Deep Learning for Distributed Channel Feedback and Multiuser Precoding in FDD Massive MIMO

Foad Sohrabi, *Member, IEEE,* Kareem M. Attiah, *Student Member, IEEE,* and Wei Yu, *Fellow, IEEE*

Abstract—**This paper shows that deep neural network (DNN) can be used for efficient and distributed channel estimation, quantization, feedback, and downlink multiuser precoding for a frequency-division duplex massive multiple-input multiple-output system in which a base station (BS) serves multiple mobile users, but with rate-limited feedback from the users to the BS. A key observation is that the multiuser channel estimation and feedback problem can be thought of as a distributed source coding problem. In contrast to the traditional approach where the channel state information (CSI) is estimated and quantized at each user independently, this paper shows that a joint design of pilot and a new DNN architecture that maps the received pilots directly into feedback bits at the user side, then maps the feedback bits from all the users directly into the precoding matrix at the BS, can significantly improve the overall performance. This paper further proposes robust design strategies with respect to channel parameters and also generalizable DNN architecture for varying number of users and number of feedback bits. Numerical results show that the DNN-based approach with short pilot sequences and very limited feedback overhead can already approach the performance of conventional linear precoding schemes with full CSI.**

Index Terms—**Channel estimation, deep neural network (DNN), distributed source coding (DSC), downlink precoding, feedback frequency-division duplex (FDD), massive multiple-input multiple-output (MIMO), quantization.**

I. INTRODUCTION

Machine learning methods, especially deep neural networks (DNNs), have recently shown great potential in dealing with complex optimization problems in various wireless communications settings, e.g., MIMO detection [1], massive multiple-input multiple-output (MIMO) hybrid precoding [2], constellation design [3], [4], user scheduling [5], etc. The data-driven approach has an advantage from both performance and complexity perspectives, especially when the models are uncertain and when conventional optimization approaches have high complexity. This paper aims to show the effectiveness of the deep learning framework for obtaining optimized feedback and beamforming strategies for the frequency-division duplex (FDD) massive MIMO system design. The key observation of this paper is that the multiuser channel estimation, quantization and feedback problem can be thought of as a *distributed source coding (DSC)* problem for which the optimal solution is analytically intractable, but where the machine learning approach can have an important advantage.

Massive MIMO is an indispensable technology for addressing the ever-increasing demand for data rate in the next generations wireless networks [6]. The canonical massive MIMO

system operates in the time-division duplex (TDD) mode in which the BS can exploit the uplink-downlink channel reciprocity via uplink training to acquire channel state information (CSI) and subsequently use the CSI for downlink precoding [7]. However, many existing wireless networks operate in FDD mode in which uplink-downlink channel reciprocity cannot be assumed. For these FDD systems, the channels need to be estimated in the downlink, then fed back to the BS for precoding. The question of how to optimally design such a channel feedback strategy is crucial for the wide adoption of FDD massive MIMO in wireless networks [8].

This paper focuses on a multiuser FDD downlink massive MIMO system in a limited-scattering environment, e.g., in millimeter wave (mmWave) band [9], where sparsity can be exploited for channel estimation. But instead of considering channel estimation as a standalone module, we consider the end-to-end system including the design of the downlink pilot, the channel estimation and quantization strategy under limited feedback, as well as the design of the downlink precoding matrix at the BS. We make a crucial observation that because downlink channel estimation and quantization take place in a distributed fashion across the users, yet are fed back to a centralized location at the BS for precoding purpose, the overall system is akin to a DSC scheme. While traditional wireless system design always performs independent quantization of each user's channel and never takes the distributed nature of channel quantization into consideration, this paper shows that:

- Optimized distributed channel compression strategy can significantly outperform the conventional independent CSI estimation and feedback scheme;
- Joint design of channel feedback and precoding has significant advantage;
- Deep neural network (DNN) can play a crucial role for the effective design of such a joint precoding and distributed channel compression strategy.

The information theoretic considerations of DSC have appeared in the literature since 1970s. The celebrated Slepian-Wolf Theorem shows that optimal lossless DSC of two or more correlated sources with separate encoders and a joint decoder can be much more efficient than independent encoding/decoding [10]. Their strategy of using binning to take advantage of the correlation between sources is further extended to lossy compression by Wyner and Ziv [11]. While many results in DSC aim to recover the correlated sources at the decoder [12], [13], the concept of DSC can also be extended to computing a function of the sources [14]–[16]. For example, [14] characterizes the rate region for distributed compression of two correlated uniform binary sources for

computing their modulo-2 sum. It is important to note that the benefit of the DSC strategy can come from exploiting not only the correlation between the sources, but also the fact that a function of the sources (rather than the sources themselves) is desired at the decoder. For example, [15] examines the distributed source coding problem for classification and [16] provides examples in which the benefit of DSC comes from designing separate encoders and joint decoder that exploit a match between the quantization scheme and the function to be computed. Such benefit exists even for the case where the sources are independent.

A key insight of this paper is the recognition that the end-to-end design of a downlink FDD precoding system can be regarded as a DSC problem of computing a function (i.e., the downlink precoding matrix) of independent sources (i.e., channels) under finite feedback rate constraints. This paper makes the case that by designing an optimized DSC strategy, we can significantly reduce the amount of feedback as compared to the conventional design based on separate CSI estimation and feedback of each user. The design of the optimal DSC strategy is, however, a difficult problem in general, even for the case in which the source distributions are completely known and the blocklength is large. This motivates us to use the deep learning technique to tackle such an optimization problem.

The deep learning framework is well suited to tackle the DSC design problem because of the following. First, as different from the convectional communications system design methodology which optimizes different blocks of a communication system separately, the deep learning framework can jointly design all the components for end-to-end performance optimization, making it suitable for designing a DSC strategy especially when the goal is to compute a function of the sources. Second, unlike the classical DSC which requires customized source coding design for each different scenario, the deep learning framework implicitly learns the channel distributions in a data-driven fashion in the process of optimizing the end-to-end communications system, without requiring tractable mathematical channel models. Third, computation using trained DNN can be highly parallelized, so that the computational burden of DNN is manageable.

A. Main Contributions

This paper shows that the end-to-end downlink precoding design problem for an FDD massive MIMO system can be viewed as a DSC problem. More specifically, the channels from the BS to the different users can be considered as the "sources", and the objective of the DSC scheme is to recover a function of these sources, namely an optimal precoder at the BS that maximizes a system objective, e.g., the sum rate of all the users, based on the rate-limited feedback. To tackle this challenging DSC design problem, this paper proposes a data-driven approach wherein the channel estimation, distributed compression, feedback at the user side, and multiuser precoding at the BS side can be efficiently designed jointly by training a DNN at each user and a DNN at the BS. In particular, we propose a novel neural network architecture that encapsulates all the components of the FDD downlink

precoding system. By properly training the proposed neural network, we jointly optimize the downlink pilots and channel estimation process, the uplink channel quantization and rate-limited feedback scheme at each user, and the multiuser downlink precoding scheme at the BS for each user.

