



**HAL**  
open science

## Self-supervised Deep Learning for mmWave Beam Steering Exploiting Sub-6 GHz Channels

Irched Chafaa, Romain Negrel, Elena Veronica Belmega, Mérouane Debbah

► **To cite this version:**

Irched Chafaa, Romain Negrel, Elena Veronica Belmega, Mérouane Debbah. Self-supervised Deep Learning for mmWave Beam Steering Exploiting Sub-6 GHz Channels. *IEEE Transactions on Wireless Communications*, 2022, 21 (10), pp.8803-8816. 10.1109/TWC.2022.3170104 . hal-03567120

**HAL Id: hal-03567120**

**<https://hal.science/hal-03567120v1>**

Submitted on 11 Feb 2022

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

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

# Self-supervised Deep Learning for mmWave Beam Steering Exploiting Sub-6 GHz Channels

Irched Chafaa, Romain Negrel, E. Veronica Belmega, *Senior Member, IEEE* and,  
Mérrouane Debbah, *Fellow, IEEE*

## Abstract

mmWave communication requires accurate and continuous beam steering to overcome the severe propagation loss and user mobility. In this paper, we leverage a self-supervised deep learning approach to exploit sub-6 GHz channels and propose a novel method to predict beamforming vectors in the mmWave band for a single *access point – user* link. This complex channel-beam mapping is learned via data issued from the *DeepMIMO* dataset. We then compare our proposed method with existing supervised deep learning and classic reinforcement learning methods. Our simulations show that choosing an appropriate beam steering method depends on the target application and is a tradeoff between data rate and computational complexity. We also investigate tuning the size of our neural network depending on the number of transmit and receive antennas at the access point. Finally, we extend our method to the case of multiple links and introduce a federated learning (FL) approach to efficiently predict their mmWave beams by sharing only the weights of the locally trained neural networks (and not the local data). We investigate both synchronous and asynchronous FL methods. Our numerical simulations show the high potential of our approach, especially when the local available data is scarce or imperfect.

## Index Terms

mmWave beamforming, deep neural networks, self-supervised learning, federated learning

I. Chafaa (irched.chafaa@ensea.fr) and E.V. Belmega (belmega@ensea.fr) are with ETIS, UMR 8051, CY Cergy Paris Université, ENSEA, CNRS, Cergy, France; I. Chafaa is also with L2S, UMR 8506, Université Paris-Saclay, CentraleSupélec, CNRS, Gif-sur-Yvette, France; R. Negrel (romain.negrel@esiee.fr) is with LIGM, UMR 8049, Université Gustave Eiffel, CNRS, ESIEE Paris, F-77454 Marne-la-Vallée, France; M. Debbah (merouane.debbah@tii.ae) is with the Technology Innovation Institute and also with the Mohamed Bin Zayed University of Artificial Intelligence, 9639 Masdar City, Abu Dhabi, United Arab Emirates. Preliminary results related to this work have been presented in [1]. [This is a preprint resubmitted to IEEE TWC on February 2022.](#)

## I. INTRODUCTION

Large antenna arrays combined with beamforming techniques represent key enablers for millimeter wave (mmWave) networks [2]. The beams allow to focus the signal energy towards the intended user and to compensate for the significant propagation losses at high frequencies [3]. Nevertheless, the transmitter's beam needs to be steered appropriately towards its user to guarantee a reliable communication link. This *beam steering* problem represents a main challenge for mmWave networks, particularly in the case of mobile users, when the beams have to be constantly adjusted. This imposes substantial training overhead and affects the overall network performance.

The beam steering problem has been studied extensively in the literature. Early approaches consist of training on a set of candidate beams following an exhaustive search [4] or an adaptive hierarchical search [5], [6] to identify the best beams in terms of a given metric (e.g., signal-to-noise ratio (SNR)). These approaches suffer from large training feedback and coordination overhead, which is incompatible with mobile mmWave applications. Other approaches estimate the channel parameters, such as the propagation path gains and angles of departure, and use them to construct beamforming vectors to steer the beams [7]–[9]. The latter works are based on compressed sensing and exploit the mmWave channel's sparsity, which reduces the training delay compared to the former ones. However, they do not scale well with large number of antennas and require prior knowledge of the channel structure and sparsity. Furthermore, they rely on static or stochastic assumptions regarding the temporal variability of the channel, which becomes problematic for non-stationary wireless environments [10].

Machine learning approaches, such as reinforcement learning (RL) and deep learning (DL), have been exploited recently to steer the beams in mmWave networks. Reinforcement learning relies on online interactions between the learning agent (access point) and its wireless environment, which results in a feedback used to determine the beam direction on-the-fly. In our previous work [10], we exploited the RL framework and specifically the adversarial multi-armed bandits (MAB) to steer the beams of both the access point (AP) and its user in a distributed manner while relying on a limited one-bit feedback. Other works [11]–[15] have employed stochastic MAB algorithms at a central entity to steer jointly the beams of both terminal nodes. The authors in [16] used Q-learning to select beams that meet quality of service requirements at the user end. In general, RL policies do not require offline training and have a relatively low

computational cost. However, they require an exploration phase to identify good beam directions (at the beginning of the communication or after significant channel variations), which may affect their performance in latency-sensitive applications [10].

Building on the wide success of neural networks, data-driven approaches have recently found their way in wireless communications [17]. In this paper, we propose to leverage a self-supervised deep learning to steer the beams of the APs towards their users in the mmWave band (downlink) using the sub-6 GHz (uplink) channel estimations. Then, federated learning is investigated to allow each AP to benefit from the locally acquired knowledge by other APs, without the need of sharing sensitive data (but only the local neural network weights). Such a collaborative approach is especially relevant when the locally available data is scarce or noisy.

*Related works:* Several existing works exploit neural networks as universal approximators to learn the relation between the wireless environment and optimal beam directions [18]–[22]. In [18], [19], a neural network is trained to exploit sub-6 GHz channel information to predict beamforming vectors at mmWave for a single transmitter-receiver link. This approach is based on a classification problem, in which the beamforming vectors are selected from a predefined discrete codebook, yielding a sub-optimal solution. Moreover, both works exploit supervised learning, which implies the need for ground truth data, i.e., a labeled input and output dataset for training. The results in [19] are also validated via real measurements with a mmWave prototype. In [20], a set of access points coordinate to serve one user by learning an appropriate beam via a supervised DL model, which exploits an omnidirectional mmWave uplink signal. The centralized nature of this approach implies heavy signaling between the different access points and a central entity, which increases the training overhead. Moreover, the omnidirectional mmWave uplink signal transmitted by a single antenna can be very limited in range and power. The authors in [21] exploit federated learning (FL) to map the mmWave channels into analog beamformers in a multi-user downlink network. Their proposed policy requires the knowledge of the mmWave channel matrices, which are more difficult to estimate and require larger training overhead compared to sub-6 GHz channels. At last, the work in [22] deals with mapping channels in space and frequency via deep neural networks, but only within the sub-6 GHz band.

*Our contributions:* In the first part of the paper, we propose a novel self-supervised deep learning method to map uplink sub-6 GHz channels into mmWave beamforming vectors. Our proposed neural network outputs directly the mmWave beamforming vectors from the sub-6 GHz uplink channels as input, by exploiting the features in the latter (e.g., second-order statistics and

other characteristics of the wireless channel) invariant at mmWave frequencies. Unlike supervised DL methods, our approach does not require ground truth data and, thus, does not require the computation of the optimal downlink beams for each channel sample in the dataset. Compared to the existing method in [18], our numerical results in the DeepMIMO setting demonstrate the superiority of our approach based on a regression and not a classification, and due to the communication-tailored loss function as opposed data-oriented proxies. We also investigate via numerical simulations how to tune the size of our neural network depending on the number of transmit and receive antennas.

We further evaluate our proposed method in the case of a *mobile user* and compare it with classical RL methods based on multi-armed bandits (MABs). Compared to MAB algorithms, DL methods provide higher communication rates at the additional cost of offline training and higher online (running) computational cost. Therefore, technical requirements and application characteristics, have to be taken into account to leverage one approach over the other.

In the second part of the paper, we extend our study to the more general case of multiple access point – user links. We propose a new federated learning scheme to predict the mmWave beamforming vectors locally, at each access point, exploiting our channel-beam mapping. Federated learning consists in the APs sharing their local models (i.e., the weights of their neural networks trained on local data) with a server, which can aggregate them into a more informed global model for all APs, leading to better performance without sharing their local datasets.

In this framework, we investigate both synchronous and asynchronous modes of uploading of the local models. The asynchronous approach allows to reduce the uploading communication cost during training and contributes to power savings as only one AP trains its neural network at each iteration. These advantages come at the cost of a certain rate degradation, which illustrates the tradeoff between *training cost vs. rate performance*.

We evaluate our proposed FL schemes and compare them to fully centralized (where the APs share their datasets with the central server) and fully distributed (no cooperation between the APs, each neural network is trained independently) benchmarks. In the case of scarce available training data, the relative rate gains of our synchronous and asynchronous FL approaches can reach up to 50% and 41% compared to the fully distributed approach, respectively. The relative rate gain reaches 14% for synchronous FL compared to centralized learning, when the quality of the training data is poor (uplink sub-6 GHz channel estimations suffer from noise). At last, we evaluate our federated learning in the case of mobile users and taking into account the downlink

multi-link interference. The results demonstrate the efficiency of our approaches in the case of scarce training data, which makes it practical for this setting.

