Alma Mater Studiorum - Univesità di Bologna
Campus di Cesena

DIPARTIMENTO DI
INGEGNERIA DELL'ENERGIA ELETTRICA E DELL'INFORMAZIONE
"GUGLIELMO MARCONI"

Corso di Laurea Magistrale in Ingegneria Elettronica e Telecomunicazioni per l'Energia

DOWNLINK POWER ALLOCATION FOR CELL-FREE MASSIVE MIMO SYSTEMS USING UNSUPERVISED LEARNING

Elaborato in
Sistemi di Telecomunicazioni LM

Relatore:
Prof. Sergio Callegari

Correlatore:
Prof. Davide Dardari

Presentata da:
Mattia Fabiani

Anno Accademico 2022/2023

Contents

Chapter 1

Introduction

1.1 Background and Motivations

The rapid diffusion of wireless devices and the ever-increasing demand for high data rates have driven the evolution of cellular communication systems. With an eye on the upcoming 6G era, where demands for even better connectivity and novel wireless applications are rising, cell-free massive multiple-input multiple-output (CF-mMIMO) systems have emerged as a promising solution. Firstly, the advent of massive MIMO technology has engendered significant improvements in wireless communication systems, predominantly in cellular networks [1]. However, conventional cellular deployments encounter several constraints regarding coverage, capacity, and interference management. CF-mMIMO [2, 3] is a novel paradigm that aims to surmount these limitations by harnessing plenty of geographically distributed access points (APs) that jointly serve multiple user equipments (UEs) within a given coverage area, thereby obviating the need for artificial cell boundaries [4–6]. This architecture offers several advantages over traditional cellular networks. Primarily, the cooperation of a larger number of APs enhances spatial diversity and augments the wireless channel quality. Moreover, surrounding UEs by a large number of low-cost APs constitute a distributed antenna array, thus providing near uniform, superior service quality within the coverage area. Owing to its capability to suppress multi-user interference and ensure ubiquitous connectivity, CF-mMIMO has been recognized as a pivotal technology for beyond 5G (B5G) networks [7].

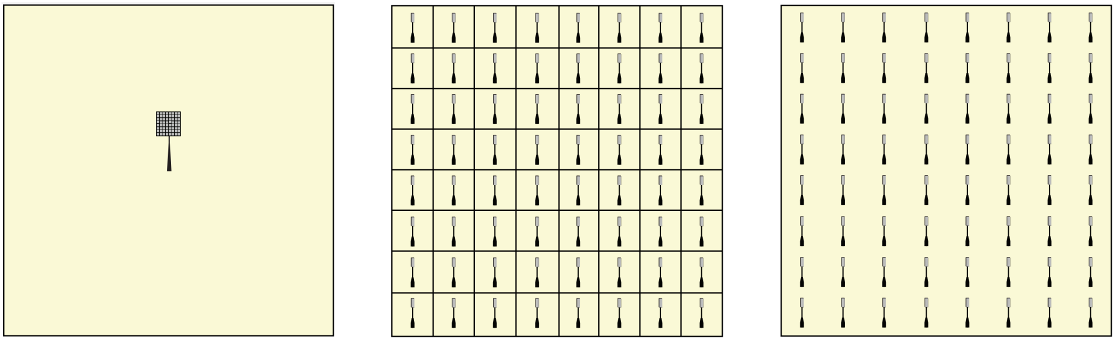
Nonetheless, the deployment of CF-mMIMO poses several challenges that must be addressed to fully harness its potential benefits. Signal processing, along with channel estimation, precoding, and resource allocation tasks, is typically performed by the central processing unit (CPU). Nevertheless, the information exchange between the distributed APs entails a significant communication overhead and computational complexity and necessitates efficient network management to achieve the real-time performance enhancement offered by the CF-mMIMO concept. Since the UEs are jointly served on the same resource blocks, the power allocation plays an important role in suppressing multi-user interference and optimizing the overall network performance. Current cellular networks, indeed, suffer from inter-cell interference, especially at the border of each cell, where a given user resides far from the serving base station.

In this work, the downlink (DL) power allocation problem is addressed in an unsupervised learning fashion. The reason behind the choice of learning-based resource allocation, rather than optimization-based [5, 7, 8], is related to the high computational complexity associated with iterative algorithms which may preclude real-time performance. On the other hand, in the learning-based approach, the main computational cost is related to the neural network training and training labels generations if supervised learning is adopted as in [9].

In this work, we solely rely on the large-scale fading (LSF) coefficients as input rather than the hard-to-obtain exact UEs locations [10]. The same network topology as [9] is considered, with sum spectral efficiency (sum-SE) optimization objective and two different precoding schemes: maximum ratio (MR) and regularized zero-forcing (RZF). There are two main ways of implementing power control: centralized, where the optimization is entirely performed by the CPU, and distributed, in which the APs use local environment information in order to optimize the power control coefficients. In this work, both methods are proposed, however, the distributed model allows for a more scalable solution when the number of APs and UEs increases, albeit the global optimum is less likely to be found. In contrast with [11], a different kind of loss function that takes into consideration the per-AP DL power constraint and the sum-SE objective is adopted relying on multi-antenna AP, thus achieving higher network-wide performance. Note that DL power control is more complicated than uplink (UL), due to a higher number of parameters to be optimized. Indeed, in DL each AP has to allocate the power to all the UEs in the network, whereas in UL a single UE has only one power coefficient to be optimized. It is worth mentioning that throughout the following sections, the proposed methodologies will be explained, ensuring that they are presented in a way that is accessible to individuals who may not have a deep background in the field of machine learning

1.2 CF-mMIMO

In the last decades, the use of data services has grown exponentially as more and more users and machines need to be network-linked. In the future, the existing infrastructure may not be able to serve the increasing demands. The CF-mMIMO



(a) Massive MIMO cell (b) Small cells (c) Cell-free

Figure 1.1: Two cellular networks (a), (b) in contrast to a cell-free setup [6].

concept is an emerging novel wireless communication technology that has brought significant attention in the literature, as one of the possible solutions to overcome the current deployment limitations. The main idea is to rebuild the existing infrastructure, delete the artificial cell boundaries, and make each UE surrounded and jointly served by a higher number of low-cost geographically dislocated APs. In this way, ubiquitous connectivity can be provided within a given coverage area.

1.2.1 Cellular and Cell-free Networks

In the '80s, the first cellular networks were introduced with the primary purpose of enabling wireless connectivity to a wide range of services over a large geographic region. Current mobile networks are built as cellular networks, where each cell uses a fraction of available resources. Each AP resides in the center of a cell and serves UEs located inside of it, where the strongest SNR is sensed. In the 5G scenario, the massive multiple-input multiple-output (MIMO) technology represents a key feature, where a given AP is equipped with a high number of active low-gain antennas each controlled by different transceiver chains. Note that in this kind of system, each AP operates individually to serve the UEs within the cell. Massive MIMO allows for more efficient use of spatial diversity, i.e. by exploiting the very high beamforming gain of the antenna array, and the received SNR is proportional to the number of antennas.

A comparison between a massive MIMO cell, small-cells, and cell-free network is shown in Fig. 1.1. In Fig. 1.1a a massive MIMO cell is depicted, in which a high number of antennas compose the AP serving the UEs within the coverage area. However, the network may be affected by large path loss (PL) fluctuations inside the cell, depending on the UE's location relative to the AP. Fig. 1.1b shows a small-cell setup, where eventually the same number of antennas as in Fig. 1.1a are geographically dislocated to create cell densification. In this scenario,

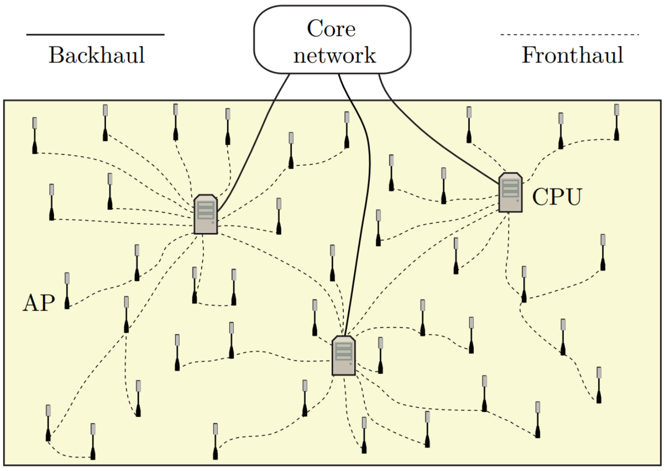


Figure 1.2: Cell-free network illustration. Many geographically distributed APs are connected to CPUs via fronthaul links and jointly serve all the UEs within the coverage area [6].

each AP is equipped with a single antenna, and a UE is served only by one of them. This architecture poses several challenges in terms of inter-cell interference, UE mobility, and performance prediction, as well as security issues. A cell-free network is proposed in Fig. 1.1c. Here, the same number of APs as in the small-cell setup is adopted, but the APs are intended to jointly serve multiple UEs in the coverage area. The term *cell-free* means that no artificial cell boundaries exist as in conventional cellular networks, and the high number of distributed APs allows for lower signal-to-noise ratio (SNR) variations compared to a massive MIMO cell. Note that in traditional cellular systems, the problem of interference persists, especially in the regions closer to the cell boundaries. By making use of a cell-free setup, many of the interference issues can be solved.

1.2.2 CF-mMIMO Topology

The CF-mMIMO definition is not related to a site-specific deployment, except that the topology should be distributed in space. For instance, Fig. 1.2 shows a possible cell-free network topology. As can be seen, very large groups of APs are connected to a certain central processing unit (CPU) via fronthaul links, and CPUs are connected to a core network via backhaul links. Fronthaul links can be wired or wireless, and are responsible for the cooperation of the APs, while the backhaul links' purpose is to receive and send internet data and other sources. Regarding the fronthaul links, there can be different ways of implementing such connections, i.e. star or sequential topology. In the former, each AP is connected to the CPU with a dedicated fronthaul link, while in a sequential topology a general fronthaul link may be shared by multiple APs. Note that the star connections could enable major performance, at the expense of a higher deployment cost due to the number of cables if the connection is wired. The illustrated topology in Fig. 1.2 is just a reference, and it refers to fronthaul links in which some APs reach the proper CPU via neighbors, eventually in a sequential fashion, i.e. radio stripes [12]. In [12], a cell-free architecture is composed of flexible radio stripes, in which the deployment cost is cut down thanks to the sequential connection topology regarding the AP-to-CPU links. Furthermore, sensors and other devices could be embedded as smart additional features, while being almost invisible as the stripes are attached to existing construction elements.

Power Control

In order to handle interference among the UEs in a CF-mMIMO network, power allocation is a crucial aspect that has to be addressed. Specifically, the power control is performed once every LSF realization, which varies on a large time scale, caused by moving objects like vehicles or other obstacles in the environment. Depending on the chosen scheme, the power allocation can be performed by the CPU (centralized approach), or directly by the APs utilizing local channel information (distributed scheme). These two methods will be further discussed in detail in this thesis.

1.2.3 Benefits and Challenges of CF-mMIMO

This section describes an overview of the main challenges and benefits of a CF-mMIMO system. One of the primary benefits of cell-free is improved coverage. By using many small APs instead of a few large ones, the signal can be distributed more evenly throughout the coverage area. This results in better signal strength and fewer dead zones. Another advantage is reduced interference. In traditional cellular systems, UEs in adjacent cells can interfere with each other, resulting in reduced signal-to-interference-plus-noise ratio (SINR). With CF-mMIMO, since all the resources are shared throughout the coverage region, the interference can be dynamically adjusted by leveraging a proper power allocation strategy at the APs. For instance [12], Fig. 1.3 shows the SE comparison between a cellular massive MIMO and a CF-mMIMO using 9 APs in a $1000\text{ m} \times 1000\text{ m}$ area. On the left side, a general UE can experience a strong interference at the cell boundaries, which causes a low performance. In contrast, the cell-free network, on the right side, provides almost uniform connectivity as the interference can be avoided thanks to the cooperation between APs, and the performance is only limited by the path loss, thus providing non-zero SE throughout all the possible UE locations.