The training of the proposed DNN is however also challenging. In the proposed DNN architecture, the feedback information bits are modeled as the outputs of binary neurons. Without any adjustments to the conventional *back-propagation* algorithm, these binary neurons would have vanishing gradients during training, which inhibit the training of network parameters. To overcome this issue, we alternatively approximate the gradients of the binary layer with a variant of *straight-through* (ST) estimator [17]–[19]. In the numerical experiments, the proposed neural network architecture trained with the ST approximation technique exhibits outstanding performance, especially when the channel training and feedback are severely rate limited. These results confirm that the deep learning framework can indeed be utilized to obtain near-optimal, yet practical, DSC strategies which are in general hard to achieve using convectional heuristic methods. Further, the results show that the performance gain comes from the overall joint DSC and precoding design that bypasses individual explicit channel estimation at each user.

While the proposed DNN has excellent performance when the DNN is trained and tested under the same environment, one of the main questions about any data-driven approach, including the proposed DNN, is whether or not its performance is generalizable to unseen system environments. Towards addressing this concern, this paper discusses in detail how to make the proposed architecture generalizable for different system parameters. In general, we categorize the system parameters into two categories: i) the system parameters that only change the input distribution of the DNN, e.g., the number of paths in the sparse channel model and the signal-to-noise ratio (SNR), and ii) the system parameters that also change the input/output dimensions of some layers in the proposed DNN, e.g., the feedback rate limits and the number of users.

For the first category of the system parameters, we numerically show that training the DNN on a wider range of system parameters can help design more robust systems when prior knowledge about these parameters is not available. For the second category, however, training a DNN that can operate for different system dimensions is more challenging. In this regard, this paper proposes to modify the neural network architecture and its training procedure as a way to enhance the generalizability of the proposed DNN with respect to the feedback rate limits as well as the number of users as follows. We propose a novel two-step training approach to design a common neural network that can operate over a wide range of feedback rate limits. In the first step of the modified approach, we train a modified version of the proposed neural network in which the outputs of the user-side DNNs are soft binary values (instead of binary values). The trained modified network is used to obtain parameters such as the pilot sequences and the channel estimation scheme. In the second step, we fix the user-side DNN, but apply different quantization resolutions to the user-side DNN output to account for different feedback rates.

For each feedback rate, we conduct another round of training in which only the BS-side DNN parameters are trainable. Numerical results show that this two-step training approach performs almost as well as the architecture that requires one separate trained DNN on both the BS-side and user-side for each value of the feedback rate.

The generalizability of the DNN with respect to the number of users can be achieved using a similar idea. In particular, we propose the following two-step training procedure. First, all different users are assumed to adopt a common set of DNN weights and biases. We then seek to design those weights and biases together with the channel estimation pilot sequences using end-to-end training of a *single-user* system. As the second step, we seek to design the BS-side DNN by training a K-user system in which all the user-side DNNs are fixed to the parameters obtained from the single-user case. This novel two-step training approach has the desirable property that the user-side operations do not depend on the number of users in the network; we only need to train and store different BS-side DNNs to handle varying number of users in the network. Numerical experiments show that for independent and identically distributed (i.i.d.) channels, the DNN trained using such a two-step training method can approach the performance of the original DNN.

B. Related Work

Most of the exiting schemes for limited feedback multiuser FDD MIMO systems fall into two categories [20]. The first category of works focus on reducing feedback overhead by exploiting the spatial and/or temporal correlation of CSI [21]–[23]. In particular, since channels in the correlated CSI scenarios can be represented as a function of uncorrelated sparse vector in some bases (e.g., angular domain for mmWave channels), users can employ sparse recovery algorithms (i.e., compressed sensing (CS) [24], [25]) to recover the sparse channel parameters and subsequently feed back the quantized version of those parameters to the BS, e.g., [26]. To design the precoding matrix, the BS reconstructs the channels from the quantized sparse channel parameters and then employs a conventional linear precoding scheme, e.g., maximum-ratio transmission (MRT) or zero forcing (ZF). Such a CS-based feedback protocol in essence adopts a separate source coding strategy of independent quantization of each users channel. But, as discussed earlier, since the precoding matrix is a function of all users' channels, information theory consideration suggests that we can do better by adopting a DSC strategy. The main points of this paper are to show that a deep learning framework can effectively undertake such a DSC design and to show by extensive numerical simulations that the proposed learning-based precoding scheme, which bypasses explicit channel estimation, can indeed achieve a significantly better performance as compared to the conventional approach of separate channel estimation and precoding, especially when downlink training and feedback resources are limited.

The second category of limited feedback precoding works is codebook based [20], with discrete Fourier transform (DFT) matrix as a common choice for the precoding codebook [27]. In the training phase of the codebook-based precoding schemes, the BS first transmits along the possible precoding directions, and each user then sends feedback about the indices of the top-p strongest received signals and their corresponding SNRs. The feedback information is finally processed at the BS for selecting the precoder of each user from the codebook. If p is set to be one, then the best possible performance of this approach is the performance of the MRT since there is no mechanism for interference management [27], [28]. On the other hand, if p is sufficiently large such that interference management becomes feasible (e.g., [29]), then a large amount of feedback bits per user is required. A thorough comparison between the CSI feedback scheme and the codebook-based precoding method for MIMO systems in rich-scattering environments is provided in [30]. This comparison shows that CSI feedback is preferred in scenarios with very limited feedback rates. The advantage of the CSI feedback scheme in very limited downlink training and feedback resources can be even more remarkable for massive MIMO systems in limited-scattering environments. This is because CSI estimation, quantization, and feedback of a few sparse channel parameters is much more efficient in terms of downlink training and feedback resource utilization, as compared to the codebook-based precoding scheme in which the number of codewords typically would have to scale with the number of antennas. For this reason, we only consider the comparison to the CSI feedback based precoding methods in this paper.

It is noteworthy that the use of DNNs in FDD systems with limited feedback has been adopted in some recent works, e.g., [31]–[35]. However, these works either focus only on a single-user scenario with no interference [31], [32], or they focus on the CSI reconstruction problem at the BS under the assumption that the perfect CSI is available at the users [32]–[35]. The work presented herein provides a more general treatment as: i) we consider the multiuser case in which each user can only sense and feedback its own channel, yet the precoding process is a function of all users' channels; ii) we provide end-to-end training of all system parameters, including downlink pilot sequences, while accounting for CSI estimation error in order to directly enhance the downlink spectral efficiency.