*Positioning:* The closest works to ours are [18], [19], [21]. As opposed to [18], [19], we formulate the channel-beam mapping problem as a regression and not a classification, which implies that the predicted beams have a continuous (and, hence higher) angular resolution and overcomes the sub-optimality caused by a quantized beam codebook. Furthermore, compared to [18], [19], we design a simpler neural network architecture, with less training parameters, optimized based on a communication-tailored loss function such that the predicted beamforming vectors maximize the communication rate directly. Hence, our approach outperforms the one in [18], [19] by combining both model (via the communication rate function) and data-oriented ingredients (via the neural network) and taking advantage of both worlds. Moreover, our proposed method exploits the self-supervised learning and not supervised learning, as in [18], [19], which results in a less costly training dataset (with no ground truth) of interest when building practical datasets. Finally, compared to [21], our trained neural network requires only the available channel state information (CSI) at sub-6 GHz to predict the mmWave beams, which is much easier to acquire than the mmWave CSI and provides better beam prediction quality (since the large mmWave bandwidth results in more corrupted and noisy uplink channel information for a limited number of antennas). Thus, our proposed channel-beam mapping relieves the system from the training overhead caused by mmWave channel estimation and online beam training in [21].

Regarding our preliminary work [1], the novel contributions in this paper are four-fold and of major importance: 1) we provide an insightful performance comparison between classic MAB-based methods and deep learning methods with respect to the communication rate and computational complexity; 2) we show how to choose the size of our neural network in function of the number of transmit and receive antennas; 3) we introduce and evaluate a novel asynchronous FL method in addition to the synchronous method in [1]; 4) we evaluate our methods in realistic settings that take into account user mobility and multi-link interference aspects.

The rest of this paper is organized as follows. We present the system model and the problem formulation in Section II. In Section III, we propose a new channel-beam mapping method and evaluate its performance in the case of a single AP–user link. Then, a distributed beam steering scheme based on federated learning is introduced in Section IV and evaluated in a multi-link mmWave network. Finally, conclusions and future extensions are summarized in Section V.

## II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a wireless network composed of  $J$  access point – user links. For simplicity, we assume that each AP serves a unique user [23]. Each AP is equipped with a sub-6 GHz receive-array of  $M$  antennas (uplink) and a mmWave transmit-array of  $N$  antennas (downlink) as illustrated in Fig. 1. Each user is equipped with a single mmWave receive antenna and a single sub-6 GHz transmit antenna. The communication in each link is performed via multiple carrier frequencies, i.e., orthogonal frequency-division multiplexing (OFDM).

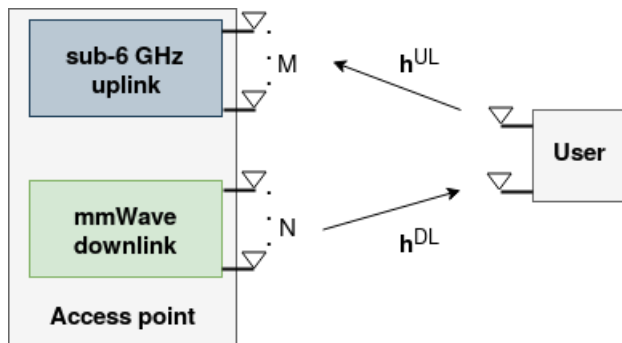


Fig. 1. An access point – user link.

We focus on a link between an AP and its intended user, under the assumption that only one link is active at a given moment or that the multi-link interference is considered as additive noise. The justification for this simplifying assumption is provided at the end of this section.

The AP aims at predicting the downlink mmWave beamforming vector for its user based on the uplink received signal at the sub-6 GHz band, which can be written in the  $\ell^{\text{th}}$  subcarrier with  $\ell \in \{1, \dots, L\}$  as follows:

$$\mathbf{y}^{\text{UL}}[\ell] = \mathbf{h}^{\text{UL}}[\ell] x^{\text{UL}}[\ell] + \mathbf{n}^{\text{UL}}[\ell], \quad (1)$$

where  $\mathbf{h}^{\text{UL}}[\ell] \in \mathbb{C}^{M \times 1}$  is the sub-6 GHz uplink channel vector;  $x^{\text{UL}}[\ell]$  is the uplink pilot symbol having an average power  $P^{\text{UL}}$  such that  $\mathbb{E}[|x^{\text{UL}}[\ell]|^2] = P^{\text{UL}}/L$ ; and  $\mathbf{n}^{\text{UL}}[\ell]$  is the additive Gaussian noise vector at sub-6 GHz.

In the downlink, an analog transceiver is used to transmit data via the mmWave antenna array. The received signal, in the  $\ell^{\text{th}}$  subcarrier, can be written as:

$$y^{\text{DL}}[\ell] = \mathbf{h}^{\text{DL}\dagger}[\ell] \mathbf{f} x^{\text{DL}}[\ell] + n^{\text{DL}}[\ell], \quad (2)$$

where  $\mathbf{h}^{\text{DL}}[\ell] \in \mathbb{C}^{N \times 1}$  is the mmWave downlink channel vector;  $\mathbf{f} \in \mathbb{C}^{N \times 1}$  is the normalized downlink beamforming vector ( $\|\mathbf{f}\|^2 = 1$ );  $x^{\text{DL}}[\ell]$  is the transmitted symbol with average power  $P^{\text{DL}}$  such that  $\mathbb{E}[|x^{\text{DL}}[\ell]|^2] = P^{\text{DL}}/L$ ; and  $n^{\text{DL}}[\ell] \sim \mathcal{N}(0, (\sigma^{\text{DL}})^2)$  is the additive Gaussian noise in the mmWave band.

The main idea of this work is to exploit the sub-6 GHz uplink channels to predict locally the downlink mmWave beamforming vector at each AP (link) using deep neural networks and federated learning. The rationale is that the uplink channels capture information regarding the wireless environment that is invariant with the frequency band (e.g., geometry of the various obstacles and buildings, higher order channel statistics, etc). This information can then be exploited to construct beamforming vectors in the mmWave band. Moreover, estimating the sub-6 GHz uplink channels requires less training overhead and exploits an already acquired technology compared to mmWave channels.

*Multi-link interference:* The uplink channel vectors  $\{\mathbf{h}^{\text{UL}}[\ell]\}_\ell$  are used in the dataset for training our neural network and also as inputs for mmWave downlink beam prediction when exploiting our neural network. In the uplink, the multi-link interference will eventually impact the quality of the estimation of the uplink sub-6GHz channels. We investigate this negative effect and show the robustness of our method to noisy or imperfect sub-6GHz channel estimations in Fig. 13 in Sec. IV-D.

Regarding the downlink, the main idea is to design a neural network taking as input only the direct sub-6 GHz channel of the served user for mmWave beam prediction and no other information regarding multi-link interference. Hence, designing a neural network that predicts the mmWave beam accounting for the downlink multi-link interference without additional input information about this interference is far from trivial. At the opposite, including such information at the input of the neural network, would require a larger dataset, a more complex network architecture, coupled with a more advanced (computationally complex) joint estimation technique of the direct channel and interference terms in the uplink.

Finally, the effect of the downlink multi-link interference on the performance of the predicted beams is expected to be limited due to the mmWave beamforming itself [24]. Otherwise stated, the performance improvement when taking the interference term explicitly into account, may not justify the additional complexity that it involves.

For all these reasons, we do not take into account the downlink multi-link interference in the design of our federated learning approach. Nevertheless, in Fig. 14 in Sec. IV-E, we show the



robustness of our approach in a multi-link setting with mobile users that do interfere with one another.

### III. CHANNEL-BEAM MAPPING VIA DL

In this section, we start by investigating the special case of a single AP – user link. We propose a novel self-supervised deep learning scheme to approximate the complex and non-linear mapping function between  $\{\mathbf{h}^{\text{UL}}[\ell]\}_{\ell=1}^L$  and the downlink mmWave beamforming vectors  $\mathbf{f}$ . Since there is no known parametric model able to capture such a relationship, data-driven approaches become essential. The used dataset contains only pairs of the channels at sub-6 GHz and mmWave and does not contain information regarding the best beams at mmWave (the ground truth), as opposed to standard supervised learning.

In the following, we explain in details the key ingredients of our proposed channel-beam mapping solution. We then evaluate its performance via extensive numerical experiments and compare it with existing methods based on classical supervised deep learning [18] and on reinforcement learning [10].