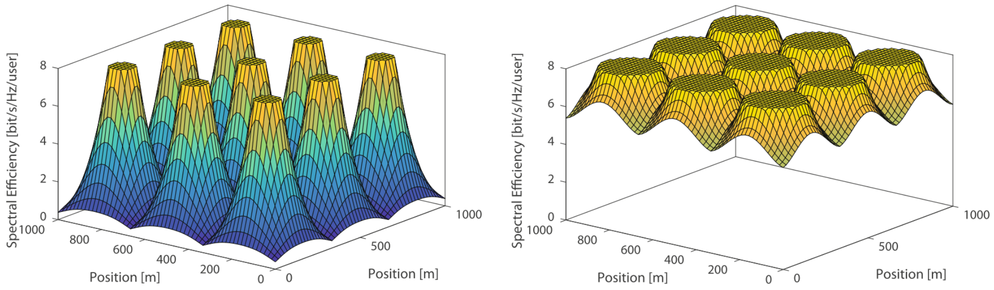


Figure 1.3: Data coverage. Left: cellular network. Right: CF-mMIMO network. SE is achieved by UEs at different locations in an area covered by nine APs that are deployed on a regular grid. Note that 8 bit/s/Hz was selected as the maximal SE, corresponding to uncoded 256-QAM.

Two main benefits inherited by massive MIMO are *channel hardening* and *favorable propagation.* Both conditions become more and more valid as the number of antennas at the transmitter grows up to infinity. Channel hardening refers to the effective channel of a given UE to become asymptotically deterministic, i.e. the random channel realizations are close to the mean value. On the other hand, in a multi-user environment, favorable propagation is experienced when the channels of different UEs are asymptotically orthogonal, thus mitigating the interference between UEs.

However, there are many challenges associated with the post-cellular network, such as the increased complexity of the system, synchronization, limited capacity of fronthaul and backhaul connections, and power allocation. Since all the resources are shared in the coverage area, the latter plays an important role that must be handled. In this thesis, we focus on a learning-based solution for the DL power allocation problem. In particular, centralized and distributed power control will be tackled using an unsupervised deep learning technique, as optimization-based solutions for the problem may not allow for a real-time application due to higher latency.

1.3 Thesis Outline

The thesis is organized as follows. Chapter 2 describes the system model, including pilot assignment, channel estimation, DL data transmission, the sum-SE maximization problem, and the principal metrics employed to evaluate the models' performance. In Chapter 3 the proposed centralized and distributed deep neural network (DNN) structures are presented, explaining how an unsupervised learning approach can be exploited for the power allocation problem. Chapter 4 summarizes the simulation results, obtained through different realizations of UEs, where a comparison with conventional optimization-based algorithms and supervised learning solutions against the proposed unsupervised learning approach takes place.

Chapter 2

System Model and Problem Formulation

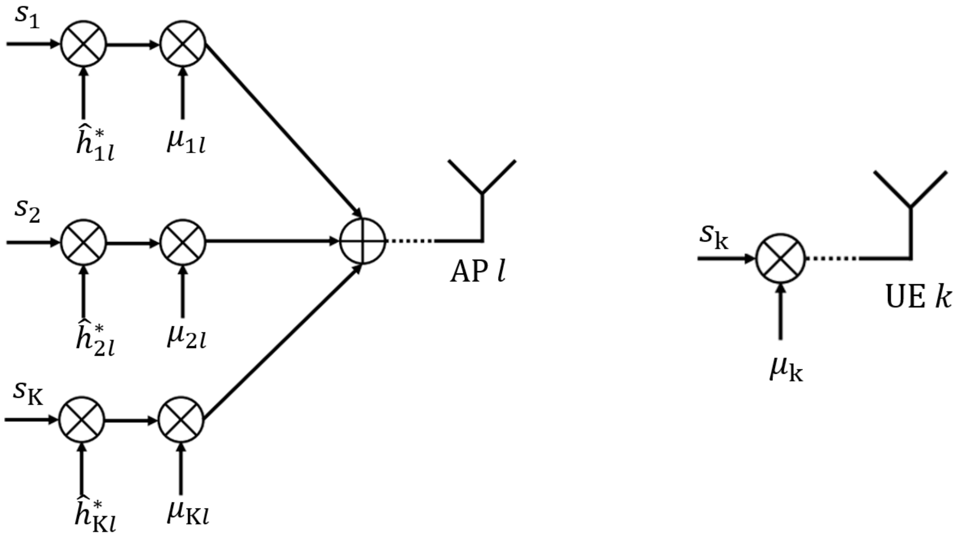
2.1 System Model

The wireless channel is a complex medium to model, due to its time variations caused by the movement of the transmitter, receiver, and eventually obstacles in between. The effects of multipath and fading are significant and may reduce the system performance if left unaddressed. That is, even small movements of the receiver, in the order of a fraction of wavelength (i.e. centimeters), could lead to substantial degradation of the channel. Nevertheless, if we examine the channel over a small time scale, it can be approximated as constant, acting as a linear time-invariant (LTI) filter. Let us suppose a multiple-input single-output (MISO) system, where the receiver is equipped with a single antenna, and the number of antennas at the transmitter is denoted as N_t. The channel model is defined as follows:

$$y = \boldsymbol{h}\boldsymbol{x} + n \tag{2.1}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_{N_t}]^T \in \mathbb{C}^{N_t \times 1}$ is the transmitted array of symbols, $y \in \mathbb{C}$ denotes the received symbol, and $\boldsymbol{h} = [h_1, \ldots, h_i, \ldots, h_{N_t}] \in \mathbb{C}^{1 \times N_t}$ is the channel vector, where each element represents the channel between the transmit antenna i and the receive antenna. $n \sim \mathcal{CN}(\mathbf{0}, \sigma^2)$ represents the additive complex Gaussian noise with zero mean and variance σ^2.

In Fig. 2.1 is showcased the signal processing performed by the single-antenna l^{th} AP, i.e. AP$_l$, and k^{th} UE, UE$_k$, considering the maximum ratio (MR) precoding scheme. It can be seen that in the DL case, Fig. 2.1a, the intended symbol s_k is firstly weighted by the conjugate of the estimated channel $\hat{h}^*_{kl}$ between UE$_k$ and AP$_l$, and then multiplied by the power allocation coefficient μ_{kl}. Subsequently, the processed signal is combined with the signals intended for all the other UEs



(a) Downlink.

(b) Uplink.

Figure 2.1: Signal processing using MR precoding/combining at the l^{th} AP (a) and k^{th} UE (b).

and transmitted. In the UL case, Fig. 2.1b, the UE$_k$ necessitates weighting the desired signal with only a single power coefficient μ_k.

In this thesis, the same system model of [9] is adopted, which refers to a CF-mMIMO system with K single-antenna UEs (i.e. $N_r = 1$) uniformly distributed in a given service area and served by L APs, each equipped with $N_t = N$ antennas, as depicted in Fig. 2.4. A block fading channel divided into time-frequency coherence blocks is considered. The structure of a generic coherence block is depicted in Fig. 2.2, consisting of a number of subcarriers and time samples over which the channel response can be approximated as constant and flat-fading. It comprises τ_c symbols, of which τ_p are reserved for pilot assignments and channel estimation purposes and τ_d for DL data transmission, i.e., $\tau_c = \tau_p + \tau_d$. Both orthogonal and non-orthogonal pilots are considered, i.e., $\tau_p = K$ and $\tau_p < K$ respectively. The channel between the UE$_k$ and AP$_l$ is denoted by $\mathbf{h}_{kl} \in \mathbb{C}^{N\times 1}$ and is modeled by correlated Rayleigh fading as $\mathbf{h}_{kl} \sim \mathcal{CN}(\mathbf{0}, \mathbf{R}_{kl})$, where $\mathbf{R}_{kl} \in \mathbb{C}^{N\times N}$ denotes the spatial correlation matrix. Finally, the average channel gain β_{kl} from a given AP$_l$ to UE$_k$ is determined by the average trace $\beta_{kl} = \frac{1}{N}\text{tr}(\mathbf{R}_{kl})$.

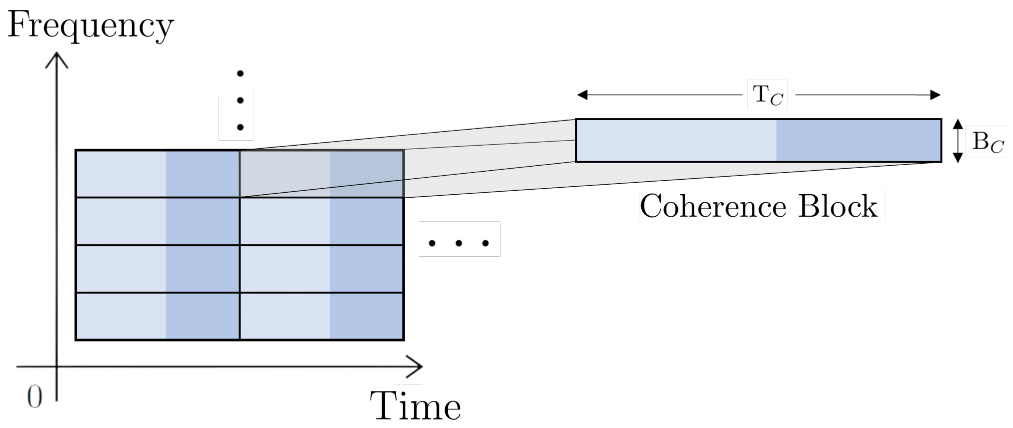


Figure 2.2: Illustration of time-frequency resources divided into coherence blocks.

2.1.1 Coherence Block

If the receiver is moving with a radial velocity v, then the channel *coherence time*, denoted as T_c, is typically defined as

$$T_c = \frac{\lambda}{2v}, \tag{2.2}$$

where λ is the wavelength. Within T_c the channel can be modeled as time-invariant. However, if the signal bandwidth is excessively large, it could exhibit frequency fading at the receiver. To overcome this issue, a limited frequency bandwidth has to be considered so that the channel is approximately constant. This quantity is defined *coherence bandwidth*, denoted as B_c, and it is related to the delay spread (DS), i.e.:

$$B_c = \frac{1}{DS}, \tag{2.3}$$

where the DS can be determined by computing the difference between the arrival times of the first and last signal paths.

2.1.2 TDD Protocol

The most used protocols for channel estimation are the time division duplex (TDD) and the frequency division duplex (FDD) protocols. In this thesis we focus on the TDD, where each coherence block contains $\tau_c = T_c B_c$ symbols and is divided into three parts: there are τ_p symbols for the channel estimation phase, τ_d dedicated for the DL data transmission and τ_u for UL data transmission. However, since only the DL power allocation is considered in this work, we do not consider the UL payload in the coherence block. Fig. 2.3 illustrates the TDD protocol packet. Note that

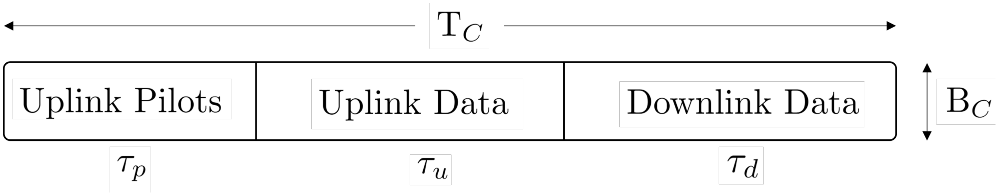


Figure 2.3: TDD protocol packet.

thanks to the channel reciprocity property of the channel, the channel estimation performed with the UL pilots can be utilized also for the DL data transmission. In the literature, the number of samples for the UL pilot sequence varies between 5 and 20. However, in scenarios where there are a high number of active UEs in the network, pilot contamination may arise, resulting in poor channel estimation.