C. Paper Organization and Notations

The remainder of this paper is organized as follows. Section II introduces the system model and the problem formulation for the end-to-end multiuser FDD downlink precoding system design. Section III shows how to represent the FDD downlink precoding system by a neural network architecture, and further discusses how to train the neural network. Section IV provides discussions on the generalizability of the proposed DNN, and Section IV demonstrates the implementation details of the DNN. Section VI presents the simulation results. Finally, conclusions are drawn in Section VII.

This paper uses lower-case letters for scalars, lower-case bold face letters for vectors and upper-case bold face letters for matrices. The real part, the imaginary part, and the dimensions of a complex matrix $\mathbf{V}$ are respectively given by $\Re(\mathbf{V})$, $\Im(\mathbf{V})$, and $\dim(\mathbf{V})$. Further, we use the superscripts $(\cdot)^T$, $(\cdot)^H$, and

$(\cdot)^{-1}$ to denote the transpose, the Hermitian transpose, and the inverse of a matrix, respectively. The identity matrix with appropriate dimensions are denoted by $\mathbf{I}$; $\mathbb{C}^{m \times n}$ denotes an m by n dimensional complex space; $\mathcal{CN}(\mathbf{0}, \mathbf{R})$ represents the zero-mean circularly symmetric complex Gaussian distribution with covariance matrix $\mathbf{R}$; $\mathcal{U}(a, b)$ represents a uniform distribution on the interval $[a, b]$. The notations $\mathrm{Tr}(\cdot)$, $\log_2(\cdot)$, $\log_{10}(\cdot)$, and $\mathbb{E}[\cdot]$ represent the trace, binary logarithm, decimal logarithm, and expectation operators, respectively. Finally, $\|\cdot\|$ indicates the Euclidean norm of a vector.

II. SYSTEM MODEL

A. Signal Model and Problem Formulation

Consider the downlink scenario in an FDD massive MIMO system in which a BS with M transmit antennas serves K single-antenna users. Further, assume that the BS employs linear precoding so that the transmitted signal can be written as:

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{v}_k s_k = \mathbf{V}\mathbf{s}, \tag{1}$$

where $\mathbf{v}_k \in \mathbb{C}^M$ is the precoding vector for the k-th user and forms the k-th column of the precoding matrix $\mathbf{V} \in \mathbb{C}^{M \times K}$, which satisfies the total power constraint, i.e., $\mathrm{Tr}(\mathbf{V}\mathbf{V}^H) \leq P$, and s_k is the symbol to be sent to the k-th user which is normalized so that $\mathbb{E}[\mathbf{s}\mathbf{s}^H] = \mathbf{I}$. By adopting a narrowband block-fading channel model, the received signal at the k-th user in data transmission phase can be written as:

$$y_k = \mathbf{h}_k^H \mathbf{v}_k s_k + \sum_{j \neq k} \mathbf{h}_k^H \mathbf{v}_j s_j + z_k, \tag{2}$$

where $\mathbf{h}_k \in \mathbb{C}^M$ is the vector of downlink channel gains between the BS and user k and $z_k \sim \mathcal{CN}(0, \sigma^2)$ is the additive white Gaussian noise. Given the received signal model at the k-th user in (2), the achievable rate of user k is:

$$R_k = \log_2\left(1 + \frac{|\mathbf{h}_k^H \mathbf{v}_k|^2}{\sum_{j \neq k} |\mathbf{h}_k^H \mathbf{v}_j|^2 + \sigma^2}\right). \tag{3}$$

This paper aims to design the precoding matrix $\mathbf{V}$ at the BS so as to maximize some network-wide utility. For simplicity, the rest of the paper uses the sum rate of the system as the design objective, i.e.,

$$R = \sum_k R_k. \tag{4}$$

In order to design the optimal precoding matrix, it is crucial for the BS to have access to instantaneous CSI. In this paper, we assume that the BS and the users have no prior knowledge of the channel state realizations, and they must acquire the CSI via downlink training and feedback. In particular, to obtain the required information for downlink precoding at the BS, we consider a downlink training phase, prior to the data transmission phase, in which the BS sends training pilots $\widetilde{\mathbf{X}} \in \mathbb{C}^{M \times L}$ of length L, and accordingly, the k-th user observes $\widetilde{\mathbf{y}}_k \in \mathbb{C}^{1 \times L}$ as:

$$\widetilde{\mathbf{y}}_k = \mathbf{h}_k^H \widetilde{\mathbf{X}} + \widetilde{\mathbf{z}}_k, \tag{5}$$

in which the transmitted pilots in the ℓ-th pilot transmission (ℓ-th column of $\widetilde{\mathbf{X}}$) satisfies the power constraint, i.e., $\|\widetilde{\mathbf{x}}_\ell\|^2 \leq P$, and $\widetilde{\mathbf{z}}_k \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ is the additive white Gaussian noise at user k. The k-th user then seeks to obtain the useful information in (5) for the purpose of multiuser downlink precoding and subsequently to feed back this information to the BS in the form of B information bits as:

$$\mathbf{q}_k = \mathcal{F}_k\left(\widetilde{\mathbf{y}}_k\right), \tag{6}$$

where the function $\mathcal{F}_k : \mathbb{C}^{1 \times L} \to \{\pm 1\}^B$ represents the feedback scheme adopted at user k.

Finally, the BS collects the feedback bits from all K users, i.e., $\mathbf{q} \triangleq [\mathbf{q}_1^T, \mathbf{q}_2^T, \ldots, \mathbf{q}_K^T]^T$, and seeks to design the precoding matrix $\mathbf{V}$ as a function of those feedback bits as:

$$\mathbf{V} = \mathcal{P}\left(\mathbf{q}\right), \tag{7}$$

where the function $\mathcal{P} : \{\pm 1\}^{KB} \to \mathbb{C}^{M \times K}$ represents the downlink precoding scheme. We remark that once the precoding matrix is designed at the BS, coherent detection is enabled by an additional downlink pilot transmitted along each beamforming vector to allow the receivers to calibrate detection boundaries. This procedure, which is called *dedicated training phase*, typically requires only a small amount of pilot transmission [36]. In this paper, we make a simplifying assumption that the dedicated training phase is performed perfectly without any significant overhead and accordingly the rate expression in (3) is achievable.