#### A. Dataset construction

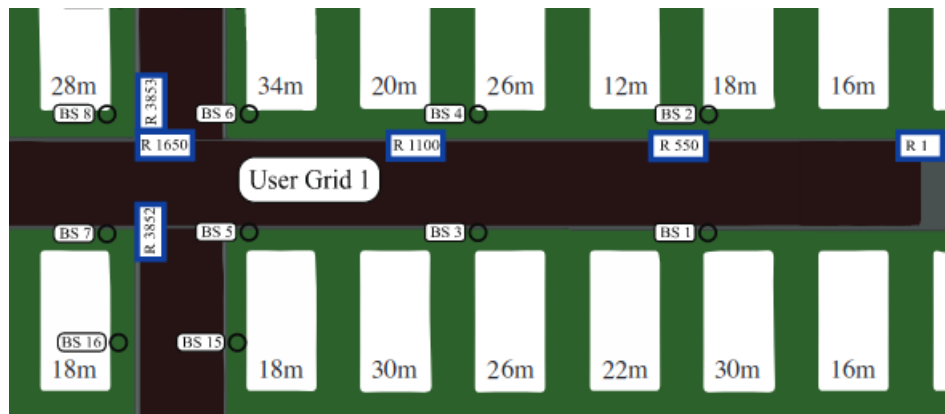


Fig. 2. Top view of the LOS scenario 'O1' of the *DeepMIMO* dataset in [25]

Our learning approach relies on the available *DeepMIMO* dataset [25] and is composed of channel pairs of the form:  $(\{\mathbf{h}[\ell]^{\text{UL}}\}_{\ell=1}^L, \{\mathbf{h}[\ell]^{\text{DL}}\}_{\ell=1}^L)$ , generated for different user positions within the predefined grid around the fixed access points. The major reasons for this choice are as follows: 1) *DeepMIMO* is an openly available dataset; 2) it employs the accurate 3D

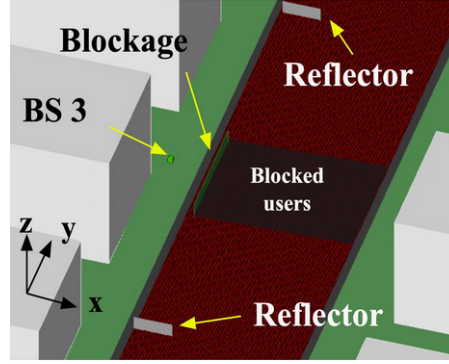


Fig. 3. The NLOS scenario of the DeepMIMO dataset [25]. The AP serves users with both LOS and NLOS propagation paths.

ray-tracing simulator *Wireless Insite* [26] to generate the uplink and downlink channels; 3) it supports various carrier frequencies in both the sub-6 GHz and mmWave bands, which fits the requirements of our problem; 4) among others, the closest work to ours [18] also employ this dataset, which enables us to compare our novel approach to that in [18].

More specifically, the uplink and downlink channels are generated using the outdoor ray-tracing scenario 'O1', which is available at 3.5 GHz and 28 GHz carrier frequencies for the LOS scenario, illustrated in Fig. 2 [25], used throughout unless specified otherwise. We also consider a new outdoor scenario as shown in Fig. 3. This scenario considers an access point (called BS3) with LOS connections with some users and NLOS connections with others, which

Parameters	uplink	downlink
AP	BS 3	BS 3
Users rows	700-1300	700-1300
Carrier frequency	3.5 GHz	28 GHz
Number of antennas	$M = 4$	$N = 64$
Antenna spacing	$\lambda/2$	$\lambda/2$
Bandwidth (GHz)	0.02	0.5
OFDM user subcarriers $L$	32	32
Number of paths	15	5
Transmit power (dBm)	–	34
Noise power (dBm/Hz)	-174	-174

TABLE I  
SYSTEM PARAMETERS.

are located in the blocked users grid.

For both scenarios, the user positions are sampled every 20 cm in the 2D row grid composed of 2751 rows and 181 columns (called User Grid 1) as shown in Fig. 2. In the single link case, we choose the third AP (called BS 3) serving a user inside its cell, placed between the rows 700 – 1300 of the grid. Unless otherwise specified, the benchmark system parameters are chosen as in Table I, similarly to the work [18] for fairness of comparison. The parameter  $\lambda$  denotes the wavelength.

Once generated, each complex entry of the channel vectors  $\mathbf{h}^{\text{UL}}[\ell]$  is decomposed into real and imaginary parts, which are stacked into a  $2LM$  real-valued vector containing the uplink channel information of all  $L$  subcarriers. This operation is done for simplicity of implementation, given that most existing deep learning libraries operate on real numbers. Similarly, the mmWave channels  $\mathbf{h}^{\text{DL}}[\ell]$  are partitioned into real and imaginary parts to form one real valued vector of dimension  $2LN$ . The obtained dataset is divided into a training dataset (80% of the total size) and a test set (the remaining 20%). The training dataset is further split into a training set (85% of its size) and a validation set (the remaining 15% of the initial training dataset). The same partition is used throughout the simulations.

Such a dataset can be built in a practical system by conducting a measurement campaign to collect and estimate both sub-6 GHz uplink and mmWave downlink channel samples for different user positions in a coverage area surrounding an access point.

Now, regarding the learning class of our proposed method: On the one hand, *standard supervised learning* relies on a training dataset that contains labeled pairs of input-output data, which in our case would amount to pairs of the type  $(\{\mathbf{h}[\ell]^{\text{UL}}\}_{\ell=1}^L, \mathbf{f})$ . On the other hand, *unsupervised learning* implies that the dataset contains only samples of inputs of the neural network, which in our case would be simply  $(\{\mathbf{h}[\ell]^{\text{UL}}\}_{\ell=1}^L)$ . Our proposed method exploits a training dataset composed of pairs of the type  $(\{\mathbf{h}[\ell]^{\text{UL}}\}_{\ell=1}^L, \{\mathbf{h}[\ell]^{\text{DL}}\}_{\ell=1}^L)$ , where the mmWave channels  $\{\mathbf{h}[\ell]^{\text{DL}}\}_{\ell=1}^L$  represent neither the inputs nor the outputs of the neural network, but an additional information to be exploited in the loss function minimized during the training process. Therefore, our method does not fit either of the two definitions.

Instead, our channel-beam mapping falls in the *self-supervised learning* paradigm [27] at the interface between supervised and unsupervised learning, where a specific output is predicted from the input but without having access to input-output labeled training data. This is precisely the case with our method where we exploit a dataset containing the pairs  $(\{\mathbf{h}[\ell]^{\text{UL}}\}_{\ell=1}^L, \{\mathbf{h}[\ell]^{\text{DL}}\}_{\ell=1}^L)$  cou-

pled with a communication-tailored loss function described below that depends on the mmWave channels  $\{\mathbf{h}[\ell]^{\text{DL}}\}_{\ell=1}^L$ , hence exploiting implicitly and autonomously the correlations in the training data.

### B. Model-based loss function

Since the main purpose is to predict *good* mmWave beams in terms of communication rate, we define a model-inspired loss function  $\mathcal{L}$ , which allows to take advantage of the known rate function in wireless OFDM systems jointly with the capability of neural networks to learn the complex and unknown channel-beam mapping function:

$$\mathcal{L} = -\frac{1}{\mathcal{B}} \sum_{i=1}^{\mathcal{B}} \mathcal{R}_i, \quad (3)$$

where  $\mathcal{B}$  is the size of the mini-batch and  $\mathcal{R}_i$  is the average data rate over the  $L$  subcarriers for the  $i^{\text{th}}$  sample:  $(\{\mathbf{h}_i[\ell]^{\text{UL}}\}_{\ell=1}^L, \{\mathbf{h}_i[\ell]^{\text{DL}}\}_{\ell=1}^L)$  of the mini-batch, and which can be written as

$$\mathcal{R}_i = \frac{1}{L} \sum_{\ell=1}^L \log_2 \left( 1 + \frac{P^{\text{DL}}}{L(\sigma^{\text{DL}})^2} |\mathbf{h}_i^{\text{DL}\dagger}[\ell] \mathbf{f}_i|^2 \right), \quad (4)$$

with  $\mathbf{f}_i$  denoting a normalized beamforming vector predicted by the neural network for the  $i^{\text{th}}$  sample (the output elements of the neural network are re-shaped into an  $N$ -dimension complex vector).

The above communication loss function, defined similarly to [28] but for a different problem<sup>1</sup>, sets our work apart from the method in [18], in which first a loss function based on some average prediction error is minimized, and then the performance of the prediction is evaluated in terms of its communication performance. Our neural network is trained and optimized to maximize directly the communication rate and skip the intermediary step. Choosing a communication-tailored loss, as opposed to a generic data-driven one as in [18], will be shown to improve the communication performance of our method via numerical results.

Further motivation is that computing a data-oriented prediction error is not possible with the available *DeepMIMO* dataset, which is only composed of channel pairs  $(\{\mathbf{h}[\ell]^{\text{UL}}\}_{\ell=1}^L, \{\mathbf{h}[\ell]^{\text{DL}}\}_{\ell=1}^L)$  and does not contain the corresponding optimal beamforming vectors  $\mathbf{f}$  (or the ground truth),

<sup>1</sup>In [28], the authors investigate deep learning methods for efficient and distributed channel estimation, quantization, feedback, and downlink multiuser precoding for a FDD massive MIMO system in which a base station (BS) serves multiple mobile users while relying on rate-limited feedback information.

differentiating our approach from a standard supervised one. Creating a different dataset composed of pairs of the type  $(\{\mathbf{h}[\ell]^{\text{UL}}\}_{\ell=1}^L, \mathbf{f})$  as in the supervised approach in [18] may be quite problematic in our regression problem because there are an infinite number of optimal beam vectors  $\mathbf{f}$  maximizing the communication rate. Indeed, the rate function above is invariant to a multiplication of  $\mathbf{f}$  by a complex scalar of unit-norm and an arbitrary selection might hinder the generalization capability of the neural network.

### C. Neural network architecture

We design a fully-connected neural network where each neuron is connected to all the neurons of the preceding and following layers as illustrated in Fig. 4. These networks are *structure agnostic*, making no particular assumptions about the inputs and serving a general purpose. Furthermore, such networks guarantee the flow of information between the inputs and outputs of each layer, which makes it able to capture any kind of dependencies between the layers (provided appropriate data and training). These characteristics make fully-connected networks suitable for our problem, since we do not have specific knowledge about the complex relationship between sub-6 GHz channels and mmWave beamformers, coupled with the different wireless parameters impacting it.

The mini-batch data is passed through a batch normalization layer to standardize its data and stabilize the learning. The uplink sub-6 GHz channel vector of dimension  $2LM$  represents the input of a deep neural network composed of 4 hidden fully-connected layers of  $S$ ,  $2S$ ,  $2S$ ,  $S$  neurons respectively with rectified linear unit (ReLU) as an activation function<sup>2</sup>. Every layer employs an L2-norm regularization with weight decay equal to  $10^{-7}$ . The output layer is a fully-connected one of size  $2N$ , which provides directly the real and imaginary parts of the mmWave beamforming vector. It is associated with an L2-normalization layer to ensure that the predicted beamforming vector is of unit norm ( $\|\mathbf{f}\|^2 = 1$ ).