2.1.3 Channel Estimation

The estimation of the wireless channels is commonly performed using UL pilots in TDD-based communication systems. However, there is a tradeoff between the number of samples dedicated to the pilot sequence, which influences the estimation accuracy and the available payload space in a packet. This is due to the fact that the packet duration cannot exceed the channel coherence time T_c. In this thesis, for channel estimation purposes, each UE is assigned a random τ_p-length pilot from an ensemble of τ_p orthogonal sequences. Denoting $t_k \in \{1, \ldots, \tau_p\}$ as the index of the pilot assigned to the UE$_k$, the signal $\mathbf{y}^{(p)}_{t_k l} \in \mathbb{C}^{N\times 1}$ obtained at AP$_l$ after correlating it with the pilot t_k is

$$\mathbf{y}^{(p)}_{t_k l} = \sum_{\substack{i=1\\ t_i = t_k}}^{K} \sqrt{\tau_p p_i}\mathbf{h}_{il} + \mathbf{n}_{t_k l}, \tag{2.4}$$

where p_i is the transmit power of UE i and $\mathbf{n}_{t_k l} \sim \mathcal{CN}\left(\mathbf{0}, \sigma^2 \mathbf{I}_N\right)$ is the additive Gaussian noise vector at AP$_l$. With this information, the channel between the UE$_k$ and the AP$_l$ can be computed using the minimum mean square error (MMSE) estimator at each AP as in [6]

$$\begin{aligned}\hat{\mathbf{h}}_{kl} &= \sqrt{\tau_p p_k}\,\mathbf{R}_{kl}\left(\sum_{\substack{i=1\\ t_i = t_k}}^{K} \tau_p p_i \mathbf{R}_{il} + \sigma^2 \mathbf{I}_N\right)^{-1} \mathbf{y}^{p}_{t_k l} \\ &\sim \mathcal{CN}\Big(\mathbf{0}, \tau_p p_k \mathbf{R}_{kl} \mathbf{\Psi}^{-1}_{kl} \mathbf{R}_{kl}\Big)\end{aligned} \tag{2.5}$$

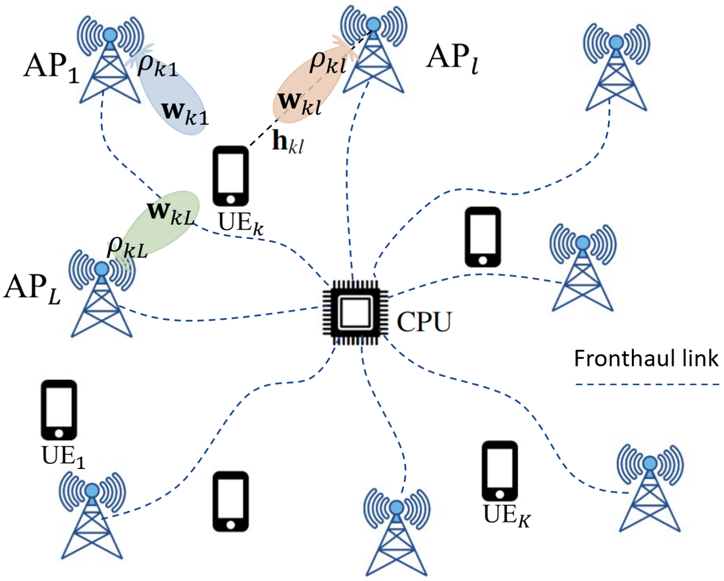


Figure 2.4: Illustration of the considered CFmMIMO network architecture.

where $\boldsymbol{\Psi}_{kl} = \mathbb{E}\left\{\mathbf{y}^p_{t_kl}\left(\mathbf{y}^p_{t_kl}\right)^H\right\} = \sum_{i=1,t_i=t_k}^{K} \tau_p p_i \mathbf{R}_{il} + \sigma^2 \mathbf{I}_N$ represents the correlation matrix of the received pilot signal.

2.1.4 Pilot Assignments

In order to access the network, each UE must be assigned a pilot sequence. Many different pilot assignment schemes exist which can be applied to CF-mMIMO, like the random pilot assignment, wherein each UE in the network is randomly assigned to a pilot belonging to a finite set of available pilot sequences. Despite being simple, this scheme could lead to bad performance, as there is the probability that co-located UEs are assigned to the same pilot sequence, leading to pilot contamination.

However, since the pilot assignment is not the main focus of this work, the low-complexity algorithm proposed in [6] is employed. Assuming there are τ_p ($\tau_p \leq K$) orthogonal pilots available, the first τ_p UEs are assigned orthogonal sequences, while the remaining UEs are assigned to pilots based on the corresponding interference. For UE$_k$ with $k = \tau_p + 1, \ldots, K$, the algorithm proceeds in two steps. First, the UE selects the AP with the strongest signal as the *Master AP*. The determined AP index is computed as:

$$l^* = \underset{l \in \{1,\ldots,L\}}{\arg\max}\, \beta_{kl}.$$

Then, the Master AP compares the channel gains with all previously assigned

$(k-1)$ UEs and selects the pilot sequence $\tau^\star$ with the lowest interference, i.e.,

$$\tau^\star = \underset{\substack{t\in\{1,\dots,\tau\}\\ \tau\in\mathcal{T}}}{\arg\min} \sum_{\substack{i=1\\ t_i=t_k}}^{k-1} \beta_{il^*}. \tag{2.6}$$

This process continues sequentially for the rest of the UEs. This pilot assignment strategy enables a greater focus on the power allocation problem and prevents multiple co-located users from being assigned the same pilot sequence, which can lead to ambiguity in the training set and different channel estimation accuracy for similar channel gains.

2.1.5 Downlink Data Transmission

In a cell-free network, a given UE is served by all the available APs, resulting in a received DL signal at UE$_k$ as follows

$$y_k^{(\mathrm{dl})} = \sum_{l=1}^{L} \mathbf{h}_{kl}^H \sum_{i=1}^{K} \sqrt{\rho_{il}}\mathbf{w}_{il} s_i + n_k, \tag{2.7}$$

where ρ_{il} is the DL power coefficient by AP$_l$ to UE$_k$ using the normalized precoding vector $\mathbf{w}_{il} \in \mathbb{C}^{N\times 1}$ such that $||\mathbf{w}_{il}|| = 1$ within the channel coherence time. Moreover, s_i represents the zero-mean symbol transmitted for UE i and $n_k \sim \mathcal{CN}(0,\sigma^2)$ is the noise at UE$_k$.

The following equation expresses the lower bound for the DL SE in a CF-mMIMO system [5, Lem. 1], where the minimum achievable SE for the UE$_k$ is

$$\mathrm{SE}_k = \frac{\tau_d}{\tau_c}\log_2(1+\mathrm{SINR}_k), \tag{2.8}$$

where

$$\mathrm{SINR}_k = \frac{(\mathbf{a}_k^T\boldsymbol{\mu}_k)^2}{\sum_{i=1}^{K}\boldsymbol{\mu}_i^T\mathbf{B}_{ki}\boldsymbol{\mu}_i - (\mathbf{a}_k^T\boldsymbol{\mu}_k)^2 + \sigma^2} \tag{2.9}$$

represents the SINR for UE$_k$ and

$$\boldsymbol{\mu}_k = [\mu_{k1}\dots\mu_{kL}]^T \in \mathbb{R}^{L\times 1},\ \mu_{kl} = \sqrt{\rho_{kl}} \tag{2.10}$$

$$\mathbf{a}_k = [a_{k1}\dots a_{kL}]^T \in \mathbb{R}^{L\times 1},\ a_{kl} = \mathbb{E}\{\mathbf{h}_{kl}^H\mathbf{w}_{kl}\} \tag{2.11}$$

$$\mathbf{B}_{ki} \in \mathbb{R}^{L\times L},\ b_{ki}^{lm} = \Re\Big(\mathbb{E}\Big\{\mathbf{h}_{kl}^H\mathbf{w}_{il}\mathbf{w}_{im}^H\mathbf{h}_{km}\Big\}\Big). \tag{2.12}$$

The arrays $\boldsymbol{\mu}_k$ and $\mathbf{a}_k$ are defined as the square root of the DL power coefficients and the signal from AP$_{1\dots L}$ to UE$_k$, respectively. The element b_{ki}^{lm} denotes the element (l, m) in matrix $\mathbf{B}_{ki}$, representing the mutual interference between UE$_k$ and UE$_i$. Note that the expected values from (2.11) and (2.12) are intended to be averaged over several channel realizations, such that the channel model is not biased by a particular setup.

Precoding Schemes

In this work, multi-antenna APs are taken into consideration. The use of multiple antennas at the transmitter has the potential to exploit spatial diversity thanks to precoding schemes. In this work, MR and regularized zero-forcing (RZF) precoding schemes are implemented at each AP, defined as follows

$$\bar{\mathbf{w}}_{kl} = \begin{cases} \hat{\mathbf{h}}_{kl} & \text{for MR,} \\ \left(\sum_{i=1}^{K} p_i \hat{\mathbf{h}}_{il} \hat{\mathbf{h}}_{il}^H + \sigma^2 \mathbf{I}_N \right)^{-1} p_k \hat{\mathbf{h}}_{kl} & \text{for RZF,} \end{cases} \tag{2.13}$$

where $\hat{\mathbf{h}}_{kl}$ is the estimated channel between AP$_l$ and UE$_k$, which is computed according to eq. (2.5). The precoding vector is defined as $\mathbf{w}_{kl} = \bar{\mathbf{w}}_{kl} / ||\bar{\mathbf{w}}_{kl}||$, such that the power budget constraint holds within each coherence block.

2.2 Problem Formulation

In this section, the DL power allocation problem is formulated. The objective is to maximize the network SE by optimizing the power coefficients. To further evaluate the performance, spectral and total energy efficiency are taken into consideration, along with Jain's fairness index.

2.2.1 Sum-SE Maximization

The goodness of the power allocation can be measured with the sum-SE metric. Therefore, in this section, the Sum-SE maximization for the power allocation problem is formulated. The objective is to find the DL power allocation coefficients

$$\{\mu_{kl} \in \mathbb{R} : \mu_{kl}^2 = \rho_{kl} : \forall k, l\}$$

that maximize the Sum-SE while guaranteeing the per-AP power budget constraints. The maximization objective is expressed in terms of the variables $\{\mu_{kl}\}$. It is worth noting that no constraint for the sign of μ_{kl} is applied, as the variables always appear in quadratic forms. The Sum-SE maximization problem is formulated as follows:

$$\begin{aligned}
&\underset{\{\mu_{kl}:\forall k,l\}}{\text{maximize}} \sum_{k=1}^{K} \log_2(1+\text{SINR}_k) \\
&\text{subject to} \sum_{k=1}^{K} \mu_{kl}^2 \leq P_{\max}^{(\text{dl})}, \quad l=1,\dots,L,
\end{aligned} \tag{2.14}$$

where $P_{\max}^{(\text{dl})}$ is the maximum available power for each AP, and removing the constant pre-log factor of the spectral efficiency (SE) in (2.8) does not influence the maximization objective. The problem (2.14) is non-convex and finding the optimal solution is highly complicated. In [9] a sub-optimal solution using a complex heuristic iterative algorithm is obtained, called weighted minimum mean square error (WMMSE), and solved with the closed-form alternative direction method of multipliers (ADMM) [5, Alg. 1].

2.2.2 Main Performance Indicators

Total Energy Efficiency

Another important metric used to assess the power allocation solution is energy efficiency, which is further formulated for a cell-free network. In this regard, the power consumption model from [13] is adopted, which takes into account the power related to the APs and fronthaul links as follows:

$$P_{\text{total}} = \sum_{l=1}^{L} P_l + \sum_{l=1}^{L} P_{fh,l}, \tag{2.15}$$

where P_l accounts for the circuit power consumption at the AP$_l$, and $P_{fh,l}$ is the power of the fronthaul link between the AP$_l$ and the CPU. To be more precise, we can express these two quantities in the following equations. The power consumed by the AP$_l$ can be modeled as

$$P_l = \frac{1}{\alpha_m} \sum_{k=1}^{K} \mu_{kl}^2 \beta_{kl} + MP_{tc,l},$$

where $0 < \alpha_m \leq 1$ coefficient represents the power amplifier efficiency, and $P_{tc,l}$ summarizes the power required to supply the circuit components related to each antenna of the AP$_l$. On the other hand, the power attributed to the connection between the CPU and a given AP is expressed as

$$P_{fh,l} = P_{0,l} + B \cdot SE(\{\mu_{kl}\}) \cdot P_{t,l}$$

where B is the system's bandwidth, whereas $P_{0,l}$ and $P_{t,l}$ denote the traffic-independent and the traffic-dependent power required to run the l^{th} link, respectively. The latter quantity is expressed in Watt per bit/s. Finally, the total energy efficiency is defined as the ratio between the sum-throughput (bit/s) and the total power consumption (Watt):

$$\mathrm{EE} = \frac{B \cdot \sum_{k=1}^{K} \mathrm{SE}_k(\boldsymbol{\mu}_k)}{P_{\text{total}}}. \tag{2.16}$$

Jain's Fairness Index

The Jain's fairness index for the DL power allocation problem can be formulated as

$$\mathcal{J}(\mathbf{SE}) = \mathcal{J}(\mathrm{SE}_1, \ldots, \mathrm{SE}_K) = \frac{\left(\sum_{k=1}^{K} \mathrm{SE}_k\right)^2}{K \sum_{k=1}^{K} \mathrm{SE}_k^2}, \tag{2.17}$$

where $\frac{1}{K} \leq \mathcal{J}(\mathbf{SE}) \leq 1$ represents an index that quantifies the fairness among all UEs. Jain's fairness index achieves the unitary value when all the UEs are allocated the same resource, which represents the best case in terms of fairness.