With the above communication models in place, the problem of maximizing the sum rate of a limited-feedback FDD system can be summarized as:

$$\underset{\widetilde{\mathbf{X}}, \{\mathcal{F}_k(\cdot)\}_{\forall k}, \mathcal{P}(\cdot)}{\text{maximize}} \sum_{k=1}^{K} \log_2\left(1 + \frac{|\mathbf{h}_k^H \mathbf{v}_k|^2}{\sum_{j \neq k} |\mathbf{h}_k^H \mathbf{v}_j|^2 + \sigma^2}\right) \tag{8a}$$

$$\text{subject to} \quad \mathbf{V} = \mathcal{P}\left(\left[\mathbf{q}_1^T, \ldots, \mathbf{q}_K^T\right]^T\right), \tag{8b}$$

$$\mathbf{q}_k = \mathcal{F}_k(\mathbf{h}_k^H \widetilde{\mathbf{X}} + \widetilde{\mathbf{z}}_k), \quad \forall k, \tag{8c}$$

$$\mathrm{Tr}(\mathbf{V}\mathbf{V}^H) \leq P, \tag{8d}$$

$$\|\widetilde{\mathbf{x}}_\ell\|^2 \leq P, \quad \forall \ell, \tag{8e}$$

in which the downlink training pilots $\widetilde{\mathbf{X}}$, feedback scheme adopted at the users $\{\mathcal{F}_k(\cdot)\}_{k=1}^{K}$, and the precoding scheme $\mathcal{P}(\cdot)$ adopted at the BS can be optimized to enhance the spectral efficiency. As illustrated in Fig. 1(a), this overall problem, which involves designing the downlink pilots, the users' channel estimation, quantization, and feedback schemes, jointly with the BS's downlink precoding scheme, can be viewed as a DSC problem. This is because the channel estimation and quantization take place in a distributed fashion across the users, and the feedback bits of all users are then processed at a central node, i.e., BS, in order to construct the precoding matrix. This is a challenging task, because designing information theoretically optimal DSC strategy is in general a difficult problem. Simple heuristic, such as independent codebook-based quantization of the channel vector at each user, is likely to be far from the optimum. The main goal of this paper is to show that a data-driven machine learning approach can be used to efficiently design such a DSC scheme.
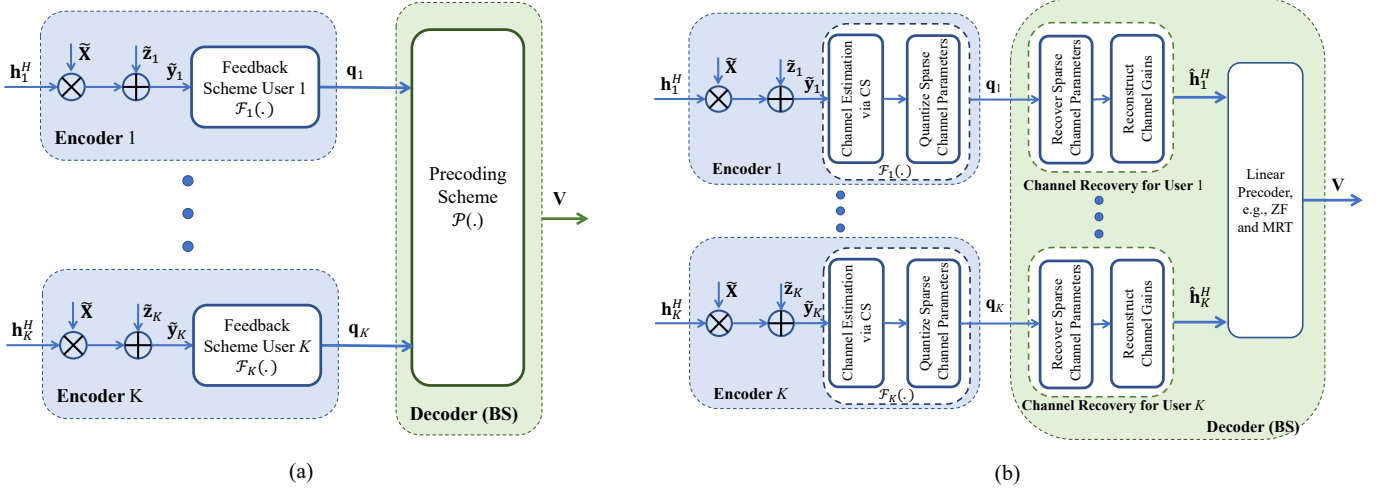
Fig. 1: (a) The end-to-end FDD downlink precoding design problem can be viewed as a DSC problem in which the downlink training pilots and feedback schemes adopted at the users are the encoders' design parameters and the precoding scheme adopted at the BS is the decoder's design parameter. (b) The conventional channel feedback scheme can be regarded as a separate source coding strategy of independent quantization of each users channel.

B. Channel Model and Conventional Approaches

This paper considers an FDD massive MIMO system operating in mmWave propagation environment [9] in which the number of scatterers is limited. Accordingly, the sparse channel of the k-th user is typically modeled with L_p propagation paths, e.g., [37]:

$$\mathbf{h}_k = \frac{1}{\sqrt{L_p}} \sum_{\ell=1}^{L_p} \alpha_{\ell,k} \mathbf{a}_t(\theta_{\ell,k}), \qquad (9)$$

where $\alpha_{\ell,k}$ is the complex gain of the ℓ-th path between the BS and user k, $\theta_{\ell,k}$ is the corresponding angle of departure (AoD), and $\mathbf{a}_t(\cdot)$ is the transmit array response vector. For a uniform linear array with M antenna elements, the transmit array response vector is:

$$\mathbf{a}_t(\theta) = \left[1, e^{j\frac{2\pi}{\lambda}d\sin(\theta)}, \ldots, e^{j\frac{2\pi}{\lambda}d(M-1)\sin(\theta)}\right]^T, \qquad (10)$$

where λ is the wavelength and d is the antenna spacing.

The sparsity of mmWave channels in the angular domain can be exploited in designing the feedback scheme. In particular, a conventional feedback scheme typically involves quantizing the estimated values of the sparse channel parameters [26]. This means that each user first employs a sparse recovery (i.e., compressed sensing) algorithm to estimate the sparse channel parameters then feeds back the quantized version of those parameters to the BS. Subsequently, the BS collects the quantized channel parameters from all K users, reconstructs the imperfect channel estimates based on these parameters, and finally employs one of the conventional linear beamforming methods, e.g., MRT or ZF, given the imperfect CSI. Such a conventional approach typically leads to a good performance only for systems with (i) sufficiently large pilot length L in which a decent sparse parameter estimation can be achieved via compressed sensing, and (ii) sufficiently large number of feedback bits B where the quantization error can be made sufficiently small.