### D. Evaluation of our channel-beam mapping

We evaluate here the performance of our proposed channel-beam mapping method in terms of communication rate in the LOS scenario 'O1' described in Sec. III-A, unless specified otherwise.

<sup>2</sup>This four-layer architecture and the specific number neurons  $S$  has been chosen as a result of empirical trials to best tradeoff the training and validation performance. We investigate in details the choice of  $S$  as a function of the number of antennas ( $M$  and  $N$ ) in Sec. III-E.

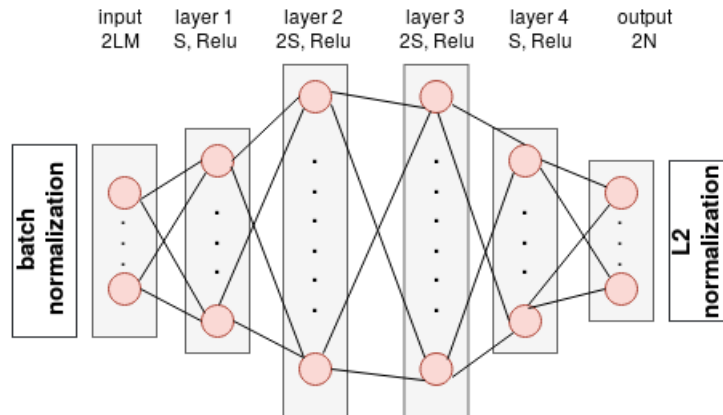


Fig. 4. Architecture diagram of the fully-connected neural network employed at the access point.

<sup>3</sup> The presented results are obtained with the neural network in Fig. 4 with  $S = 1024$  after 100 training epochs <sup>4</sup> using the adaptive moment estimation (ADAM) optimizer [29] with a learning rate of  $10^{-4}$  and a batch size of  $\mathcal{B} = 256$  samples. The different learning models are implemented and trained using *TensorFlow* [30].

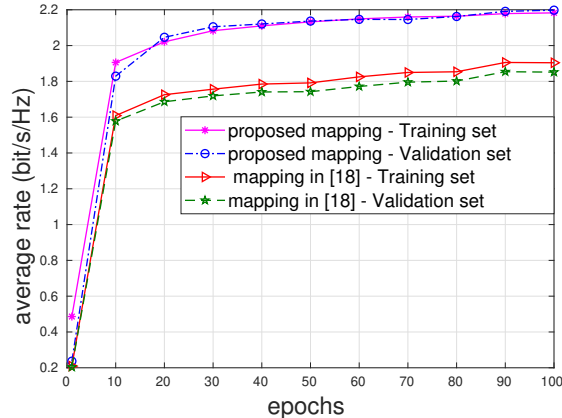
*Training and prediction performance:* In Fig. 5(a), we evaluate the communication rate, computed on the training and validation sets during 100 epochs. We can see that the average rate over the validation set follows very closely the one over the training set, which demonstrates that our method does not suffer from overfitting nor underfitting effects.

Compared to the existing method in [18], our method achieves higher rates and yields smaller gap between the training and validation rates, which indicates a better generalization performance. Indeed, combining a regression formulation with our communication-tailored loss function allows our method to have better beam prediction quality and higher rates.

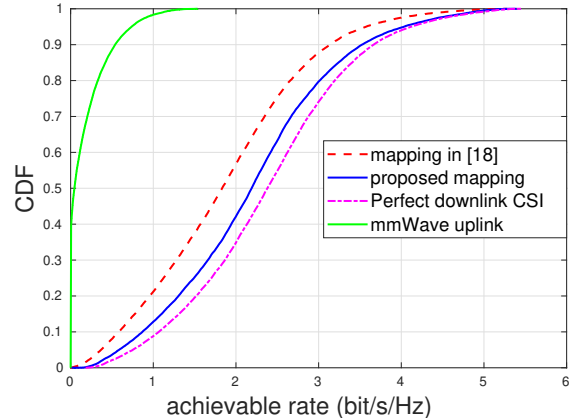
In Table II, we report the average rates obtained at the end of the training, when the weights of the neural network are optimized over the training, validation and test (disjoint to the first two) sets. The first row corresponds to scenario 'O1' and the second row will be discussed later on. The results over the test set show that our method is capable of predicting good beam directions in new channel conditions that have not been seen during training, illustrating thus

<sup>3</sup>The LOS scenario 'O1' is used for all simulation results throughout the paper except for Fig. 7(a) and Fig. 7(b), in which we show the robustness of our approach to NLOS propagation.

<sup>4</sup>An epoch is when the entire training set is passed forward and backward through the neural network once.



(a) Average rate on the training and validation sets. Our method yields higher rates and does not suffer from overfitting nor underfitting.



(b) Empirical CDF of the achievable rate over the test set. Our proposed channel-beam mapping is closer to the ideal case.

Fig. 5. Performance over the train, validation and test disjoint sets in scenario 'O1' of DeepMIMO.

good generalization capabilities. In practical settings, our deep learning method would likely require re-training using real data obtained from measurements in that setting (which is the case with any method exploiting neural networks and is not specific to this work). Nevertheless, the good generalization capability indicates that our proposed neural network architecture will not require structural changes to perform well in a new practical setting.

Moreover, In Fig. 5(b), we plot the empirical cumulative distribution function (CDF) of the average rate obtained on the test set containing samples unseen by the neural network during its training. We also compare the results with the perfect downlink CSI case, which assumes perfect and instantaneous knowledge of the mmWave channels (ideal benchmark). In addition, we evaluate the performance of our neural network trained with uplink and downlink channels in the mmWave band at 28 GHz (instead of sub-6 GHz channels in the uplink), in the same setting otherwise. First, it can be seen that our method performs better than [18] and closer to the ideal case, indicating better generalization performance, which confirms the results in Fig.

	Training set	Validation set	Test set
LOS scenario 'O1'	2.20	2.18	2.19
NLOS scenario	1.16	1.11	1.10

TABLE II  
AVERAGE RATE VALUES (BIT/S/Hz).

5(a). Second, Fig. 5(b) shows that training with uplink mmWave channels yields much lower rates (98% of the achieved rates are below 1 bit/s/Hz). This can be explained by the poor quality of mmWave uplink channels (whose difficult propagation conditions can not be overcome by a low number of 4 uplink antennas) and/or by the fact that our neural network architecture may not be suitable in this case.

*Robustness to mmWave frequency changes:* In Fig. 6, we evaluate the generalization capability (or robustness) of our trained neural network with respect to the downlink mmWave frequency. The rising question is: *can the optimized neural network trained at 28 GHz still be used to predict good beams at a different mmWave frequency?* To answer this question, we exploit two differently trained neural networks with datasets of the type (UL channels at 3.5 GHz, DL channels at 28 GHz) and (UL channels at 3.5 GHz, DL channels at 60 GHz), respectively, to predict beamforming vectors at 60 GHz, in the same setting otherwise.

Fig. 6 demonstrates that the beam predictions at 60 GHz (downlink mmWave) provide almost identical rates irrespective from the mmWave carrier frequency of the training data. The root mean square difference between the predicted beams by both approaches is around 0.11, while the difference between their achieved rates over the test set is 0.003 bit/s/Hz. This means that, once trained, our neural network can be used to predict beams at different mmWave frequencies (for the same wireless environment) with almost identical rate performance and no re-training required. Note that the rate values in Fig. 6 at 60 GHz are smaller than the ones in Fig. 5(b) at 28 GHz because the propagation loss increases for higher frequencies.

*Robustness to NLOS propagation:* We further evaluate the performance of the proposed channel-beam mapping in the second scenario described in Sec. III-A involving both LOS and NLOS users in Fig. 7.

In Fig. 7(a), we evaluate the average rate, computed on the training and validation sets during 100 epochs of training. Our proposed method exhibits a small gap between the average rates over the training and validation sets, which demonstrates that our method does not suffer from overfitting or underfitting effects in the NLOS setting and indicates as well a good generalization capability.

Indeed, the generalization capability of our method is confirmed by the average rate obtained over the test set (containing samples unseen by the neural network during its training) compared with the average rates over the training and validation sets provided in the second row of Table II. In addition, in Fig. 7(b), we plot the CDF of the rate obtained over the test set. We observe only



a small gap with the rates obtained with the ideal benchmark assuming perfect and instantaneous knowledge of the mmWave channels.

We can also notice that the obtained rates in the perfect CSI case in Fig. 7(b) are lower than those in Fig. 5(a), where all users have LOS paths to their APs. This drop in rate performance is to be expected and directly related to the propagation conditions (NLOS vs. LOS) and not inherent to our method. To sum up, all these results indicate that our proposed method performs well in both LOS or NLOS environments as it allows to predict accurately mmWave beamforming vectors for both LOS and NLOS users exploiting the sub-6 GHz channel information.

Another important issue is the robustness to channel estimation errors. So far, we have considered noiseless data for training and testing our proposed method. In Sec. IV-D, we will investigate the effects of noisy uplink channels and show the interest of our federated learning approach.

