $L = 4, 5, \dots, 49$  is also generated. By extending the testing range beyond the training values for both UEs and APs, the model's ability to generalize to new dynamic network configurations can be effectively assessed.

##### B. Learning model parameters

The transformer architecture includes  $M = 2$  encoder layers with four attention heads and a model dimension  $d_{\text{mod}} = 32$ . The model's training incorporates a dropout rate of 0.1 to reduce overfitting by randomly deactivating connections during the training. The learning rate is set to 0.001 for consistent and controlled optimization using the

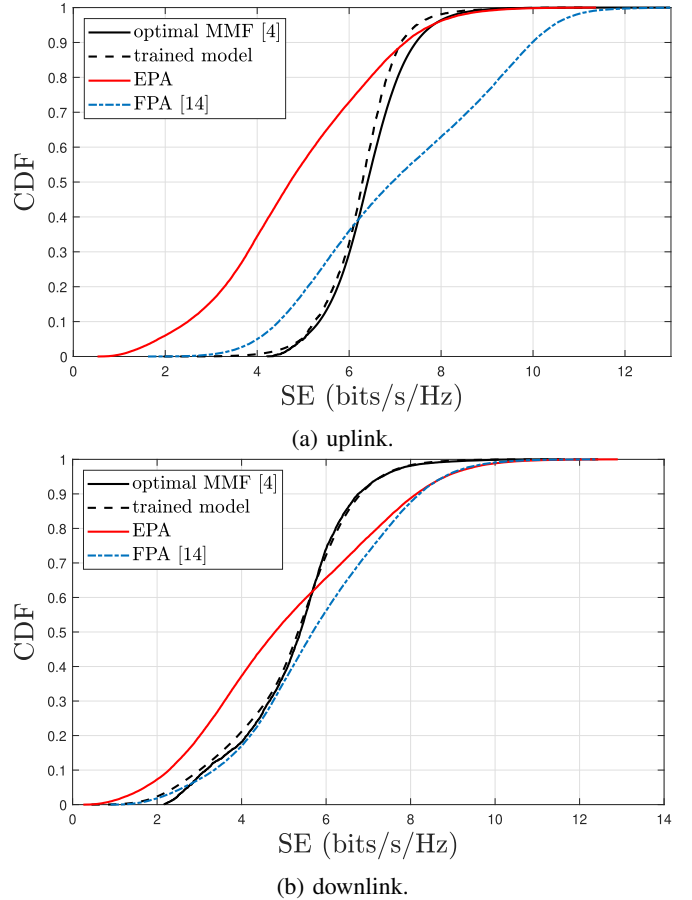


Fig. 2: CDF of average per-UE SE in UL and DL on the test set of different values of  $K$  and  $L$ . The trained model performs close to the optimal one and generalizes well on different network configurations using the same architecture.

well-known adaptive moment estimation with weight decay (AdamW) optimizer [13], which combines fast convergence with improved regularization. The training is performed with 10 epochs for each chosen value of  $K$  and  $L$ , with a batch size of 32 samples.

##### C. Performance evaluation

We plot the cumulative distribution function (CDF) of the average per-UE SE obtained by evaluating our trained model on test data (unseen during training) across diverse configurations of  $2 \leq K \leq 100$  and  $4 \leq L \leq 36$ , including values beyond the training range. The UL and DL results are shown in Fig. 2a and Fig. 2b, respectively, and comparisons are made with *i*) the optimal closed-form solution from [4], *ii*) equal power allocation (EPA), and *iii*) fractional power allocation (FPA) [14]. The results illustrate that the SE achieved by the trained model closely follows the optimal solution, highlighting its ability to deliver near-optimal performance for both UL and DL. Importantly, the proposed model achieves smaller difference between lower and upper tails of the CDF, indicating better fairness among UEs, unlike EPA and FPA CDFs (despite occasional intersections with the optimal CDF). These results show not only the ability to predict near-optimal



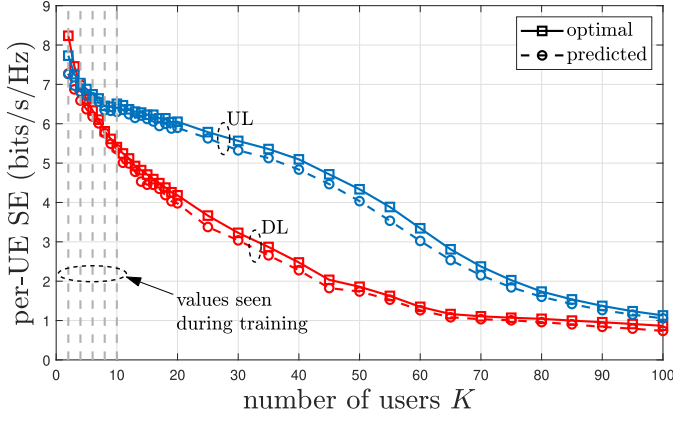


Fig. 3: Average per-UE SE for different  $K$  values. The trained model adapts well to unseen values of  $K$  without retraining or reconfiguring the model's architecture.