This paper aims to show that it is possible to design FDD system with excellent performance even with short training sequences and small amount of limited feedback information bits. The above conventional channel feedback scheme illustrated in Fig. 1(b) has room for improvement, because it amounts to a separate source coding strategy of independent quantization of each user's channel. However, because the estimated channels from all the users are used jointly at the BS to compute a downlink precoding matrix, a *distributed source coding* strategy can do better [13], [14], [16]. This is true even if the channels to each user are uncorrelated [16]. The design of DSC strategy is however quite challenging. In this paper, we propose a deep learning framework to undertake such a design. In particular, we propose a neural network architecture employing a DNN at each user to map the received pilots directly into feedback bits, and a DNN at the BS to map the feedback bits from all the users directly into the precoding matrix.

We note that the proposed neural network architecture also takes the design of downlink training pilots into account by modeling $\widetilde{\mathbf{X}}$ as a linear neural network layer. The block diagram of the proposed neural network architecture that represents an end-to-end FDD downlink system is illustrated in Fig. 2. The details of the proposed neural network architecture are discussed in the next section.

III. FDD DOWNLINK PRECODING SYSTEM DESIGN USING DEEP LEARNING

In this section, we present how to use neural networks to model an FDD downlink system described in Section II, which involves two phases: (i) downlink training and uplink feedback phase, and (ii) downlink data transmission phase. Further, we discuss how to train the proposed neural network architecture in order to jointly design the downlink training pilots $\widetilde{\mathbf{X}}$, the feedback scheme adopted at each user $\mathcal{F}_k(\cdot), \forall k$, and the downlink precoding scheme $\mathcal{P}(\cdot)$.
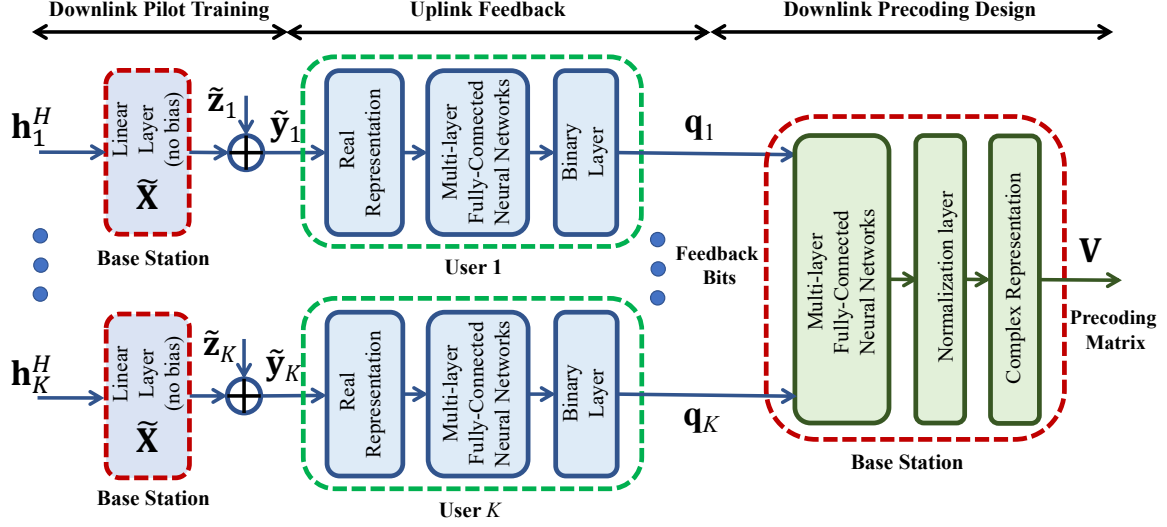
Fig. 2: The block diagram of the proposed neural network architecture that represents an end-to-end K-user FDD downlink precoding system which involves two phases: (i) downlink training and feedback phase, and (ii) data transmission phase.

A. DNN Representation of DSC Strategy

In order to represent the FDD downlink precoding system described in Section II as a neural network, we need to model downlink pilot transmission and the users' operations in the downlink training phase as well as the BS's operations in the data transmission phase.

1) Downlink Pilot Training: We begin by modeling the first part of the downlink training phase, i.e., downlink pilot transmission, in which the BS sends training pilots $\widetilde{\mathbf{X}} \in \mathbb{C}^{M \times L}$ in L downlink transmissions, and consequently the k-th user observes $\widetilde{\mathbf{y}}_k = \mathbf{h}_k^H \widetilde{\mathbf{X}} + \widetilde{\mathbf{z}}_k$. By considering $\mathbf{h}_k^H$ as the input, it is easy to see that the received signal at each user in the downlink training phase can be modeled as the output of a fully-connected neural network layer with linear activation function, in which the weight matrix is $\widetilde{\mathbf{X}}$ and the bias vector is zero, followed by an additive zero-mean noise with variance σ^2.

To ensure that the designed weight matrix $\widetilde{\mathbf{X}}$ satisfies the per-transmission power constraint P, we adopt a weight constraint under which each column of $\widetilde{\mathbf{X}}$ satisfies $\|\widetilde{\mathbf{x}}_\ell\|^2 \leq P$. We remark that in the machine learning literature weight constraints are employed as means of regularization to reduce overfitting in DNNs, e.g., [38]. However, in this paper, we adopt a particular choice of weight constraint explained above to model the physical constraint on the transmit power level of a cellular BS.

2) Uplink Feedback: Upon receiving $\widetilde{\mathbf{y}}_k$ in the downlink training phase, the main objective of user k is to summarize its observation from $\widetilde{\mathbf{y}}_k$ and to feed back that summary as B information bits to the BS for the purpose of downlink precoding. This procedure can be represented by an R-layer fully-connected DNN in which the feedback bits of user k can be written as:

$$\mathbf{q}_k = \text{sgn}\left(\mathbf{W}_R^{(k)} \sigma_{R-1}\left(\cdots \sigma_1\left(\mathbf{W}_1^{(k)} \bar{\mathbf{y}}_k + \mathbf{b}_1^{(k)}\right) \cdots\right) + \mathbf{b}_R^{(k)}\right),$$
(11)

where $\mathbf{q}_k \in \{\pm 1\}^B$, $\{\mathbf{W}_r^{(k)}, \mathbf{b}_r^{(k)}\}_{r=1}^R$ is the set of the trainable parameters for user k, σ_r is the activation function for the r-th layer, and the sign function $\text{sgn}(\cdot)$ is the activation function of the last layer to generate bipolar feedback bits for each component of $\mathbf{q}_k$. In (11), the real representation of $\widetilde{\mathbf{y}}_k$, i.e.,

$$\bar{\mathbf{y}}_k \triangleq \left[\Re\left(\widetilde{\mathbf{y}}_k\right), \Im\left(\widetilde{\mathbf{y}}_k\right)\right]^T,$$
(12)

is considered as the input of the DNN since most of the existing deep learning libraries only support real-value operations. Further, the dimensions of the trainable weight matrices and the bias vectors in (11) are respectively:

$$\dim\left(\mathbf{W}_r\right) = \begin{cases} \ell_r \times 2L, & r = 1, \\ \ell_r \times \ell_{r-1}, & r = 2, \ldots, R-1, \\ B \times \ell_{r-1}, & r = R, \end{cases}$$
(13)

and

$$\dim\left(\mathbf{b}_r\right) = \begin{cases} \ell_r \times 1, & r = 1, \ldots, R-1, \\ B \times 1, & r = R, \end{cases}$$
(14)

where ℓ_r is the number of neurons in the r-th hidden layer.