### E. Neural network design

In the previous subsection, the results were obtained for a carefully tuned neural network size:  $S = 1024$ . Here, we justify this choice. More precisely, in Fig. 8 we investigate how to tune the size  $S$  of the neural network as a function of the number of receive (uplink) and transmit (downlink) antennas:  $M$  and  $N$ . Intuitively,  $S$  has to be sufficiently large to capture the complex relationship between the inputs (sub-6 GHz channels) and outputs (mmWave beams) of the

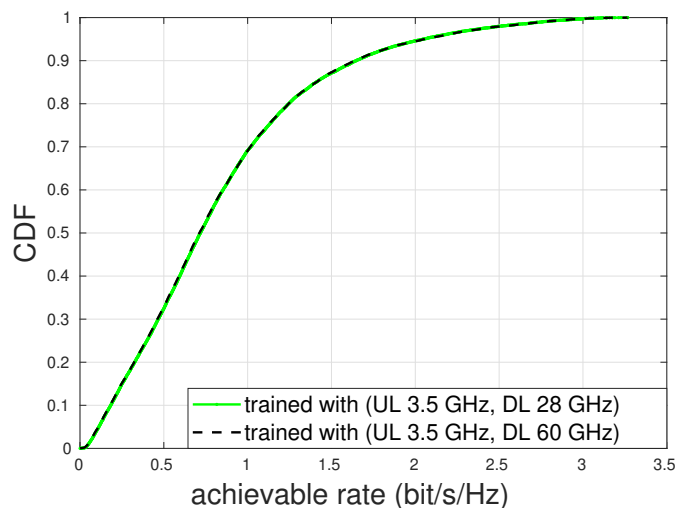


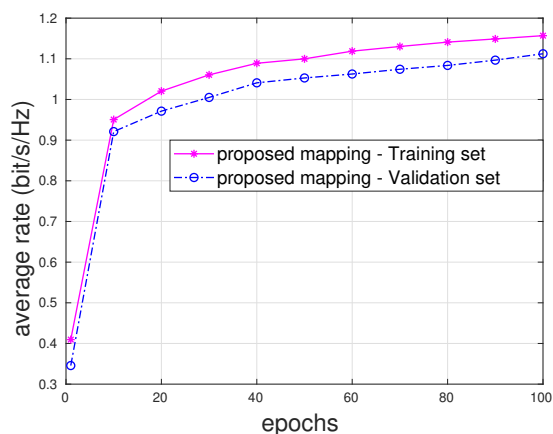
Fig. 6. Empirical CDF of the achievable rate over the test set at 60 GHz when the network is trained with downlink channel samples at both 28 GHz and 60 GHz. Our approach is robust to changes in the mmWave frequency.

neural network. However, the larger the  $S$ , the higher the computational complexity (for both the training phase and the prediction or running phase) and also the larger the training dataset has to be. The idea is to choose the smallest value of  $S$  providing the best rate performance.

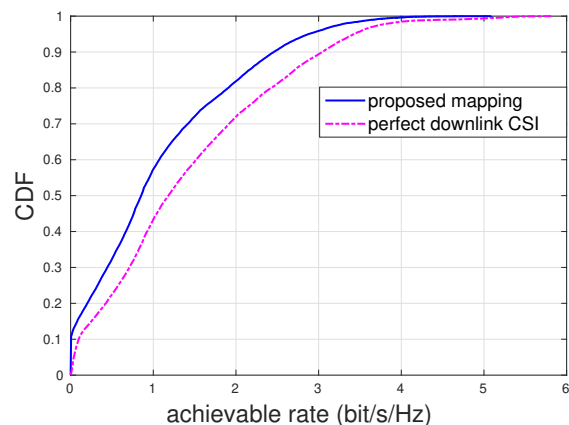
To this aim, we proceed by using the common methodology to tune the hyper-parameters of neural networks, which consists of an exhaustive search that maximizes the prediction performance over the validation set. The advantage of such a method lies in its implementation simplicity and its generality; it can be used in different settings or problems. Moreover, to the best of our knowledge, a different and optimal methodology is not yet available.

In Fig. 8(a), we evaluate the average rate over the validation set for different sizes of the neural network  $S \in [2, 2^{12}]$  and different number of uplink antennas  $M \in \{1, 4, 64\}$ , for fixed  $N = 64$  and  $L = 32$ . For any value  $M$ , the rate increases as the network size becomes larger until hitting a ceiling, from which no more significant rate improvement is observed. To further increase the rate performance, the number of uplink antennas  $M$  has to be increased to provide more information at the input of the neural network. The small gap between the rate obtained for  $M = 4$  and  $M = 64$  indicates that  $M = 4$  provides already sufficient information to reach higher rates. Increasing  $M$  further does not result in a significant rate improvement.

The network size, which is required to reach the rate ceiling, decreases with the input dimension  $M$ . Indeed, Fig. 8(a) shows that the rate ceiling is reached for  $S = 2^{10}$  when  $M = 1$ , for  $S = 2^9$  when  $M = 4$ , and for  $S = 2^8$  when  $M = 64$ . Therefore, increasing the input size  $M$

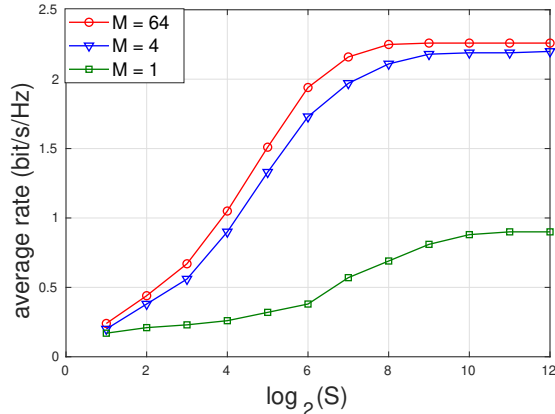


(a) Average rate on the training and validation sets. Our method does not suffer from overfitting nor underfitting.

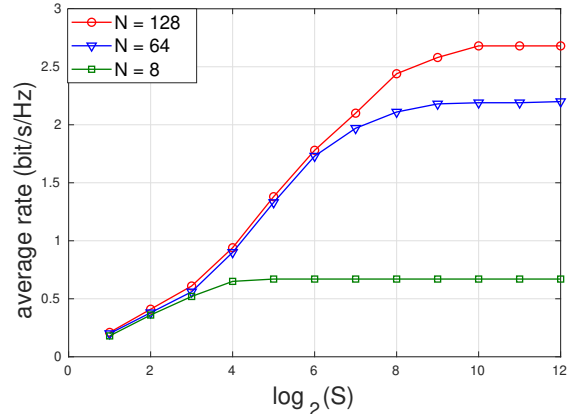


(b) Empirical CDF of the achievable rate over the test set.

Fig. 7. Performance over the train, validation and test disjoint sets in the NLOS scenario. Our proposed approach is robust to NLOS propagation as the performance results are similar to those obtained for scenario 'O1'.



(a) Impact of the input size  $M$  and the neural network size  $S$ , for  $N = 64$  and  $L = 32$ . When the input size  $M$  increases, the network size decreases.



(b) Impact of the output size  $N$  and the neural network size  $S$ , for  $M = 4$  and  $L = 32$ . When the output size  $N$  increases, the network size increases as well.

Fig. 8. Hyper-parameters tuning via empirical trials over the validation set.

provides more information at the input of the network, which results in a reduction of the size of the neural network.

In Fig. 8(b), we evaluate the average rate over the validation set for different sizes of the neural network  $S \in [2, 2^{12}]$  and different numbers of downlink antennas at the AP  $N \in \{8, 64, 128\}$ , for fixed  $M = 4$  and  $L = 32$ . The rate increases as a function of  $S$  until hitting a ceiling as in Fig. 8(a). Similarly to the above, increasing the number of transmit downlink antennas  $N$  allows to increase the beams resolution (higher beamforming gain) and, hence to achieve higher rates.

The size of the neural network should be increased with the output size  $N$ . More specifically, the rate ceiling is hit for  $S = 2^4$  when  $N = 8$ , for  $S = 2^9$  when  $N = 64$ , and for  $S = 2^{10}$  when  $N = 128$ . This is intuitive as a more complex architecture is required to capture the relationship between a fixed-size input and an increasing size output.

At last, this analysis explains the good performance of our channel-beam mapping presented in Sec. III-D for the chosen neural network size  $S = 1024$ .

#### F. Deep learning vs. multi-armed bandits

The objective here is to compare our proposed method with classic reinforcement learning, and more specifically, multi-armed bandits (MABs) in [10], [13]. For this, we consider the same evaluation scenario: a mobile user served by a fixed AP; the user's mobility follows the model in [10] with a maximum speed of 80 km/h, maximum acceleration of 2 m/s<sup>2</sup> and maximum rotation

speed of  $\pi/4$  rad/s. The initial user position is chosen randomly (via uniform distribution) and it is restricted to its AP range: rows 700 – 1300 of the 2D grid described in Sec. III-A. The channels are generated with *DeepMIMO* based on the user time-varying positions.

Aside from the deep learning method in [18] and the ideal downlink CSI case described earlier, we consider the following reinforcement learning algorithms:

**EXP3**: exponential or multiplicative weights for non-stationary (or adversarial) multi-armed bandits (MABs) [31];

**MEXP3**: an improved version of EXP3, inspired from the mmWave channel characteristics [10];

**UCB**: upper confidence bound algorithm for stochastic MABs [13].

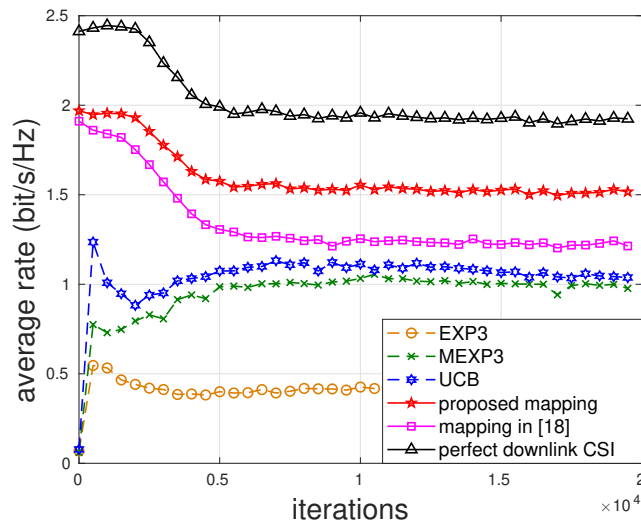


Fig. 9. Achievable rate of a mobile user. Our method outperforms the others and remains the closest to the ideal case despite the channel variations.