Chapter 3

Learning-Based Power Allocation

3.1 Unsupervised Learning Approach

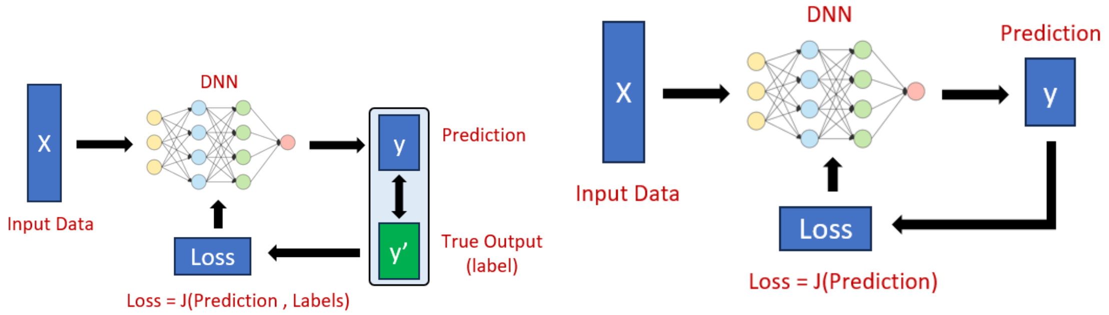
This section aims to outline the essential characteristics of the unsupervised learning approach, which is adopted in this work for power allocation purposes. The capacity to approximate almost any non-linear function mapping is one of the most fascinating features of neural networks, along with the ability to handle huge amounts of parameters. However, to achieve such capabilities, the DNN models need extensive training before being deployed. It is important to emphasize that the size of the training dataset, i.e. training set, is not constrained; in fact, having more data for training is typically beneficial for neural network generalization. In deep learning, indeed, the goal is to enable a model to predict unknown patterns. Furthermore, to measure the quality of the training, the loss function metric is employed. It maps the DNN output to a scalar value as follows

$$\begin{cases} \mathcal{L}_{\text{sup}} = J_1(y_{\text{pred}}, y_{\text{true}}) \\ \mathcal{L}_{\text{unsup}} = J_2(y_{\text{pred}}), \end{cases} \tag{3.1}$$

where $\mathcal{L}_{\text{sup}}$ and $\mathcal{L}_{\text{unsup}}$ represent the supervised and unsupervised loss function, whereas y_{pred} and y_{true} are the DNN prediction and expected outcome, respectively.

There are two main ways of training a DNN: supervised and unsupervised learning, whose differences are shown in Fig. 3.1. In the supervised learning approach, each entry of the training set is associated with a label, y_{true}, which is required to further compute the loss function. For instance, mean squared error (MSE) and mean absolute error (MAE) are two examples of supervised loss functions, defined as

$$\mathcal{L}_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^{n} \left(y_{\text{pred},i} - y_{\text{true},i} \right)^2$$



(a) Supervised. (b) Unsupervised.

Figure 3.1: Different ways of training a neural network: supervised and unsupervised learning techniques.

$$\mathcal{L}_{\text{MAE}} = \frac{1}{n}\sum_{i=1}^{n}\Big|y_{\text{pred},i} - y_{\text{true},i}\Big|,$$

where n is the dimension of the output layer. In both cases, when the loss function approaches zero indicates that the model is effectively approximating the target solution. However, the unsupervised learning counterpart implies some important differences in the training phase. Firstly, the expected outcome is not required, since the loss function only depends on the DNN prediction. This represents a huge advantage as label generation is one of the main drawbacks of supervised learning techniques, being time-consuming and resulting in additional complexity. In this regard, one can arbitrarily define a custom loss function that the DNN model aims to minimize. It is worth noting that, unlike the supervised learning approach, the loss value is not constrained to be zero, and can eventually become negative signed. In this work, the outlined advantages of unsupervised learning are exploited, and the following sections will explore the application of this approach in a CF-mMIMO network.

3.2 Centralized DNN

In a centralized approach, the CPU is responsible for managing the network-wide resources. A fully-connected feed-forward DNN is indeed defined at the CPU, where the inputs are the set of LSF coefficients $\{\beta_{kl} : \forall(k,l)\}$, as depicted in Fig. 3.2a, sensed by all the APs. The best square root of the transmit power coefficients provided by the model $\{\mu^*_{kl} : \forall(k,l)\}$ are intended to serve all the UEs in the network.

3.2.1 Heuristic Pre-processing

The LSF coefficients are properly processed before being set as the DNN input. We adopt the same heuristic scaling as [9] which is similar to [14]. The fractional DL power allocation, according to the authors, results in improved performance during the DNN training, and is formulated as follows:

$$\rho'_{kl} = \sqrt{P^{\text{dl}}_{\text{max}}} \frac{(\beta_{kl})^v}{\sum_{i=1}^{K}(\beta_{il})^v} \quad \forall(k,l), \tag{3.2}$$

where v is a tunable parameter that attenuates the channel gains. The term "fractional" refers to the fact that the heuristic scaling in eq. (3.2) assigns a fraction of the available power at AP_l to each UE, according to the LSF coefficients.

3.2.2 DNN Structure

In Table 3.1 the DNN structure is summarized. The inputs are the scaled version of all the channel gains, as shown in eq. (3.2), while the square roots of the power allocation coefficients are the output of the DNN. *Relu* and *linear* activations are used in the hidden and output layer(s) respectively. The number of layers, as well as neurons per layer, have been chosen from [9] and then adapted via trial and error methodology. The use of batch normalization layers keeps a limited range of values, resulting in increased numerical stability, thus mitigating the occurrence of exploding or vanishing gradients during the training phase. A detailed representation of the training process is shown in Fig. 3.3a, where the heuristic pre-processing of the input and the loss function computation are highlighted.

3.3 Distributed DNN

To allow for a scalable solution, a fully-distributed DNN implementation is proposed. In this scheme, as illustrated in Fig. 3.2 and depicted in more detail in Fig. 3.3b, each AP is equipped with a proper DNN, trained using solely local channel information from all the UEs $\{\beta_{kl} : \forall k\}$, easily measurable at the AP level. A fully-connected feed-forward DNN is employed, whose structure is illustrated in Table 3.2. It is worth noting that the distributed DNN model may not provide an optimal solution to the problem, as it only takes local information as input for each DNN. This is in contrast to a centralized scheme, where all the network-wide information is known at the CPU.

With the aim of maximizing the training performance of the distributed DNN, the raw LSF coefficients need to be properly scaled before being processed by the model. In this case, rather than computing the ensemble at the CPU, each $\text{AP}_l, \forall l$ performs the heuristic scaling given in (3.2), $\forall k$.

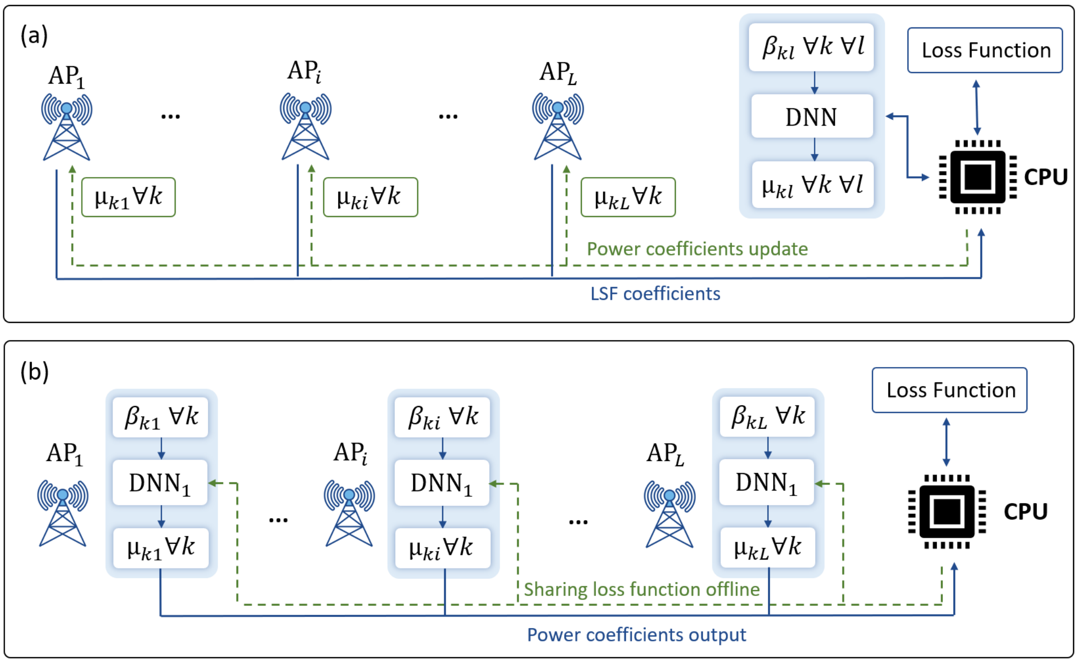


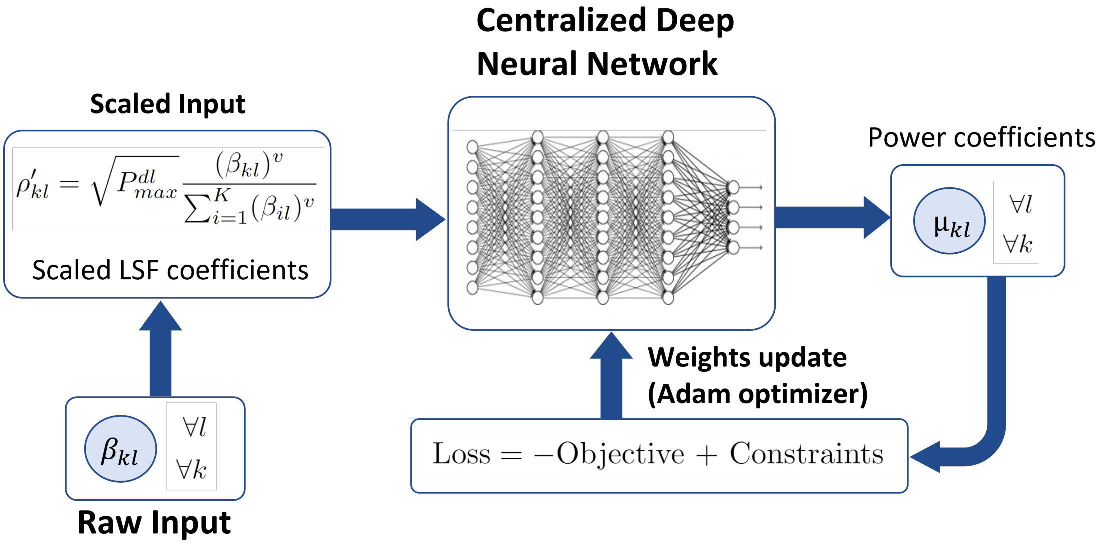
Figure 3.2: Learning stage illustration for (a) centralized and (b) distributed DNNs.