3) Downlink Precoding Design: Under the assumption of an error-free feedback channel between each user and the BS, the BS collects the information bits of all users, then designs the precoding vectors as a function of these information bits. Analogous to the user side, the operations at the BS can be modeled by another DNN with T dense layers, where the t-th layer includes ℓ_t' neurons. In particular, the real representation of the collection of the precoding vectors, i.e.,

$$\mathbf{v} = \left[\text{vec}\left(\Re\left(\mathbf{V}\right)\right)^T, \text{vec}\left(\Im\left(\mathbf{V}\right)\right)^T\right]^T,$$
(15)

can be written as:

$$\mathbf{v} = \widetilde{\sigma}_T \left(\widetilde{\mathbf{W}}_T \widetilde{\sigma}_{T-1} \left(\cdots \widetilde{\sigma}_1 \left(\widetilde{\mathbf{W}}_1 \mathbf{q} + \widetilde{\mathbf{b}}_1 \right) + \cdots \right) + \widetilde{\mathbf{b}}_T \right),$$ (16)

where $\widetilde{\sigma}_t$, $\widetilde{\mathbf{W}}_t$, and $\widetilde{\mathbf{b}}_t$ are the activation function, the weights, and the biases in the t-th layer, respectively, and the collection of feedback bits of all K user, i.e., $\mathbf{q} = [\mathbf{q}_1^T, \ldots, \mathbf{q}_K^T]^T$, is the input vector to the DNN. In order to ensure that the total power constraint is satisfied, a normalization layer with activation function:

$$\widetilde{\sigma}_T(\mathbf{u}) = \min(\sqrt{P}, \|\mathbf{u}\|) \frac{\mathbf{u}}{\|\mathbf{u}\|}$$ (17)

is employed at the last layer of the DNN.

The block diagram of the overall proposed neural network architecture that represents an end-to-end two-phase FDD downlink precoding system is illustrated in Fig. 2. In this neural network the trainable parameters are the training pilot matrix $\widetilde{\mathbf{X}}$, the DNN parameters $\Theta_{\mathrm{R}}^{(k)} \triangleq \{\mathbf{W}_r^{(k)}, \mathbf{b}_r^{(k)}\}_{r=1}^R$ at the user side, and the DNN parameters $\Theta_{\mathrm{T}} \triangleq \{\mathbf{W}_t, \mathbf{b}_t\}_{t=1}^T$ at the BS side.

B. DNN Training with a Hidden Binary Layer

We now describe the training of the DNN architecture in Fig. 2 for the sum rate maximization objective as stated in the following:

$$\max_{\widetilde{\mathbf{X}}, \{\Theta_{\mathrm{R}}^{(k)}\}_{k=1}^K, \Theta_{\mathrm{T}}} \mathbb{E}_{\mathbf{H}, \widetilde{\mathbf{z}}} \left[\sum_k \log_2 \left(1 + \frac{|\mathbf{h}_k^H \mathbf{v}_k|^2}{\sum_{j \neq k} |\mathbf{h}_k^H \mathbf{v}_j|^2 + \sigma^2} \right) \right],$$ (18)

where the expectation is over the distribution of the channels, i.e., $\mathbf{H} \triangleq [\mathbf{h}_1, \ldots, \mathbf{h}_K]^H$, and the distribution of the noise in the downlink training phase, i.e., $\widetilde{\mathbf{z}} \triangleq [\widetilde{\mathbf{z}}_1, \ldots, \widetilde{\mathbf{z}}_K]^T$. The parameter space consists of the training pilot matrix, the users' feedback schemes, and the BS's precoding scheme.

We assume certain distributions of the channels and the noise in the downlink training phase and accordingly generate a large set of channel and noise realizations for the training purpose. The training problem for (18) can then be efficiently tackled by employing stochastic gradient descent (SGD) algorithms in which the expectation in (18) is approximated with the empirical average over the training samples. SGD-based training algorithms require partial derivatives of the loss function, here the sum rate expression, with respect to all the trainable parameters in order to update the parameters in each iteration. The partial derivatives are computed via *back-propagation* method which is an efficient implementation of the chain rule in directed computation graphs.

Due to the fact that the derivative of the output of a binary thresholding neuron is zero almost everywhere (with the exception of the origin where the function is not differentiable), the conventional back-propagation method cannot be directly used to train the neural layers prior to that binary layer. A common practice in the machine learning literature to overcome this issue is to approximate the activation function of a binary thresholding layer by another smooth differentiable function during the back-propagation phase [17]–[19]. Such approximation of a binary layer in the back-propagation phase,

which is first introduced by Hinton in [17], is known as straight-through (ST) estimation. A variant of the ST estimator, called sigmoid-adjusted ST, is to replace the derivative factor with the gradient of the function $2\,\mathrm{sigm}(u) - 1$, where $\mathrm{sigm}(u) = 1/(1 + \exp(-u))$ is the sigmoid function. It is shown in [19] that the performance of the sigmoid-adjusted ST estimator can be further improved by adopting the slope-annealing trick, in which the slope of the sigmoid function is gradually increased as training progresses. In particular, the sigmoid-adjusted ST with slope annealing estimator approximates the sign function $\mathrm{sgn}(u)$ in the back-propagation phase with a properly scaled sigmoid function as:

$$2\,\mathrm{sigm}(\alpha^{(i)}u) - 1 = \frac{2}{1 + \exp(-\alpha^{(i)}u)} - 1,$$ (19)

where $\alpha^{(i)}$ is the annealing factor in the i-th epoch satisfying $\alpha^{(i)} \geq \alpha^{(i-1)}$. In this paper, we adopt the sigmoid-adjusted ST with annealing during the back-propagation phase to compute the gradients of the binary layer considered at the last stage of the user side in the neural network in Fig. 2. Further implementation details are provided in Section V.