*Pre-training vs. learning on-the-fly*: In all MAB algorithms above, the AP computes the beams *on-the-fly* via an online process relying only on a strictly causal 1-bit (ACK/NACK) reward mechanism and do not rely on any *a priori* information about the environment. At each iteration, the AP chooses a beam from a predefined discrete codebook of size  $A$  for transmission. At the end of the transmission, the value 1 is fed back to the transmitter if the achieved rate in all carriers were above the threshold 1 bit/s/Hz<sup>5</sup>, and zero otherwise. Based on this feedback, the beam selection process is updated and a new beam is chosen. For comparison reasons, the MAB

<sup>5</sup>The value of this threshold has been chosen empirically for this specific setting as it provides a good tradeoff between *exploration vs. exploitation* and leads to good rate performance over the span of our simulation.

algorithms use the same discrete beamforming codebook as in [18] composed of  $A = 64$  beams. At the opposite, our deep learning method relies on a pre-trained and tuned neural network on the relevant DeepMIMO setting, as detailed in the previous sections.

*Feedback information:* On the one hand, the MABs algorithms requires only one bit of feedback, i.e., 1-bit (ACK/NACK) reward mechanism, at each iteration. This bit of information is then used to adapt the beam selection strategy and beam choice at the next iteration, as detailed above. On the other hand, our deep learning method requires the uplink sub-6 GHz channel information, i.e., a float vector of size  $2LM$ , and then runs the trained neural network to output the beam selection at each iteration.

*Average communication rate:* In Fig. 9, we plot the average rate obtained with the various methods as a function of iterations, where each iteration corresponds to 20 ms, i.e., the beam coherence time.<sup>6</sup> The curves are averaged for 1000 independent user trajectories. We remark that our proposed channel-beam mapping outperforms all other methods and that the deep learning approaches outperform MAB ones, obtaining up to 50 % higher rates. The rate decay of deep learning methods and of the ideal case is only related to the wireless environment and the user mobility. Indeed, the average distance between the user and its access point increases with time (over the iterations) and over the different trajectories, which explains the average rate decrease.

The gap between our method and MAB algorithms can be explained by the fact that deep learning methods have the advantage of being pre-trained and optimized for this problem, as opposed to MAB algorithms, which learn the good beams in an online and adaptive manner (via trial and error). This gap is also due to the quantization and its implied sub-optimality for MABs. Indeed, we see that the gap between the deep learning method in [18], which also suffers from this quantization, and MABs is much smaller.

At last, the best performance of UCB among the three MAB algorithms is explained by the fact that, in this particular single-link setting, there is no adversarial component and the wireless environment is stochastic, in which UCB is known to be optimal in terms of regret.

*Computational complexity:* MAB algorithms have a relatively low complexity and come with worst-case theoretical guarantees in terms of regret. As mentioned above, they are not pre-trained but learn the best beams on the fly, which implies that a certain amount of exploration

<sup>6</sup>This duration corresponds to the time interval during which the beams remain valid according to our mobility conditions [20] and the 5G NR standard [32].

time (or number of iterations) is required before achieving good performance results. Roughly speaking, UCB attains its  $\varepsilon$ -optimal performance ( $\varepsilon$  regret level) after  $1/\varepsilon$  iterations (up to logarithmic factors), whereas EXP3 and MEXP3 attains it after  $1/\varepsilon^2$  iterations. The complexity of one iteration of these algorithms scales linearly with the codebook size:  $\mathcal{O}(A)$  [10].

To characterize the complexity of our deep learning method, we focus only on the exploitation or running phase. Predicting a mmWave beam via our method has a complexity of  $\mathcal{O}(LMS + S^2 + SN)$  at each iteration.

For a more detailed complexity comparison between the two methods (DL and RL), we can focus on the number of floating point operations performed during one running iteration. For deep learning methods, the number of operations per iteration is proportional to the number of neurons in the neural network (eg. multiplication by the weights, adding bias, ...) and is about  $10^6$  operations in our network and  $10^7$  in [18]. For MAB methods, the number of operations is proportional to the codebook size  $A$  (much smaller than the number of neurons in neural networks), and is roughly around  $10^2$  operations. The number of operations during one iteration can also be used as an indicator of the energy consumption of the various methods: more operations per iteration implies more power consumption.

To sum up, the two methodologies are inherently different and, we believe that a completely fair comparison would be very difficult if not impossible to perform. This is precisely why we have presented various aspects of comparison and not just the rate performance in the running phase, which would be unfair.

Having said that, we observe that deep learning methods outperform classical reinforcement learning ones in terms of communication rate (up to 50% higher rates) at the cost of being more computationally complex ( $10^4$  times more complex) and of requiring offline training based on relevant and sufficient data. On the flip side, reinforcement learning and MABs do not require offline training, they rely on simple feedback information and low-cost iterations, and also come with performance guarantees in terms of regret. Nevertheless, the explosion of deep learning applications can be explained by the advances of the electronics industry and the exponential growth of Internet traffic resulting in new devices (at the edge of the network) with high processing capabilities and large amounts of available data. Choosing the appropriate solution will depend on the target application and on its characteristics and requirements in terms of energy consumption, processing capabilities, data availability, latency, etc.

#### IV. MULTI-LINK BEAM STEERING VIA FEDERATED LEARNING

In this section, we investigate the more general case of multiple AP–user links. There are several ways in which our channel-beam method derived in Sec. III can be extended. One naive idea would be for each AP to perform individual training of their neural network based on locally available data and independently from one another. Such a fully distributed approach might perform quite poorly when local data is scarce and when cooperation would be beneficial.

At the opposite, a centralized approach would require the APs to send all their training data to a central node, which performs the neural network training and then sends the resulting neural network parameters back to each AP. This raises several issues in terms of data privacy and communication overhead.

In this section, we propose a different learning approach in between these two extremes by exploiting federated learning [33]. In this framework, the APs send to a server only their neural network parameters and not their actual datasets, which improves the users’ privacy: exchanging channel data with a central server may lead to leaking critical information such as users’ location, mobility patterns, etc. Moreover, federated learning also enables parallel computation, which speeds up the training process by splitting the computational load among the different APs. Finally, compared to the fully distributed approach, the server is able to compute a global and more informed model (based on the knowledge acquired by all APs), which is then fed back to the APs, leading to higher local prediction performance.

Our FL approach (illustrated in Fig. 10) consists of a training phase and an exploitation phase, which are detailed hereafter.

##### A. *FL training*

In federated learning, the training phase is performed collaboratively and two different methods are proposed here: the synchronous FL (SFL) and asynchronous FL (AFL). In SFL, a training epoch corresponds to a local epoch at every AP, whereas in AFL, it corresponds to a local epoch at only one AP (the updating AP). In the centralized or individual approaches, a training epoch is reached when all the batches (or all training samples) have been visited by the loss minimization method.

In SFL, all APs optimize locally their neural network based on their available data as detailed in Sec. III, at each training epoch. An epoch is reached when the loss minimization method at each AP has visited all its available local samples. Then, the parameters or weights of the local

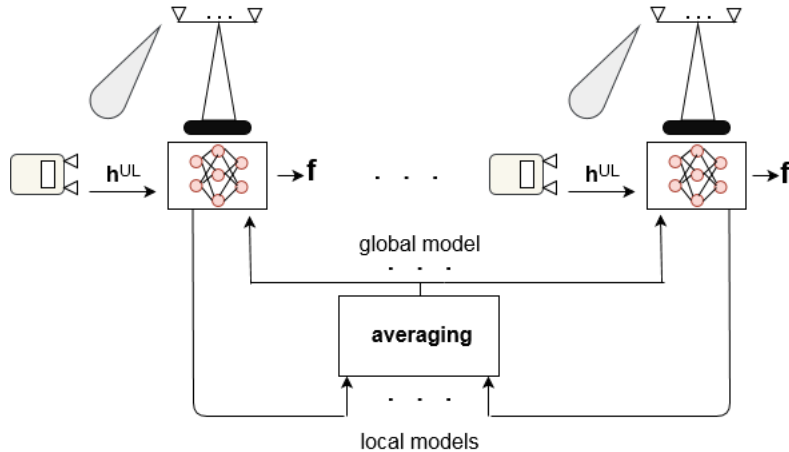


Fig. 10. Federated learning for beam prediction based on sub-6 GHz channels.

networks are uploaded to a server for aggregation (a simple average operation), as illustrated in Fig. 10. On the contrary, in AFL, the APs take turns and only one AP optimizes its neural network based on their local available data as in Sec. III at each training epoch. This AP then uploads its optimized network parameters to the server, which are then averaged with the previous global network parameters to obtain the new network update. For both FL methods, the updated global network parameters are downloaded by each AP and the process repeats until the end of the training phase.

Compared with SFL, the AFL method reduces the communication overhead at each training epoch at the cost of having to perform more training epochs to reach the same performance.