3.3.1 DNN Structure

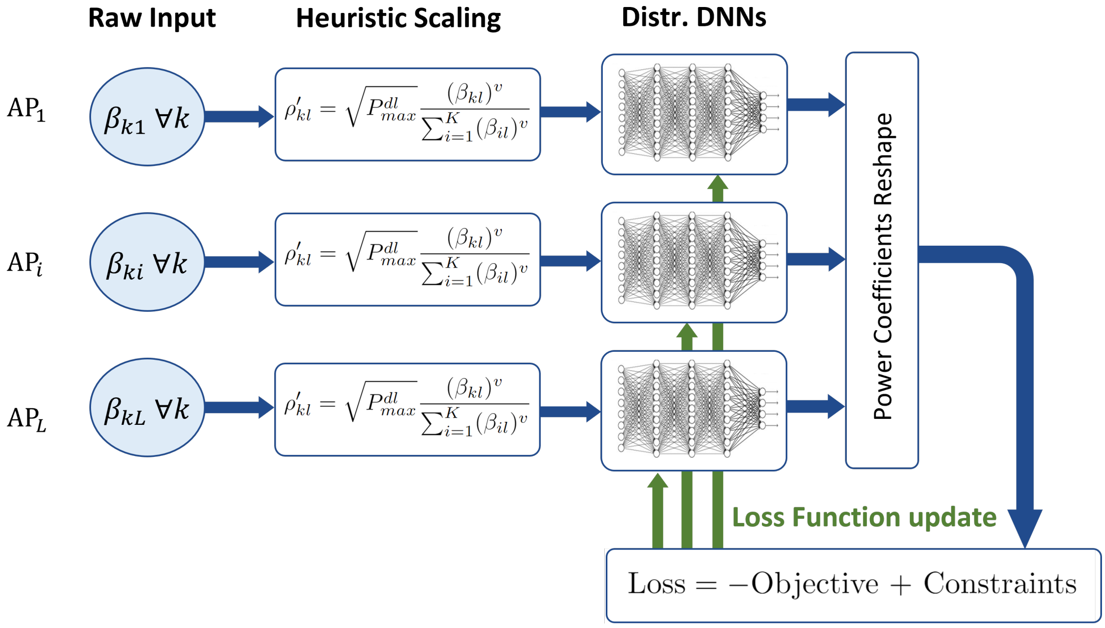
The DNN structure related to a given AP_l is summarized in Table 3.2, it encompasses three hidden layers with batch normalization layers in between, for the same reason as per the centralized DNN. The layout of the distributed DNN also includes an input layer with K neurons and an output layer with the same dimension. *Relu* and *Linear* activation functions are employed for the hidden and output layer(s) respectively.

3.4 Distributed DNN with Side Information

Aiming to improve the performance of the distributed DNN, we incorporate an additional side information from all APs. The employment of a supplementary feature vector confers higher performance than the classic distributed model, at the expense of slight increase of computations by the UEs to share the additional information.


Centralized Deep Neural Network
Scaled Input
$\rho'_{kl} = \sqrt{P^{dl}_{max} \frac{(\beta_{kl})^v}{\sum_{i=1}^{K} (\beta_{il})^v}}$
Scaled LSF coefficients
Power coefficients
μ_{kl} $\forall l$ $\forall k$
β_{kl} $\forall l$ $\forall k$
Raw Input
Weights update (Adam optimizer)
Loss = −Objective + Constraints

(a) Centralized DNN.


Raw Input
Heuristic Scaling
Distr. DNNs
AP_1
AP_i
AP_L
$\beta_{k1}\ \forall k$
$\beta_{ki}\ \forall k$
$\beta_{kL}\ \forall k$
$\rho'_{kl} = \sqrt{P^{dl}_{max} \frac{(\beta_{kl})^v}{\sum_{i=1}^{K} (\beta_{il})^v}}$
Power Coefficients Reshape
Loss Function update
Loss = −Objective + Constraints

(b) Distributed DNN.

Figure 3.3: Structure of the DNN training for the DL power allocation problem.

Table 3.1: Structure of the proposed centralized DNN.

	Size	Parameters	Activation Function
Input	KL	-	-
Layer 1 (Dense)	480	154080	Relu
Batch Norm	480	1920	-
Layer 2 (Dense)	640	307840	Relu
Batch Norm	640	2560	-
Layer 3 (Dense)	480	307680	Relu
Layer 4 (Dense)	KL	153920	Linear

Table 3.2: Structure of the proposed Distributed DNN, related to one AP.

	Size	Parameters	Activation Function
Input	K	-	-
Layer 1 (Dense)	40	840	Relu
Batch Norm	40	160	-
Layer 2 (Dense)	60	2460	Relu
Batch Norm	60	240	-
Layer 3 (Dense)	40	2440	Relu
Layer 4 (Dense)	K	820	Linear

Table 3.3: Structure of the proposed Distributed DNN with Side Information, related to one AP.

	Size	Parameters	Activation Function
Input	$2K$	-	-
Layer 1 (Dense)	60	1260	Relu
Batch Norm	60	240	-
Layer 2 (Dense)	120	7320	Relu
Batch Norm	120	480	-
Layer 3 (Dense)	60	7260	Relu
Batch Norm	60	240	-
Layer 4 (Dense)	40	2440	Relu
Layer 5 (Dense)	K	820	Linear

3.4.1 DNN Structure

The architecture of the distributed DNN with side information is outlined in Tab 3.3. Owing to the additional feature vector, the input dimension is doubled with respect to the distributed DNN, thus having $2K$ input neurons. Then, the dimension of the hidden layers enlarges to 128 in the second hidden layer and it subsequently decreases until the output layer. As for the other proposed models, batch normalization layers have been employed in between the hidden layers with *Relu* activation functions for the hidden layers and *Linear* activation concerning the output layer.

3.4.2 Side Information

In addition to the classic distributed DNN scheme, another feature vector (side information) is considered as input to each DNN. The main reason for introducing it is to attain higher spectral efficiency by exploiting the knowledge of non-local channel statistics. Specifically, the distributed model only accounts for local channel information, thereby precluding the knowledge of the inter-relations between the LSF coefficients. Therefore, along with (3.2), the side information can be utilized to produce an additional DNN input by integrating it in the equation below:

$$\rho''_{kl} = \sqrt{P^{\text{dl}}_{\max}} \frac{(\beta_{kl})^v}{\sum_{i=1}^{L}(\beta_{ki})^v} \quad \forall k, \tag{3.3}$$

where it expresses the ratio of the LSF coefficient from the AP$_l$ to UE$_k$, to the LSF coefficients sensed from all APs to UE$_k$. The side information can be extrapolated without the need to exchange data between APs via fronthaul links. Let us assume moving the UE side the (3.3) denominator computation. By utilizing this approach, each UE then broadcasts the relation between itself and every other AP in the network using standard control signaling channels [9].

3.5 Custom Loss Functions

When a large number of parameters have to be optimized using unsupervised learning, Lagrange multipliers can be exploited to translate the sum-SE maximization problem in a custom loss function [15]. The flexibility in defining custom loss functions is one of the advantages of unsupervised learning. In this section, we propose different types of loss functions each tailored to specific computation, and they all share a common objective of maximizing the sum-rate. Of particular importance is the fact that all designed loss functions incorporate a term dedicated to meeting the per-AP power constraint.

Sum-SE Loss Function

We define a custom loss function as follows

$$\mathcal{L}_{\text{sum-SE}} = \underbrace{-\frac{1}{K}\sum_{k=1}^{K}\log_2(1+\text{SINR}_k)}_{\text{Objective}} + \underbrace{C\sum_{l=1}^{L}\text{relu}\left(\sum_{k=1}^{K}\mu_{kl}^2 - P_{\max}^{\text{dl}}\right)}_{\text{Constraints}}, \tag{3.4}$$

where it contains the opposite of the sum-rate and a term proportional to the DL power budget constraints, using a constant $C = 0.1$ as a Lagrange multiplier. Note that by minimizing the loss function defined in (3.4), we can go beyond the limits of sub-optimal labels and improve the DNN's performance as long as the per-AP power budget constraint is met[1].

Max-prod Loss Function

Another custom loss function based on maximizing the product of the SINR among the UEs, namely max-prod, is defined as follows

$$\mathcal{L}_{\text{max-prod}} = -\log_2\prod_{k=1}^{K}\text{SINR}_k + \underbrace{C\sum_{l=1}^{L}\text{relu}\left(\sum_{k=1}^{K}\mu_{kl}^2 - P_{\max}^{\text{dl}}\right)}_{\text{Constraints}}, \tag{3.5}$$

where the first term is derived from [11], and the second is added in order to ensure that the power budget constraint is met.

Max-min Loss Function

Lastly, the max-min loss function is customized from [11]. The term max-min refers to the maximization of the minimum SINR among the UEs. Let σ represent the sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}},$$

[1] The SINR_k computation requires the knowledge of $\mathbf{a}_k$ $\forall k$ and $\mathbf{B}_{ki}$ $\forall(k,i)$ as in (2.9). However, this information is required only during the training phase: once the models are fully trained they solely rely on the LSF coefficients during inference.

then, the custom max-min loss function is defined as follows

$$\begin{aligned}\mathcal{L}_{\text{max-min}} = &- \sum_{k=1}^{K} \sigma\left(\frac{0.3}{\text{SINR}_k + 0.001}\right)\\ &\underbrace{+ A \cdot \text{SINR}_{\text{min}} + C \sum_{l=1}^{L} \text{relu}\left(\sum_{k=1}^{K} \mu_{kl}^2 - P_{\text{max}}^{\text{dl}}\right)}_{\text{Constraints}},\end{aligned} \tag{3.6}$$

where A and C are constant values and, again, the power budget constraint term is added to act as a penalty in the case it is not met. The first two terms of the max-min loss function are inherited from [11]. It is worth noting that the max-min loss not only focuses on maximizing the minimum SINR. In fact, the sum of sigmoid functions ensures UE fairness, by optimizing the SINR of the UEs with better channel conditions as well.

3.6 LSF-based AP Selection

CF-mMIMO networks are characterized by a large number of fronthaul connections, compared to co-located mMIMO systems, to allow data transfer between the CPU and the APs. Consequently, this section presents an LSF-based AP selection algorithm from [13] that mainly focuses on seeking a balance between the AP involvement while maintaining a comparable performance achievable by utilizing all available APs. To this end, the strongest subset of APs that contribute to spatial diversity gains, i.e. the closest, are selected. Due to the impact of path loss, APs located at a significant distance from a specific UE contribute less to its overall performance, therefore, the selection of APs is based on the largest LSF coefficients. The algorithm is described in Alg. 1. It is worth noting that the selection is performed after the DNN inference. In this process, the power coefficients of non-selected APs are set to zero, ensuring the APs' deactivation. This step is determined by the criterion that considers the $\delta_{\%}$ of the channel gain, as follows

$$\sum_{l=1}^{|\mathcal{L}_k|} \frac{\overline{\beta}_{lk}}{\sum_{i=1}^{L} \beta_{ik}} \geq \delta_{\%}, \tag{3.7}$$

where $\overline{\beta}_k = \{\overline{\beta}_{1k}, \ldots, \overline{\beta}_{Lk}\}$ is the set of LSF coefficients sorted in descending order. The subset $\mathcal{L}_k$, with $|\mathcal{L}_k| \leq L$, denotes the group of APs for the k^{th} UE that satisfies the condition stated in eq. (3.7), and $\delta_{\%} \in (0, 1]$ is the AP selection parameter, indicating the desired percentage threshold. Note that, as $\delta_{\%}$

Algorithm 1: Largest-LSF-based AP selection algorithm for UE$_k$.

```
Input: LSF coefficients $\beta_k = \{\beta_{1k}, \dots, \beta_{Lk}\}$, selection parameter $\delta_{\%}$
Output: Selected subset of APs $\mathcal{L}_k$, $|\mathcal{L}_k| \leq L$
$\mathcal{L}_k = \emptyset$          // Set of active APs for the $k^{\text{th}}$ UE
$S = 0$             // Cumulative sum
$\overline{\beta}_k \leftarrow$ Sort $\beta_k$ in descending order
for $\overline{\beta}_{lk}$ in $\overline{\beta}_k$ do
    Calculate $\frac{\overline{\beta}_{lk}}{\sum_{i=1}^{L} \overline{\beta}_{ik}}$
    Update $S = S + \frac{\overline{\beta}_{lk}}{\sum_{i=1}^{L} \overline{\beta}_{ik}}$
    $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup \text{AP}_{\mathbf{i}(l)}$
    if $S \geq \delta\%$ then
        Break
    end
end
return $\mathcal{L}_k$
```

diminishes, the selection criterion becomes more stringent, leading to fewer power coefficients set to zero, which enables system optimization based on the desired tradeoff between performance and energy efficiency.