IV. GENERALIZABILITY

A. Towards Robust DNN Design

The training of the proposed DNN architecture assumes a specific channel environment. The natural question that arises for any deep learning-based algorithm, is then how generalizable the proposed DNN is, if it is trained under one set of system parameters, but tested under a different set of parameters. For the problem under consideration, we can group the system parameters into two categories. The first category consists of the parameters that only change the input distribution, e.g., the channel parameters and the noise statistics. For these parameters, training under a variety of system parameters typically enhances robustness. By considering the number of channel paths L_p as an example, we numerically show in Section VI that by training the proposed DNN over different number of paths, the robustness of the proposed DNN can be enhanced. However, for the second category of the system parameters, which also change the input/output dimensions of some layers of the DNN, training the DNN to operate for different system dimensions is more challenging. In this case, modification of the proposed DNN architecture and/or devising novel training procedures are needed. In the next two subsections, we explain how to enhance the generalizability of the proposed DNN with respect to the number of feedback bits B and the number of users K.

B. Towards Generalizability for B

In the proposed neural network architecture in Fig. 2, it appears that we have to train a different DNN for each value of feedback rate limit B since the dimension of the output of each user's DNN and accordingly the dimension of the input to the BS's DNN are determined by the value of B. This would be a tedious task for practical implementation if the amount of feedback can possibly vary. In practical system design, it is desirable to train a common neural network that

can operate over a wide range of feedback capacities. To address this need, we propose the following two-step training approach. We propose to first train a modified version of the proposed neural network in which the output of the user-side DNN is not binary anymore and instead is soft binary valued, i.e., real numbers belongs to $[-1, 1]$, generated by S neurons with hyperbolic tangent ($\tanh$) activation functions. After this modified network has been trained, we obtain the empirical probability distribution function (PDF) of the output of the $\tanh$ layer, then design the optimal scalar quantizer for this distribution for different values of quantization bits (i.e., Q) according to the Lloyd-Max algorithm. After the encoder parameters including the training pilot sequences and the user-side DNN parameters are obtained, as the second step of the training procedure, we seek to design the decoder parameters at the BS to generate the precoding matrix. In particular, the BS receives a Q-bit quantized version of S soft binary signals from each user, and the task of the BS-side DNN is to map these $K \times S$ quantized signals to the precoding matrix such that the average sum rate is maximized. The weights and biases of the BS-side DNN can be learned using the SGD-based training. Note that in this scheme the amount of feedback per user is equal to $B = S \times Q$, hence by varying different quantization levels Q, we can train the same DNN to operate for different values of B.

C. Towards Generalizability for K

The proposed architecture has a separate DNN at each user. At a first glance, it may appear that they need to be trained separately; further, their trained parameters would also depend on the total number of users in the system. Our experimental results, however, suggest that as long as the channel distribution for different users are i.i.d., one common user-side DNN can be adopted, regardless the number of users in the system. Motivated by this observation, we propose to first learn the encoding parameters (including the pilot matrix and the channel estimation/feedback scheme) by training a single-user system. Then we adopt the same encoding DNN for all users, and only train the parameters of the BS-side DNN separately depending on the total number of users in the system. Such a design is much more efficient than training different user-side DNNs at different users for different systems. Only at the BS side, we need to train and store different DNNs for handling different numbers of users in the network.

V. IMPLEMENTATION DETAILS

We implement the proposed neural network in Fig. 2 using two open-source deep learning libraries, namely TensorFlow [39] and Keras [40]. We employ a variant of the SGD-based training method, called *Adam optimizer* [41], with a mini-batch size of 1024 and a learning rate progressively decreasing from 10^{-3} to 10^{-5}. Unless otherwise stated, we use 4-layer fully-connected DNN at the user side as well at the BS side, i.e., $R = T = 4$, while the number of hidden neurons of different layers at the user side and at the BS side are $[\ell_1, \ell_2, \ell_3, \ell_4] = [1024, 512, 256, B]$ and $[\ell'_1, \ell'_2, \ell'_3, \ell'_4] = [1024, 512, 512, 2MK]$, respectively. For faster convergence,

each dense layer is preceded by a batch normalization layer [42]. Further, we adopt the rectified linear unit (ReLU) activation function at the hidden layers. In the simulations, we slowly increase the annealing parameter in the i-th epoch according to $\alpha^{(i)} = \max\{1.001\alpha^{(i-1)}, 10\}$ where $\alpha^{(0)} = 0.5$ and each epoch consists of 200 mini-batches.

In order to optimize the downlink training pilot matrix $\widetilde{\mathbf{X}}$, we define $\widetilde{\mathbf{X}}$ as a training variable in TensorFlow whose initial value is randomly generated according to i.i.d. complex Gaussian distribution with zero mean and variance $\sqrt{P/M}$ such that the transmitted pilots in the ℓ-th pilot transmission satisfies the power constraint, i.e., $\|\widetilde{\mathbf{x}}_\ell\|^2 \leq P$. To ensure that the final designed pilot matrix also satisfies such a power constraint, we always normalize the updated $\widetilde{\mathbf{X}}$ in each iteration such that $\|\widetilde{\mathbf{x}}_\ell\|^2 = P$.

We fix the distribution of the channels as well as the distribution of the noise in the downlink training phase, so that we can generate as many data samples as needed for training the DNN. This assumption enables us to investigate the ultimate performance of the deep learning-based precoding design for FDD systems with limited feedback. The investigation on the minimal size of the training data set that leads to a reasonable performance is a direction for future work.

We monitor the generalization performance of the DNN during training by computing the achieved average sum rate by the DNN for a holdout validation data set of 10^4 samples, and keep the model parameters that have achieved the best generalization performance so far. The training procedure is terminated when the generalization performance for the validation data set has not improved over a large number of (e.g., 100) epochs. After the proposed DNN in Fig. 2 is trained, we have access to the design of the training pilot matrix $\widetilde{\mathbf{X}}$, the feedback scheme for each user $\mathcal{F}_k(\cdot)$, and the BS's precoding scheme $\mathcal{P}(\cdot)$.

VI. NUMERICAL RESULTS

We now illustrate the performance of the proposed deep learning-based precoding method for FDD systems with limited feedback. We compare the performance of the proposed deep learning-based precoding framework with that of the conventional MRT and ZF precoding methods. For each of the MRT and ZF precoding baseline methods we consider four different system settings: (i) perfect CSI at the BS and the users; (ii) perfect CSI at the users with limited-feedback links between each user and the BS; (iii) no prior CSI available at either the BS or the users with infinite-capacity feedback links; and (iv) no prior CSI available at either the BS or the users with limited-capacity feedback links. Before presenting the numerical results, we first provide a brief explanation of each of the baseline methods.