### B. FL exploitation

After the training phase, the final global neural network is exploited by all the APs locally. Each AP runs this trained network to predict the mmWave beamforming vectors by feeding it with locally estimated sub-6 GHz uplink channels.

During the exploitation, the APs can collect new data to refine and update the global model in case of major changes in the wireless environment.

### C. Dataset construction

Unless stated otherwise, the system parameters in the *DeepMIMO* dataset are the same as in Table I. We consider a system of 4 AP–user links in the 'O1 scenario represented in Fig. 2.



Following the nomenclature in *DeepMIMO*, the chosen access points are: BS 1, BS 4, BS 6, and BS 7; their users are positioned within their specific cells characterized by the grid rows: 1 – 599, 600 – 1200, 1201 – 1550 and 1551 – 2200, respectively.

#### D. Evaluation of our FL approach

In the rest of the paper, we evaluate the performance of our proposed distributed mmWave beam steering in terms of overall network average rate, which is computed as the average of the rates of all AP–user links.

We start by evaluating the average rate of our federated learning approach for both synchronous (SFL) and asynchronous (AFL) updates of the global model. We compare both methods to the following benchmarks.

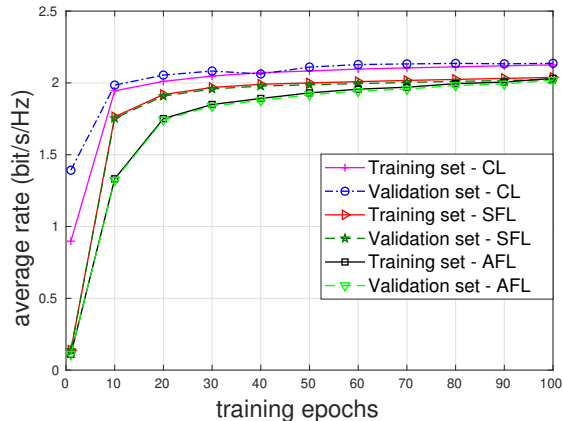
**Centralized learning (CL):** performed by a central authority using a neural network trained on the global dataset including all local datasets of the APs;

**Perfect downlink CSI:** an ideal scenario with a perfect and instantaneous knowledge of the downlink mmWave channels, which are used to construct the beamforming vectors for each subcarrier such that  $\mathbf{f}^*[\ell] = \mathbf{h}^{\text{DL}}[\ell]/\|\mathbf{h}^{\text{DL}}[\ell]\|$  to maximize the received power at the receiver;

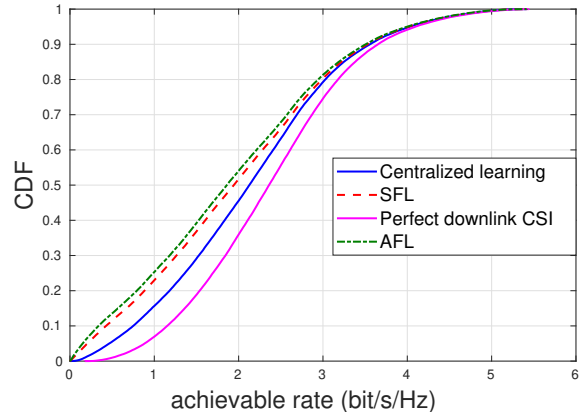
**Individual learning (IL):** each AP trains its neural network using its local dataset independently from the others (fully distributed learning).

*FL training and prediction performance:* In Fig. 11(a), we illustrate the evolution of the average rate computed on the training and validation sets for CL, SFL and AFL methods. First, notice that the values of the average rate on the training and validation sets are close for all methods, which implies that the proposed neural network performs well without data underfitting or overfitting and after only 20 training epochs. The rates achieved by the FL schemes approach the centralized performance. As expected, AFL method is a little slower than SFL, but reaches eventually comparable performance. AFL reduces the neural network parameter exchange between APs and the server by 75% (only one out of the four APs performs local training and sends the network parameters at each training epoch) at the cost of a relatively small performance loss, highlighting the *signaling vs. performance tradeoff*. Aside from simplifying the training, AFL also reduces the power consumption by 75%.

Fig. 11(b) represents the empirical cumulative distribution function (CDF) of the average rate over the test sets for the CL, AFL and SFL methods and also for the perfect downlink CSI



(a) Average rate evaluated on the training and the validation sets. AFL and SFL rates perform close to centralized learning.



(b) Empirical CDF of the achievable rate over the test set. Our FL methods perform close to the centralized learning.

Fig. 11. Performance of our proposed FL methods over the train, validation and test disjoint sets in scenario 'O1' of DeepMIMO.

beams. For each learning method, the already trained neural networks are exploited to predict downlink mmWave beamforming vectors from uplink sub-6 GHz channels.

We remark that both our FL methods (AFL and SFL) perform close to CL and are not far from the ideal performance, indicating a good generalization performance on unseen test data. This means that, our proposed FL methods can provide almost the same average rate performance as CL, while maintaining the advantages of the FL framework. Moreover, the AFL method performs almost as good as SFL after the same number of training epochs, while enjoying 75% less signaling overhead and power consumption during the training process.

*Robustness to scarce data:* In Fig. 12, we evaluate the average rate over the test sets for different training set sizes relative to the total training set size of each access point. We also evaluate here the individual learning scheme. First, increasing the training set size improves the average rate for all schemes, because of the better generalization performance. When the local available data is scarce, our FL methods clearly outperforms individual learning and reaches by up to 50 % higher rates with SFL and 41% with AFL, showing the interest of sharing the network parameters in such cases. When the amount of local available data is sufficiently large, federated learning does not bring any advantage and the parameter averaging operation (performed at the central node during the training phase) leads to a sub-optimal performance compared with individual learning.

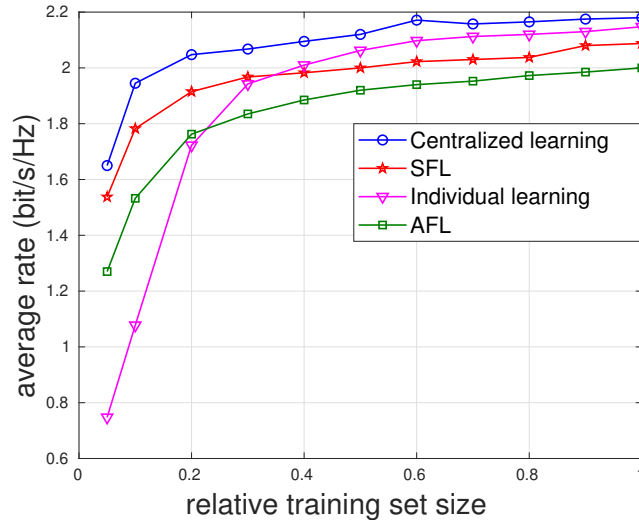


Fig. 12. Impact of the training set size. For scarce training data, our FL schemes outperform the individual learning and approach centralized learning.

*Robustness to imperfect sub-6GHz channel estimation:* In order to test the robustness of our FL schemes to imperfect data, we train the various methods when the uplink channels are contaminated with different levels of noise and then we test the resulting neural networks on unseen input data. We assume that the sub-6 GHz channel estimator is unbiased and that the estimation error is an additive Gaussian random variable of finite variance, which is a common and reasonable assumption [5], [34].

Fig. 13 illustrates the evolution of the average rate over the test sets as a function of different noise variance levels (in between  $10^{-11}$  and  $10^{-8}$ ) corrupting the uplink sub-6 GHz channel estimations (at the input of neural networks) during both the test and training phases. The empirical variance of the uplink sub-6 GHz channels is of around  $10^{-9}$ . The SFL scheme outperforms centralized and individual learning at high noise levels when the quality of the uplink sub-6 GHz channel estimations is poor. For a noise variance of  $10^{-8}$ , the relative gain is of 14% compared to centralized learning. This result can be explained by the averaging step of the SFL scheme, which acts as a regularization and noise smoothing operation, thus improving its generalization performance in this regime. The same advantage is not observed with AFL, as it does not average all local models at every training epoch (no lower noise smoothing effect).

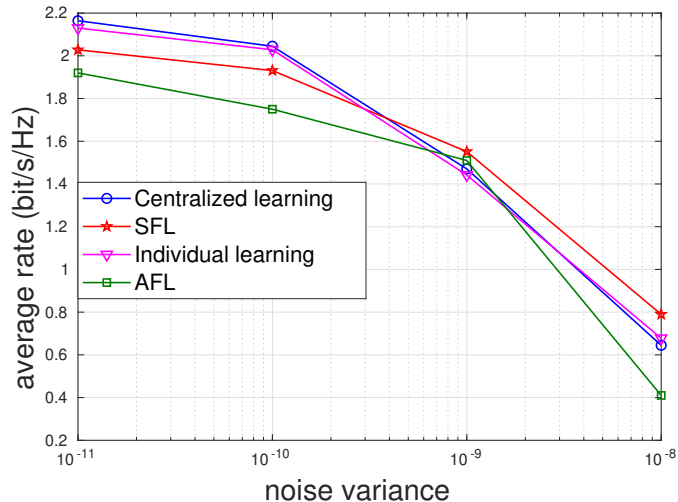


Fig. 13. Impact of noisy training and test data (imperfect sub-6 GHz channel estimation). Our SFL scheme outperforms centralized and individual learning in the high noise regime.