3.7 Complexity Analysis

This section provides a computational complexity analysis related to the proposed DNN-based power allocation, compared to existing solutions that make use of optimization-based algorithms [5] and supervised learning-based frameworks, such as [9]. The complex optimization-based algorithm WMMSE-ADMM [5] is characterized by a computational complexity of $\mathcal{O}(L^3K)$, as the $L \times L$ matrix inverse in [5, Eq. (48)] has the major impact in terms of big-O, and it has to be performed for each AP-UE pair in the network.

As stated in sec. 3.1, the unsupervised learning approach circumvents the computational complexity associated with the WMMSE-ADMM algorithm, since the generation of training labels is not required. It is worth emphasizing that the label generation phase in supervised learning-based approaches mainly contributed to the overall computational complexity. However, the supervised and unsupervised learning methods share the complexity in training the neural networks. In particular, a DNN equipped with Γ layers, with γ_i being the number of neurons of the

Table 3.4: Computational time comparison in milliseconds.

	ADMM [5]	**Sup.** [9]	**Uns. Distr.**	**Uns. Distr-SI**	**Uns. Centr.**
MR	88,6	8,8	8,8	10	4,6
RZF	131,7	9,2	9,2	10,3	5,2

i^{th} layer, the total amount of required multiplications and additions is $\gamma_i\gamma_{i-1}$ for $i = 1, \dots, \Gamma$. In addition, each layer has to compute $\sum_{i=1}^{\Gamma} \gamma_i$ activation functions. The DNN model training has to be performed only once. The distributed DNN represents a more scalable architecture than the centralized one, as the input and output layers do not depend on L. Hence, as the number of APs L increases, the centralized DNN's overall size may not constitute an implementable solution because of the training complexity. Furthermore, in the inference stage, the centralized solution requires all the APs to transmit the relative LSF coefficients to the CPU, which then predicts the power coefficients through the trained DNN and sends them back to the APs via the fronthaul links. This process results in latency in the communication between APs and the CPU. However, this phenomenon does not arise in the distributed DNN solution, as each AP utilizes its local estimated LSf coefficients to predict its own power coefficients.

Moreover, the computational time comparison between the WMMSE-ADMM, supervised, and proposed unsupervised learning solutions is provided in Tab. 3.4. It is obtained by averaging the inference stage over multiple iterations in an Intel i7 Core processor. Each entry in Tab. 3.4 refers to the entire power allocation process. It can be seen that the learning-based power allocation requires about one order of magnitude less than the optimization-based one, resulting in about 10 ms. This substantial reduction in the inference time allows for real-time power allocation updates, demonstrating the capability of the deep learning approach to mitigate the computational time associated with traditional optimization-based methods, such as the WMMSE-ADMM algorithm [5].

Chapter 4

Performance Evaluation

In this chapter, the performance of the proposed DNN models is evaluated, comparing them to benchmark solutions in terms of spectral efficiency and energy efficiency. Additionally, the training performance of the custom DNN loss functions is analyzed.

4.1 Simulation Setup

In this section, a comprehensive description of the simulation setup is provided, which pertains to an urban microcell environment. Our study focuses on a CF-mMIMO network operating within a $1000\,\mathrm{m} \times 1000\,\mathrm{m}$ coverage area, whose parameters are illustrated in Tab. 4.1. The simulation setup comprises $L = 16$ APs, equipped with $N = 4$ antennas, jointly serving a population of $K = 20$ uniformly distributed UEs. Precisely, the APs are placed in a 4×4 grid with an additional random noise term. Fig. 4.2 illustrates a snapshot of the system. The performance evaluation dataset encompasses 10.000 UE distributions, independent of the training set. We consider both orthogonal and non-orthogonal pilots' case ($\tau_p = \{10, 20\}$) over a $20\,\mathrm{MHz}$ bandwidth channel with a $-94\,\mathrm{dBm}$ receiver noise power. The maximum DL available power for each AP is designated as $P_{\max}^{\mathrm{dl}} = 1$ W, whereas each UE transmits with a power of $p_i = 100$ mW during the channel estimation phase. The path loss model β_{kl} between the UE$_k$ and the AP$_l$ matches well with the 3GPP urban microcell standard at the carrier frequency of 2 GHz, as [1, 5].

$$\beta_{kl} = -30.5 - 36.7 \log_{10}\left(\frac{d_{kl}}{1\mathrm{m}}\right) \tag{4.1}$$

where d_{kl} represents the distance between UE$_k$ and AP$_l$, and the path loss exponent is $\alpha = 3.67$.

Table 4.1: Table of the Simulation Parameters

Parameter	**Value**
Area of interest (wrap around)	$1000\,\mathrm{m} \times 1000\,\mathrm{m}$
Carrier frequency	$2\,\mathrm{GHz}$
Bandwidth	$20\,\mathrm{MHz}$
Number of APs	$L = 16$
Number of UEs	$K = 20$
Number of antennas per AP	$N = 4$
Pathloss exponent	$\alpha = 3.76$
Per-AP maximum DL transmit power	$P^{\mathrm{dl}}_{\max} = 1\,\mathrm{W}$
UL transmit power	$p_i = 100\,\mathrm{mW}$
UL/DL noise power	$-94\,\mathrm{dBm}$
Coherence block length	$\tau_c = 200$
Pilot sequence length	$\tau_p = 10$

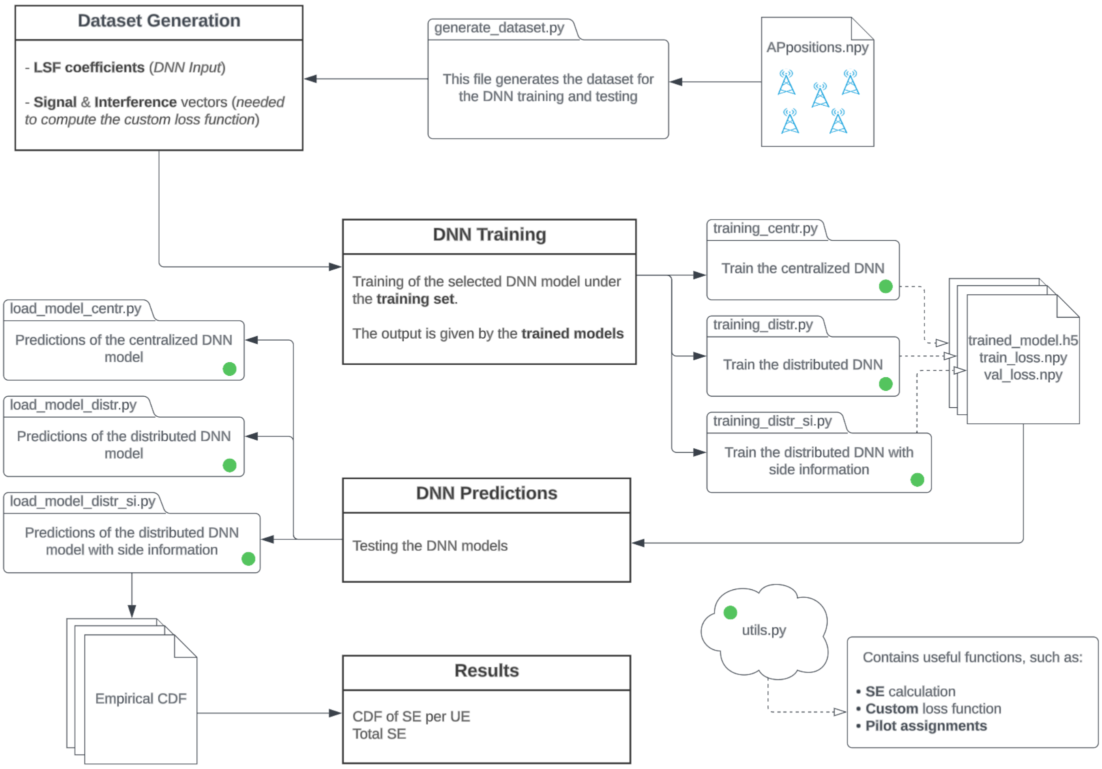

Dataset Generation
- LSF coefficients (DNN Input)
- Signal & Interference vectors (needed to compute the custom loss function)
generate_dataset.py
This file generates the dataset for the DNN training and testing
APpositions.npy
DNN Training
Training of the selected DNN model under the training set.
The output is given by the trained models
training_centr.py
Train the centralized DNN
training_distr.py
Train the distributed DNN
training_distr_si.py
Train the distributed DNN with side information
trained_model.h5
train_loss.npy
val_loss.npy
load_model_centr.py
Predictions of the centralized DNN model
load_model_distr.py
Predictions of the distributed DNN model
load_model_distr_si.py
Predictions of the distributed DNN model with side information
DNN Predictions
Testing the DNN models
Empirical CDF
Results
CDF of SE per UE
Total SE
utils.py
Contains useful functions, such as:
• SE calculation
• Custom loss function
• Pilot assignments

Figure 4.1: Flowchart description of the proposed unsupervised learning framework.

The centralized and distributed DNN models are trained with the methodologies depicted in Fig. 3.3a and 3.3b, respectively, utilizing the Tensorflow framework and an Nvidia RTX A6000 Graphing Processing Unit (GPU). Regarding the dataset, 50.000 network setups are considered each generated with varying UE locations, given a fixed APs topology. The dataset is partitioned into 90% for the training set and 10% for the validation set. The Adam optimizer facilitates gradient descent, and the learning rate is progressively adjusted from an initial value of 0.001. Early stopping is adopted to circumvent overfitting during training, and the number of epochs and batch size are determined via a trial-and-error methodology.

4.2 Code Instructions

The flowchart in Fig. 4.1 illustrates the step-by-step process for executing the proposed unsupervised learning framework in Python language. Particular emphasis is attributed to the description of the key stages involved in the process, from data preparation to model evaluation.

To generate the dataset for feeding the DNN models, the `generate_dataset.py` file has to be run. It stores the entire dataset of channel gains, signals, and interference of the network for the regularized zero-forcing (RZF) and maximum ratio (MR) precoding schemes, i.e.: `dataset__betas_MR.npy`, `dataset__a_MR.npy`, and `dataset__B_MR.npy`.

After the dataset generation, the desired DNN model has to be selected and trained between `training_[model].py`, where `[model]` can be either `centr`, `distr`, or `distr-si`, depending on the chosen DNN. Each code outputs a file with the extension `.h5` containing the trained weights of the DNN, and training/validation loss functions inside the `loss/[model]` folder. Subsequently, the DNN testing can be performed by running `load_model_[model].py`, which takes as input the trained DNN weights and the dataset and outputs the spectral efficiency of the current solution. Indeed, `load_model_[model].py` performs the DNN inference and calculates the cumulative density function (CDF) of the SE per user and total SE averaged over plenty of user distributions. Finally, the results are grouped and loaded by the `plots.py` file, which generates the figures used in this work and in [16].