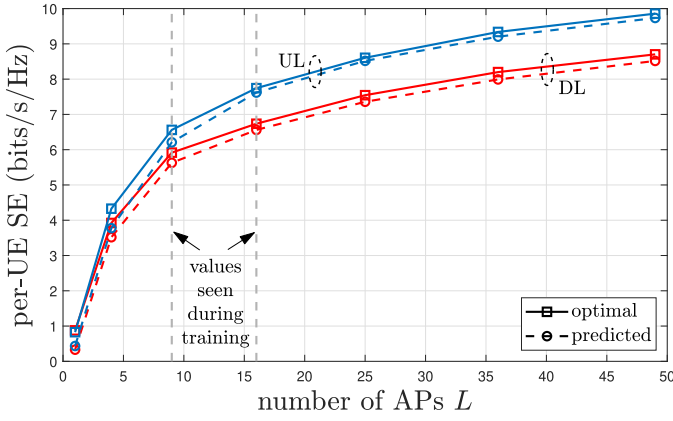


Fig. 4: Average per-UE SE for different numbers of APs. The model handles varying numbers of APs without a significant performance degradation.

UL and DL powers, but also a flexibility for different UE load and AP configurations (using the same model without redesigning its architecture or retraining).

Fig. 3 compares the per-UE SE, achieved by the optimal and predicted powers, for different values of  $K$  and  $L = 16$  of the test set. First, we notice that the SE decreases when  $K$  increases, as expected, but gradually, showing that the system handles the increasing user load efficiently. Moreover, Fig. 3 shows that the trained model achieves near-optimal performance even for unseen and higher numbers of UEs  $K > 10$ . This is achieved using just UE and AP positions as inputs and without retraining the model, redesigning its architecture, or employing additional data-processing.

In Fig. 4, the trained model is evaluated on the test data with  $K = 10$  for varying numbers of APs  $L$  as typically happens in *user-centric* cell-free mMIMO. As expected, the per-UE SE increases as  $L$  increases. Furthermore, the SE obtained with the trained model follows the optimal one, illustrating a good generalization across different AP counts for both UL and DL. Again, this is achieved without changing the model's architecture or employing additional processing to the data.

## V. CONCLUSIONS

This letter presented a supervised learning approach to train a TNN for jointly predicting UL and DL powers in a wireless network, using only the spatial coordinates of UEs and APs. As a case study, the max-min problem in a cell-free mMIMO system was considered. The proposed model efficiently handles varying numbers of UEs and APs without requiring retraining or adjustments to the neural network, leveraging the transformer's architecture and dynamic training on diverse configurations. Additionally, the model is designed to address both UL and DL power allocation through dedicated output layers, which leads to more flexibility. The model is trained, using labels that inherently reflects channel conditions, to learn a mapping that mimics the behavior of the closed-form solution and provide fairness among users. Numerical results showed that our model achieves near-optimal performance across varying system parameters, highlighting its flexibility for dynamic power allocation. While our model has a computation advantage compared to current methods, it still grows quadratically with the number of users due to the attention mechanism in transformers. Thus, to reduce complexity for very large systems, future work will explore efficient transformer variants using sparse or local attention [11], which can potentially reduce the complexity from  $\mathcal{O}(MK^2d_{\text{mod}})$  to linear or logarithmic in  $K$ .

## REFERENCES

- [1] L. Tan, *Resource Allocation and Performance Optimization in Communication Networks and the Internet*. Boca Raton, FL: CRC press, 2017.
- [2] M. Farooq, H. Q. Ngo, and L.-N. Tran, "Accelerated projected gradient method for the optimization of cell-free massive MIMO downlink," in *Proc. Intl. Symp. Personal, Indoor and Mobile Radio Commun. (PIMRC)*, London, UK, 2020.
- [3] Ö. T. Demir, E. Björnson, and L. Sanguinetti, "Foundations of user-centric cell-free massive MIMO," *Foundations and Trends® in Signal Processing*, vol. 14, no. 3-4, pp. 162–472, 2021.
- [4] L. Miretti, R. L. G. Cavalcante, S. Stańczak, M. Schubert, R. Böhnke, and W. Xu, "Closed-form max-min power control for some cellular and cell-free massive MIMO networks," in *Proc. IEEE Veh. Technol. Conf.*, Helsinki, Finland, 2022.
- [5] D. Kim, H. Jung, and I.-H. Lee, "A survey on deep learning-based resource allocation schemes," in *Proc. Intl. Conf. Information and Commun. Technol. Convergence (ICTC)*, Jeju Island, South Korea, 2023.
- [6] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 2018.
- [7] A. K. Kocharlakota, S. A. Vorobyov, and R. W. Heath Jr, "Pilot contamination aware transformer for downlink power control in cell-free massive MIMO networks," *arXiv preprint arXiv:2411.19020*, 2024.
- [8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Conf. Neural Inf. Process. Systems (NIPS)*, Long Beach, CA, USA, 2017.
- [9] M. Dwarampudi and N. Reddy, "Effects of padding on LSTMs and CNNs," *arXiv preprint arXiv:1903.07288*, 2019.
- [10] F. Alrasheedi, X. Zhong, and P.-C. Huang, "Padding module: Learning the padding in deep neural networks," *IEEE Access*, vol. 11, pp. 7348–7357, 2023.
- [11] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI open*, vol. 3, pp. 111–132, 2022.
- [12] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, "A review of activation function for artificial neural network," in *Proc. IEEE World Symp. Applied Machine Intell. & Informatics (SAMII)*, Herlany, Slovakia, 2020.
- [13] P. Zhou, X. Xie, Z. Lin, and S. Yan, "Towards understanding convergence and generalization of AdamW," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46d, no. 9, pp. 6486–6493, 2024.
- [14] R. Nikbakht and A. Lozano, "Uplink fractional power control for cell-free wireless networks," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, Shanghai, China, 2019.