### E. Interfering multiple links and user mobility

In the following, we evaluate our FL approaches in the presence of downlink multi-link interference and user mobility. The downlink interference for an AP – user link is defined as the mmWave signals received from other APs. The mobility of the four users is modeled as in Sec. III-F. The APs exploit similar neural networks, which are trained offline, having little available data: only 10% of the training set, to predict their beams during each time slot of 20 ms. Note that the offline training phase has been performed without taking into account the downlink inter-link interference as discussed in Sec. II. The results are compared to the fully centralized, individual learning and MAB approaches from Sec. III and are illustrated in Fig. 14. The rate of the perfect downlink CSI curve in Fig. 14 is evaluated in an ideal scenario without multi-link interference representing a performance upper bound. The curves are averaged over 1000 random and independent trajectories for each user.

We can see that the SFL method outperforms individual learning, the MAB-based approaches and is close to the centralized one when the training is done with little data (10% of the available training set) even in the presence of downlink interference. This illustrates the utility of the SFL method to improve the rate performance when the available training data is scarce and when sharing the local model with other APs is beneficial.

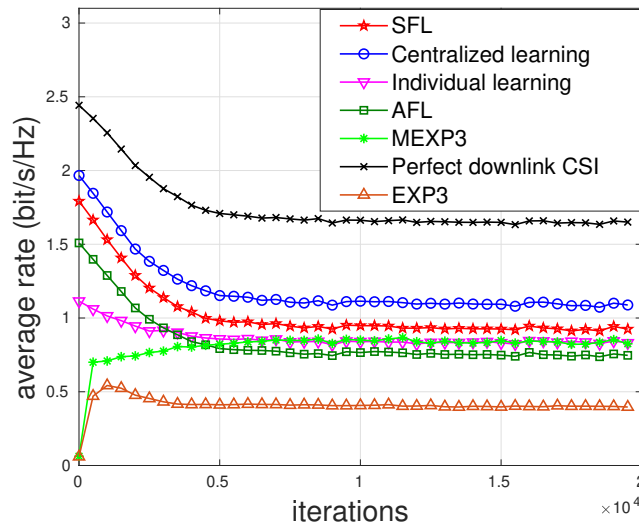


Fig. 14. When the available data is scarce, SFL outperforms the individual learning and reduces the rate gap with centralized learning.

## V. CONCLUSIONS AND PERSPECTIVES

In this paper, we address the beam steering problem in mmWave communications. First, we propose a self-supervised deep learning method to design a novel channel-beam mapping exploiting the channel knowledge at sub-6 GHz to predict mmWave beamforming vectors. The performance of the proposed mapping is evaluated and compared to existing methods based on supervised deep learning or on classic reinforcement learning and multi-armed bandits (MABs) in the DeepMIMO setting. Our self-supervised method outperforms existing methods based on supervised deep learning, by combining a regression idea with a communication-tailored loss function. Compared to MAB algorithms, our deep learning method performs better in terms of rate at the cost of being more complex in the running phase and of requiring offline training on relevant and available data.

In the second part of the paper, we extend our proposed mapping to a federated learning framework used to predict the beams of multiple access point (AP) – user links. Federated learning consists in the APs sharing the weights of their locally trained neural networks with a server to improve their prediction performance, while preserving their datasets locally. Our numerical results demonstrate the high potential of federated learning in the case of scarce and corrupted available training data (at each AP) compared to fully centralized (when sharing their datasets) and fully distributed learning (no sharing).

Interesting future perspective include: taking into account the multi-link interference in the neural network design, architecture and training; exploiting deep reinforcement learning techniques (that combine the advantage of modeling complex relationships of deep learning with the online adaptability of reinforcement learning) or recurrent neural networks to explicitly take into account the user mobility and channel temporal dynamics; considering multiple antennas and beamforming at the user end; considering hybrid (analog and digital) beamformers, etc.

## REFERENCES

- [1] I. Chafaa, R. Negrel, E. V. Belmega, and M. Debbah, "Federated channel-beam mapping: from sub-6 GHz to mmWave," in *IEEE WCNC 2021, Workshop on Distributed Machine Learning*, Mar. 2021.
- [2] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.
- [3] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter wave cellular wireless networks: Potentials and challenges," *arXiv preprint arXiv:1401.2560*, 2014.
- [4] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE J. Sel. Areas Commun*, vol. 32, no. 6, pp. 1164–1179, 2014.
- [5] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. Sel. Topics Signal Process*, vol. 8, no. 5, pp. 831–846, 2014.
- [6] "Wireless lan medium access control (MAC) and physical layer (PHY) specifications amendment 3: Enhancements for very high throughput in the 60 GHz band," *IEEE 802.11 working group and others, IEEE Computer Society*, 2012.
- [7] D. E. Berraki, S. M. Armour, and A. R. Nix, "Application of compressive sensing in sparse spatial channel recovery for beamforming in mmWave outdoor systems," in *IEEE WCNC*, 2014, pp. 887–892.
- [8] J. Choi, "Beam selection in mmWave multiuser MIMO systems using compressive sensing," *IEEE Trans. Commun*, vol. 63, no. 8, pp. 2936–2947, 2015.
- [9] N. González-Prelcic, A. Ali, V. Va, and R. W. Heath, "Millimeter-wave communication with out-of-band information," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 140–146, 2017.
- [10] I. Chafaa, E. V. Belmega, and M. Debbah, "One-bit feedback exponential learning for beam alignment in mobile mmwave," *IEEE Access*, vol. 8, pp. 194 575–194 589, 2020.
- [11] M. Hashemi, A. Sabharwal, C. E. Koksall, and N. B. Shroff, "Efficient beam alignment in millimeter wave systems using contextual bandits," *arXiv preprint arXiv:1712.00702*, 2017.
- [12] A. Asadi, S. Müller, G. H. A. Sim, A. Klein, and M. Hollick, "FML: Fast Machine Learning for 5G mmWave Vehicular Communications," in *IEEE INFOCOM*, 2018, pp. 1961–1969.
- [13] J.-B. Wang, M. Cheng, J.-Y. Wang, M. Lin, Y. Wu, H. Zhu, and J. Wang, "Bandit inspired beam searching scheme for mmWave high-speed train communications," *arXiv preprint arXiv:1810.06150*, 2018.
- [14] V. Va, T. Shimizu, G. Bansal, and R. W. Heath Jr, "Online learning for position-aided millimeter wave beam training," *arXiv preprint arXiv:1809.03014*, 2018.
- [15] W. Wu, N. Cheng, N. Zhang, P. Yang, W. Zhuang *et al.*, "Fast mmWave beam alignment via correlated bandit learning," *arXiv preprint arXiv:1909.03313*, 2019.
- [16] R. Wang, O. Onireti, L. Zhang, M. A. Imran, G. Ren, J. Qiu, and T. Tian, "Reinforcement learning method for beam management in millimeter-wave networks," in *2019 UK/China Emerging Technologies (UCET)*. IEEE, 2019, pp. 1–4.

- [17] A. Zappone, M. Di Renzo, and M. Debbah, “Wireless networks design in the era of deep learning: Model-based, AI-based, or both?” *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, 2019.
- [18] M. Alrabeiah and A. Alkhateeb, “Deep learning for mmwave beam and blockage prediction using sub-6 GHz channels,” *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5504–5518, 2020.
- [19] M. S. Sim, Y.-G. Lim, S. H. Park, L. Dai, and C.-B. Chae, “Deep learning-based mmwave beam selection for 5G NR/6G with sub-6 GHz channel information: Algorithms and prototype validation,” *IEEE Access*, vol. 8, pp. 51 634–51 646, 2020.
- [20] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, “Deep learning coordinated beamforming for highly-mobile millimeter wave systems,” *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.
- [21] A. M. Elbir and S. Coleri, “Federated learning for hybrid beamforming in mm-wave massive mimo,” *IEEE Communications Letters*, vol. 24, no. 12, pp. 2795–2799, 2020.
- [22] M. Alrabeiah and A. Alkhateeb, “Deep learning for TDD and FDD massive MIMO: Mapping channels in space and frequency,” in *53rd IEEE ACSSC*, 2019, pp. 1465–1470.
- [23] K. Shen and W. Yu, “Fractional programming for communication systems — part I: Power control and beamforming,” *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, 2018.
- [24] H. Zhao, R. Mayzus, S. Sun, M. Samimi, J. K. Schulz, Y. Azar, K. Wang, G. N. Wong, F. Gutierrez, and T. S. Rappaport, “28 GHz millimeter wave cellular communication measurements for reflection and penetration loss in and around buildings in New York city,” in *2013 IEEE international conference on communications (ICC)*. IEEE, 2013, pp. 5163–5167.
- [25] A. Alkhateeb, “DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications,” *arXiv preprint arXiv:1902.06435*, 2019.
- [26] Remcom, “Wireless Insite,” <http://www.remcom.com/wireless-insite>.
- [27] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, “Self-supervised learning: Generative or contrastive,” *IEEE Trans. on Knowledge and Data Engineering*, 2021.
- [28] F. Sahrabi, K. M. Attiah, and W. Yu, “Deep learning for distributed channel feedback and multiuser precoding in FDD massive MIMO,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4044–4057, 2021.
- [29] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [30] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th OSDI*, 2016, pp. 265–283.
- [31] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, “Gambling in a rigged casino: The adversarial multi-armed bandit problem,” in *IEEE FOCS*, 1995, pp. 322–331.
- [32] 3GPP, “NR; physical channels and modulation (release 15),” *3rd Generation Partnership Project (3GPP)*, vol. 38.211, p. V15.2.0, 2018.
- [33] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, “Federated learning,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.
- [34] Y. Chen and C. Tellambura, “Performance analysis of maximum ratio transmission with imperfect channel estimation,” *IEEE Communications Letters*, vol. 9, no. 4, pp. 322–324, 2005.