4.3 Simulation Results

4.3.1 Orthogonal vs Non-orthogonal Pilots

This subsection aims to assess the performance of the centralized and distributed DNN models previously described in section 3. The sum-SE maximization objec-

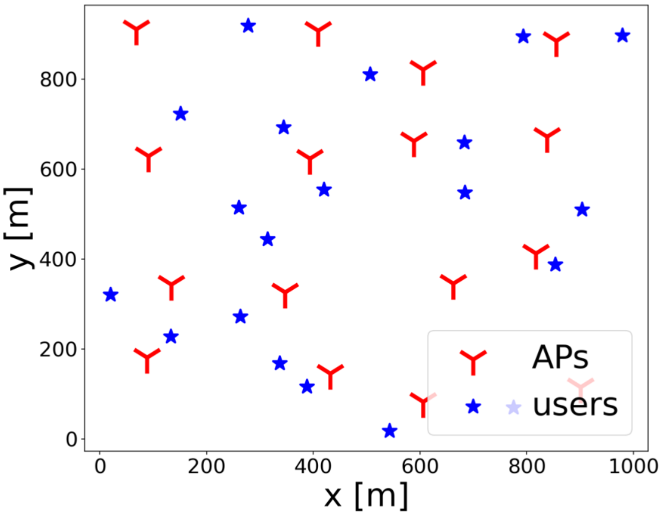
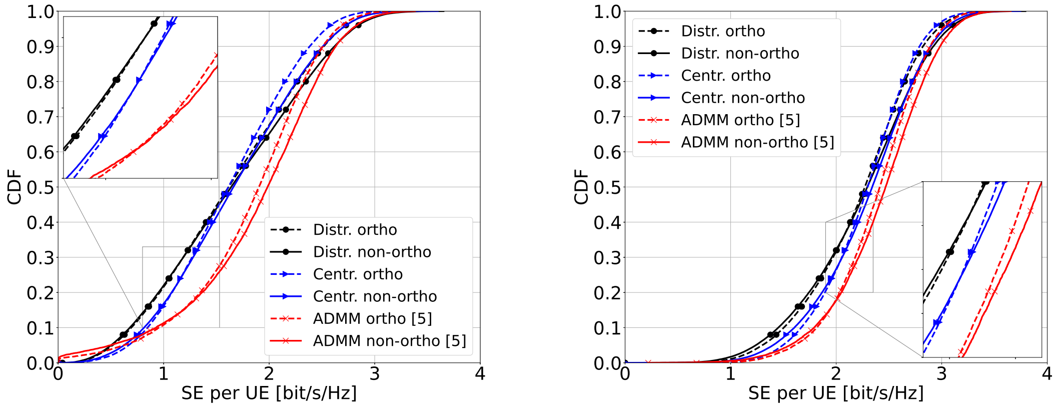


Figure 4.2: A realization of the cell-free network simulation, with uniformly distributed UEs in the squared area.

tive is considered, and the CDF of the spectral efficiency per UE is shown in Fig. 4.3, under plenty of UE locations. In particular, Fig. 4.3a and Fig. 4.3b consider the CDF of spectral efficiency per UE according to MR and RZF precoding schemes, respectively. As a benchmark, the proposed unsupervised DNN models are compared to the optimization-based WMMSE-ADMM [5] algorithm utilized in [9] to generate the training labels. Let us focus on the UEs that experience bad channel conditions, that is, lower values of spectral efficiency. Conversely, a slight performance improvement in the orthogonal pilots' case can be noticed, regardless of the precoding scheme being MR or RZF. Indeed, pilot contamination happens when utilizing non-orthogonal pilots, leading to poor channel estimation for the most unfortunate UEs due to the added interference. Given UEs with good channel conditions, i.e. high SINR, the channel estimation accuracy is higher. In this case, it is not crucial to have more pilot symbols (i.e. orthogonal pilot's case), and having more space for data in the coherence block allows for better throughput when non-orthogonal pilots are used. Overall, our proposed DNNs exhibit similar capabilities in approximating the power coefficients when utilizing either orthogonal or non-orthogonal pilots. As a result, the subsequent sections will focus on a more realistic scenario with non-orthogonal pilots.

4.3.2 Custom Loss Functions Comparison

The unsupervised learning approach has proven to be significantly more versatile and practical compared to its supervised counterpart. Hence, in this subsection,



(a) MR precoding.

(b) RZF precoding.

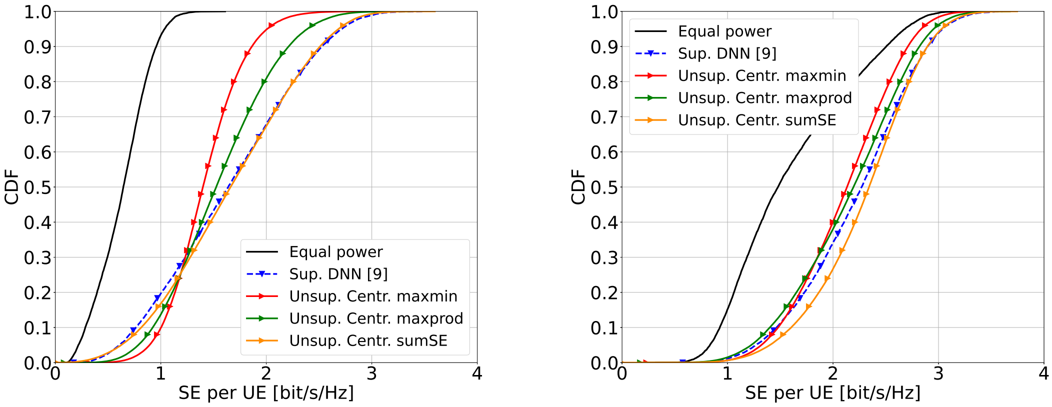
Figure 4.3: Empirical CDF of the DL spectral efficiency per UE, orthogonal and non-orthogonal pilots with $\mathcal{L}_{\text{sum-SE}}$ loss function, utilizing different precoding schemes.

the benefits of unsupervised learning are exploited by leveraging its flexibility to approximate the power allocation coefficients using custom-designed loss functions.

Performance of the DNN models

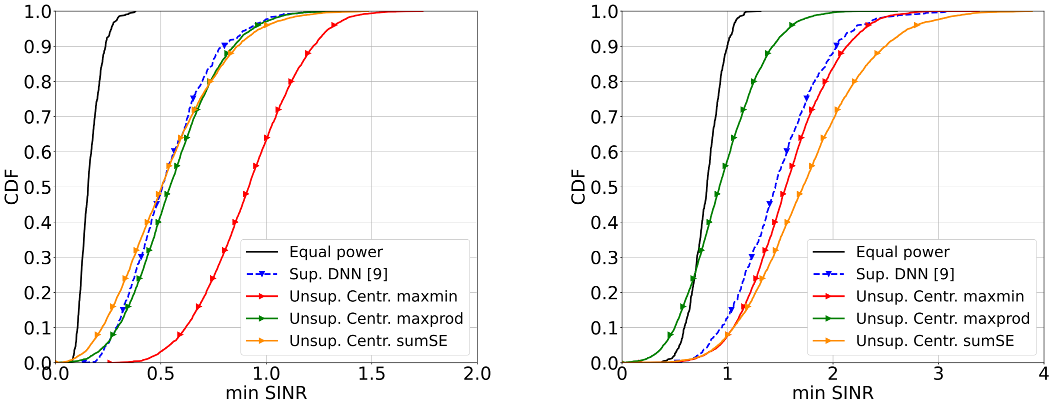
Fig. 4.4 illustrates the spectral efficiency per UE for the centralized DNN, where Fig. 4.4a and Fig. 4.4b have been obtained using MR and RZF precoding, respectively. For the sake of comparison, we show the effectiveness of the $\mathcal{L}_{\text{max-min}}$ (3.6) and $\mathcal{L}_{\text{max-prod}}$ (3.5), along with the $\mathcal{L}_{\text{sum-SE}}$ (3.4) loss function. It is evident that the centralized DNN based on the $\mathcal{L}_{\text{sum-SE}}$ loss function is the best configuration for the RZF precoding, outperforming the supervised learning-based solution. However, for MR precoding in Fig. 4.4a, $\mathcal{L}_{\text{max-prod}}$ focuses on proportional fairness, which strikes a good balance between max-min fairness and sum-SE maximization. On the other hand, $\mathcal{L}_{\text{max-min}}$ yields enhanced fairness among the users with inferior channel conditions, albeit at the expense of sacrificing a portion of the spectral efficiency for the most fortunate UEs.

In fact, as Fig. 4.5a depicts the CDF of the minimum SINR for MR precoding, it is noticeable how the $\mathcal{L}_{\text{max-min}}$ loss function provides the best UE fairness, which is quantified in Table 4.2 with the Jain's fairness index. It is worth recalling that the unitary value is achieved when all the UEs experience the same SE, which represents the best case in terms of fairness. According to Table 4.2, $\mathcal{L}_{\text{max-min}}$ loss function provides the best fairness with a value of 0.95. Regarding Fig. 4.5b, which refers to the CDF of the minimum SINR for the RZF precoding case, the model that showcases the highest performance is the one with $\mathcal{L}_{\text{sum-SE}}$ custom loss



(a) MR precoding. (b) RZF precoding.

Figure 4.4: CDF of the DL spectral efficiency per UE regarding the centralized DNN, evaluated with different custom loss functions.

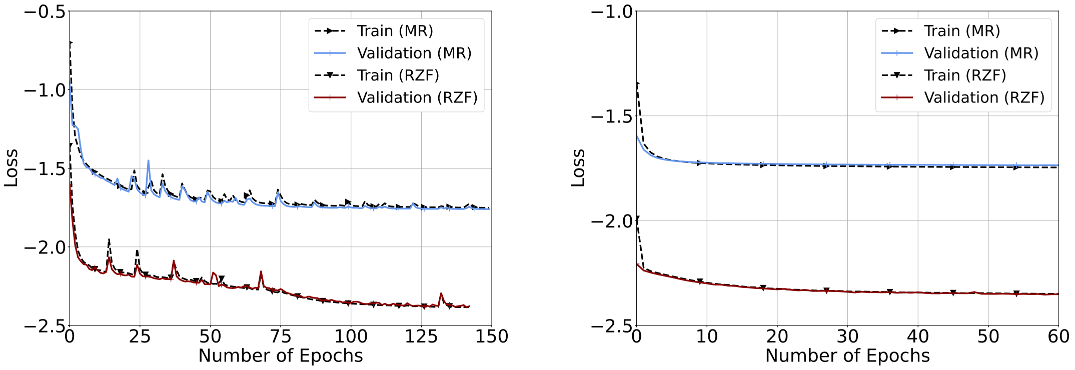


(a) MR precoding. (b) RZF precoding.

Figure 4.5: CDF of the minimum SINR evaluated for different custom loss functions, equal power allocation, and supervised learning counterpart.

Table 4.2: Jain's fairness index table.

	Sum-SE	**Max-Min**	**Max-Prod**	**Equal P.**	**Sup.**
MR	0,88	0,95	0,89	0,87	0,86
RZF	0,96	0,96	0,92	0,89	0,95



(a) Centralized DNN

(b) Distributed DNN

Figure 4.6: Training loss function for centralized and distributed DNN.

function, with a UE fairness of 0.96. Therefore, the following subsections will consider only the loss function with better results, that is $\mathcal{L}_{\text{sum-SE}}$.

Training Loss

The training loss function curve for the centralized and distributed DNNs is shown in Fig. 4.6. Fig. 4.6a showcases the centralized DNN's train and validation loss in the case of MR and RZF precoding. It can be seen that the train and validation curves decrease and are close to each other, while the learning stage is interrupted through the use of early stopping, aimed to avoid overfitting. Moreover, the presence of the spikes is attributed to the power constraint term in expression (3.4) being temporarily violated. This is due to the fact that the centralized model, aiming to improve the overall performance, attempts to increase the transmitting power at the APs, resulting in a penalty in the $\mathcal{L}_{\text{sum-SE}}$ loss function. On the other hand, the distributed DNN loss function in Fig. 4.6b shows a similar trend: the loss decreases with increasing epochs and the absence of the spikes can be attributed to the fact that the smaller DNN model only controls one AP, requiring fewer epochs to train and providing smoother learning.

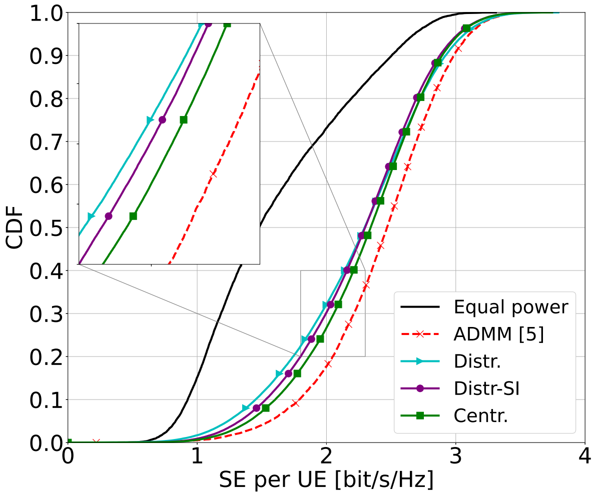


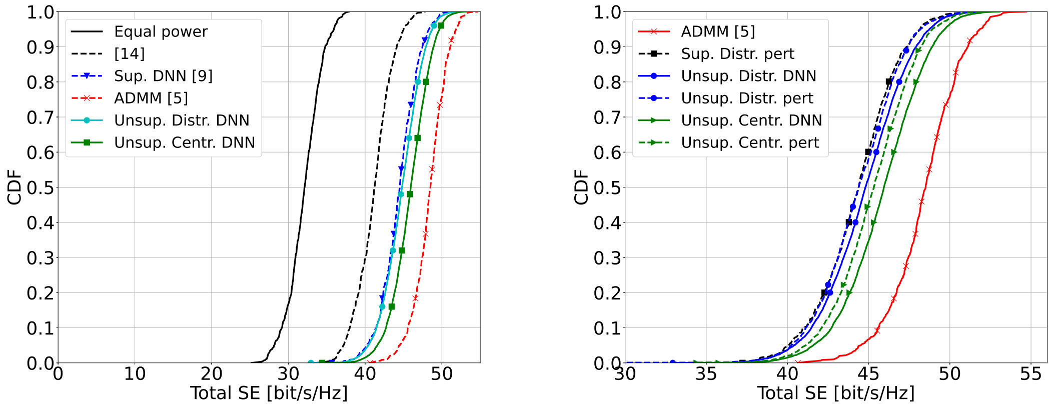
Figure 4.7: Empirical CDF of spectral efficiency per UE, comparison between the proposed DNN models using the $\mathcal{L}_{\text{sum-SE}}$ loss function with RZF precoding.

4.3.3 DNN Models Comparison

This subsection evaluates the performance of the proposed DNN models, described in section 3, by considering the CDF of the spectral efficiency per UE. In particular, the centralized, distributed, and distributed with side information DNNs are considered with RZF precoding, as illustrated in Fig. 4.7. Furthermore, as a benchmark, equal power allocation and the ADMM results are depicted as well, since the latter demonstrates a near-optimal solution for the problem. As stated in the literature [17,18], learning-based solutions have more potential for improvement when precoding schemes aimed to suppress interference are utilized, such as the RZF precoding. In fact, the latter precoding scheme allows the centralized DNN to fully exploit the network-wide LSF coefficients, thus reaching the closest result to the ADMM and providing around 10% of improvement compared to the 90%-likely spectral efficiency of the distributed DNN. Moreover, the distributed model with side information outperforms the traditional distributed DNN, leveraging the additional non-local information about the environment. Nonetheless, it does not attain the same level of approximation as the centralized solution.

4.3.4 Total Spectral Efficiency Comparison

This subsection investigates the performance of the proposed unsupervised DNN models under the total spectral efficiency metric. Indeed, Fig. 4.8a depicts the total SE of the CF-mMIMO network, obtained through many UE configurations. In particular, Fig. 4.8a shows the total spectral efficiency using MR and RZF



(a) Without perturbation in the dataset.

(b) Perturbation in the dataset.

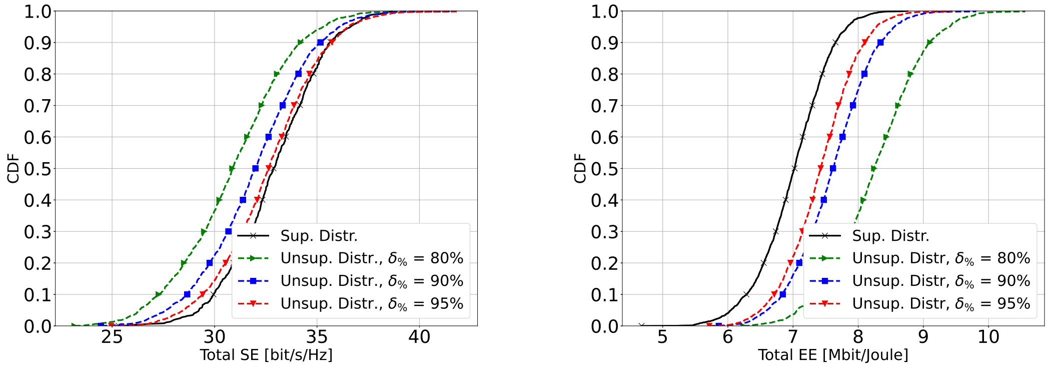
Figure 4.8: CDF of the total SE per UE under RZF precoding.

precoding schemes, respectively. As equation (2.14) is formulated to maximize the total spectral efficiency, its impact on the performance of the DNN models is evident. Notably, Fig. 4.8a demonstrates that the proposed unsupervised DNN models significantly outperform the heuristic power allocation solution [14] more than 20%, as well as the equal power distribution. In addition, even the supervised counterpart is being outperformed by our distributed and centralized DNN models, with a relative improvement of 5 % and 10 %, respectively. The quasi-optimal power allocation solution is provided by the ADMM algorithm [5], where a major gap can be noticed in the figure.

Furthermore, to prove the robustness of the unsupervised learning approach, we feed the DNN with a perturbated version of the dataset, considering a log-normal shadowing effect characterized by a standard deviation of 1 dB and truncated at 3 dB. The introduction of a small perturbation in the LSF input simulates the presence of obstacles that can be encountered in a real environment. Specifically, in Fig. 4.8b, the CDF of total spectral efficiency is depicted. It can be seen that the perturbation has a negligible effect on the distributed DNN, while degrading the performance of about 1.3 % in the case of the centralized DNN. In fact, the input vector contains all the LSF coefficients of the cell-free network. As a consequence, any variation in the LSF coefficients directly impacts the inter-relations among the AP-UE pairs.

4.3.5 AP Selection and Total Energy Efficiency

To further improve the unsupervised DNN models' performance from an energy efficiency point of view, an LSF-based AP selection is employed after the DNN power coefficients prediction. Fig. 4.9a illustrates the CDF of the total SE, re-



(a)

(b)

Figure 4.9: Effect of the AP selection on (a) total SE and (b) total EE, considering the distributed DNN with MR precoding.

Table 4.3: AP SELECTION: TRADEOFF BETWEEN PERFORMANCE AND ENERGY EFFICIENCY.

Threshold $\delta_{\%}$	**APs Serving** [%]	90%-**likely** **SE degradation**[%]
95 %	29,5 %	1,7 %
90 %	19,2 %	4 %
80 %	12,6 %	8,9 %

ferring to the distributed DNN with MR precoding and $\mathcal{L}_{\text{sum-SE}}$ loss function. To illustrate the impact of the AP selection, we showcase the difference between the supervised DNN solution and the one provided by the distributed unsupervised DNN model with the AP selection algorithm. It can be noticed a slight spectral efficiency performance degradation in Fig. 4.9a as the $\delta_{\%}$ parameter decreases. This is due to the fact that after the distributed DNN's power allocation inference, many more power coefficients are set to zero according to the AP selection process. However, the performance degradation is balanced with the system's energy efficiency. Indeed, Fig. 4.9b shows how the AP selection algorithm influences the network energy efficiency by improving it.

In particular, by selecting the APs contributing to the 95% of largest LSF coefficients (i.e. $\delta_{\%} = 0.95$), the total spectral efficiency performance in Fig. 4.9a is almost identical to the case where the AP selection is not used. However, by deactivating many power coefficients, the total energy efficiency of our proposed DNN model significantly outperforms the supervised one, as can be seen in Fig. 4.9b. To quantify the tradeoff between the total spectral efficiency performance and total energy efficiency, Tab. 4.3 indicates the average percentage of the active power coefficients and spectral efficiency degradation with different values of $\delta_{\%}$. For instance, if $\delta_{\%} = 0.95$, a UE is served by only 30% of APs on average, leading to a degradation of only $1,7\%$ compared to the use of all the APs for a given UE. It is worth noting that, in most cases, the DNN inference strongly meets the per-AP power budget constraint, with an abundance of power coefficients having near-zero values. This implies that the AP selection algorithm plays an additional role, as a subset of the closest APs is clustered to serve each UE, thus enabling the user-centric approach.

Conclusions

In this thesis, the challenging problem of the DL power allocation in a CF-mMIMO network has been tackled, harnessing the potential of deep learning, particularly unsupervised learning. Firstly, the sum-SE maximization objective is formulated, as the main metric utilized to assess the DNN models' performance. Then, centralized and distributed implementations in a CF-mMIMO have been proposed in an unsupervised learning fashion. The centralized scheme leverages a single DNN, residing at the CPU, that processes all the LSF coefficients. In contrast, the distributed approach relies on local channel statistics, with each AP equipped with its own DNN. Unlike supervised methods, the adopted unsupervised approach does not require the label generation phase, thereby circumventing the associated computational complexity. The additional degree of freedom has been exploited to define the three different custom loss functions, tailored for each optimization objective while guaranteeing the per-AP power budget constraint. After the dataset generation, which involves multi-antenna APs with MR and RZF precoding schemes, the unsupervised DNN models have been trained. Simulation results, in terms of CDF of the SE per UE and total SE, have demonstrated that the proposed unsupervised learning framework outperforms state-of-the-art optimization-based and supervised learning solutions, providing real-time performance. Finally, the use of the largest LSF-based AP selection algorithm allowed for energy efficiency and UE fairness improvement. Indeed, the AP selection algorithm makes sure that each UE is being served by the APs with the highest LSF coefficients, i.e. the closest APs.

Although the current work has yielded promising results, there remains room for improvement. Reconfigurable intelligent surfaces (RISs) are innovative surfaces featuring adaptable elements that actively control electromagnetic signals, leading to improved wireless connectivity. Future works may consider the RIS integration in a CF-mMIMO network to further increase the overall throughput or, eventually, reduce the number of APs in the system. To this end, the proposed unsupervised learning framework can be exploited to optimize the communication in a RIS-aided CF-mMIMO system.

List of Figures

List of Tables

Bibliography

[1] 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, 2019.

[2] E. Nayebi, A. Ashikhmin, T. L. Marzetta, and H. Yang, "Cell-free massive MIMO systems," in *2015 49th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2015, pp. 695–699.

[3] J. Zhang, S. Chen, Y. Lin, J. Zheng, B. Ai, and L. Hanzo, "Cell-free massive MIMO: A new next-generation paradigm," *IEEE Access*, vol. 7, pp. 99 878–99 888, 2019.

[4] S. Buzzi and A. Zappone, "Downlink power control in user-centric and cell-free massive MIMO wireless networks," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2017, pp. 1–6.

[5] S. Chakraborty, Ö. T. Demir, E. Björnson, and P. Giselsson, "Efficient downlink power allocation algorithms for cell-free massive MIMO systems," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 168–186, 2020.

[6] Ö. T. Demir, E. Björnson, L. Sanguinetti *et al.*, "Foundations of user-centric cell-free massive MIMO," *Foundations and Trends® in Signal Processing*, vol. 14, no. 3-4, pp. 162–472, 2021.

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

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

[9] M. Zaher, Ö. T. Demir, E. Björnson, and M. Petrova, “Learning-based downlink power allocation in cell-free massive MIMO systems,” *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 174–188, 2022.

[10] C. Zhang, P. Patras, and H. Haddadi, “Deep learning in mobile and wireless networking: A survey,” *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[11] Y. Zhang, J. Zhang, S. Buzzi, H. Xiao, and B. Ai, “Unsupervised deep learning for power control of cell-free massive MIMO systems,” *IEEE Transactions on Vehicular Technology*, 2023.

[12] G. Interdonato, E. Björnson, H. Quoc Ngo, P. Frenger, and E. G. Larsson, “Ubiquitous cell-free massive MIMO communications,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, pp. 1–13, 2019.

[13] H. Q. Ngo, L.-N. Tran, T. Q. Duong, M. Matthaiou, and E. G. Larsson, “On the total energy efficiency of cell-free massive MIMO,” *IEEE Trans. on Green Commun. and Net.*, vol. 2, no. 1, pp. 25–39, Nov. 2017.

[14] G. Interdonato, P. Frenger, and E. G. Larsson, “Scalability aspects of cell-free massive MIMO,” in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[15] R. Nikbakht, A. Jonsson, and A. Lozano, “Unsupervised learning for parametric optimization,” *IEEE Communications Letters*, vol. 25, no. 3, pp. 678–681, 2020.

[16] M. Fabiani, A. Abdallah, A. Celik, and A. M. Eltawil, “Unsupervised learning-based downlink power allocation for CF-mMIMO networks,” Kuala Lumpur, Malaysia, Dec. 2023, p. 6.

[17] M. Bashar, P. Xiao, R. Tafazolli, K. Cumanan, A. G. Burr, and E. Björnson, “Limited-fronthaul cell-free massive MIMO with local MMSE receiver under rician fading and phase shifts,” *IEEE Commun. Lett.*, vol. 10, no. 9, pp. 1934–1938, Jun. 2021.

[18] S. Chakraborty, E. Björnson, and L. Sanguinetti, “Centralized and distributed power allocation for max-min fairness in cell-free massive MIMO,” in *2019 53rd asilomar conference on signals, systems, and computers*. IEEE, 2019, pp. 576–580.