A. Baseline Methods

1) MRT/ZF with Full CSI at Transmitter (CSIT): If full CSI is available at the transmitter (e.g., the BS), conventional
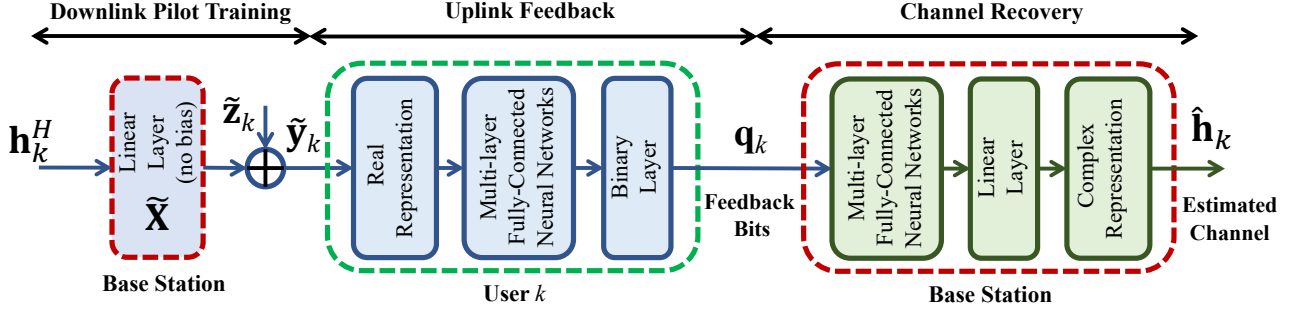
Fig. 3: The block diagram of the neural network architecture which models the idea of per-user channel estimation. The estimated channels obtained by the trained DNN are then used to design the linear precoders.

multiuser linear precoding schemes such MRT or ZF can be used, in which the precoding matrix is respectively given by:

$$\mathbf{V}_{\text{MRT}} = \gamma_{\text{MRT}}\mathbf{H}^H, \tag{20}$$

$$\mathbf{V}_{\text{ZF}} = \gamma_{\text{ZF}}\mathbf{H}^H(\mathbf{H}\mathbf{H}^H)^{-1}, \tag{21}$$

where $\mathbf{H}$ is the instantaneous CSI, and γ_{MRT} and γ_{ZF} are constants ensuring that the power constraint is satisfied.

2) MRT/ZF with Full CSI at Receiver (CSIR) and Finite-Capacity Feedback: Under the perfect CSI assumption at the user side, user k has a perfect knowledge about its sparse channel parameters, i.e., $\{\alpha_{\ell,k}, \theta_{\ell,k}\}_{\forall \ell=1}^L$. User k aims to transfer this knowledge to the BS by sending the index of the quantized version of the channel parameters, $\{\hat{\alpha}_{\ell,k}, \hat{\theta}_{\ell,k}\}_{\forall \ell=1}^L$, over an error-free B-bit finite-capacity feedback link. By using the channel model in (9), the BS can then reconstruct the estimated channel vectors, $\hat{\mathbf{h}}_k, \forall k$, from those quantized channel parameters. Finally, the MRT and ZF precoder matrices can be computed respectively via expressions (20) and (21) in which the instantaneous CSI matrix $\mathbf{H}$ is replaced with the matrix of channel estimates $\hat{\mathbf{H}} \triangleq [\hat{\mathbf{h}}_1, \ldots, \hat{\mathbf{h}}_K]^H$. In the simulations, we assume that each user allocates $\frac{B}{3L_p}$ quantization bits to each real parameter of the channel, i.e, $\{\Re(\alpha_{\ell,k}), \Im(\alpha_{\ell,k}), \theta_{\ell,k}\}_{\forall \ell=1}^{L_p}$. Further, by assuming that the distribution of the channel parameters is known at the BS and the user, we consider the optimal quantization scheme for the sparse channel parameters obtained by the Lloyd-Max quantization algorithm.

3) MRT/ZF with Downlink (DL) Training and Infinite-Capacity Feedback: In this baseline, no prior channel knowledge is initially assumed at either BS or the users. However, each user estimates its corresponding channel vector in the DL training phase and subsequently feeds the estimated channel back to the BS over an infinite-capacity link. As a result, the BS constructs the MRT/ZF precoder based on the estimated channels at the users in the downlink training phase. For estimating sparse channels in mmWave environment, user k seeks to recover the sparse channel parameters by employing CS techniques. Here, we adopt a widely-used CS algorithm orthogonal matching pursuit (OMP) [25].

4) MRT/ZF with DL Training and Finite-Capacity Feedback: Similar to the previous case, the channel parameters are first estimated at each user independently, then quantized

versions of the parameters are fed back to the BS via finite-capacity feedback links. Different DL training methods are investigated, e.g., OMP for sparse channel recovery. But in particular, to illustrate the gain of the proposed DSC strategy as compared to the separate source coding strategy, we implement a scheme that uses a deep learning strategy to perform channel estimation, quantization, feedback and reconstruction at the BS, followed by conventional linear precoding. We train a neural network architecture as shown in Fig. 3 to jointly design the training pilots, feedback scheme of the user, and the channel reconstruction function at the BS so that the reconstructed channels at the BS for all users are as accurate as possible under the finite capacity feedback constraint. Accordingly, we choose the average mean squared error (MSE) of the channels as the loss function, i.e., $\mathcal{L} = \mathbb{E}\left[\|\hat{\mathbf{h}}_k - \mathbf{h}_k\|^2\right]$. For implementing this neural network, we follow the same implementation strategies described in Section V. The estimated channels are then used to compute the MRT and ZF precoders. Note that the channels for different users are i.i.d., so the same DNN structure can be used for all users.

B. Performance Comparison to Baseline Methods

The numerical simulations in this subsection demonstrate the performance of the proposed deep learning-based precoding for an FDD system as compared to the described baseline methods. Here, we consider an FDD massive MIMO system in which a BS with $M = 64$ half-wavelength-spaced antennas serves $K = 2$ users in an $L_p = 2$ path environment. We assume that the complex gain of each path is modeled by a Gaussian distributed random variable, i.e., $\alpha_{\ell,k} \sim \mathcal{CN}(0,1)$, and the corresponding AoD is modeled by a uniform distributed random variable, i.e., $\theta_{\ell,k} \sim \mathcal{U}(-30°, +30°)$. We set the signal-to-noise-ratio as $\text{SNR} = 10\log_{10}(\frac{P}{\sigma^2}) = 10\text{dB}$. We note that some system parameters are chosen to be relatively small, e.g., the number of users and the number of channel paths, such that we would be able to train the proposed neural network under different settings in a relatively short period of time. While the numerical results for such relatively small system parameters can indeed serve as a proof of concept for the proposed deep learning-based precoding approach, we also provide discussions and simulations on how to make
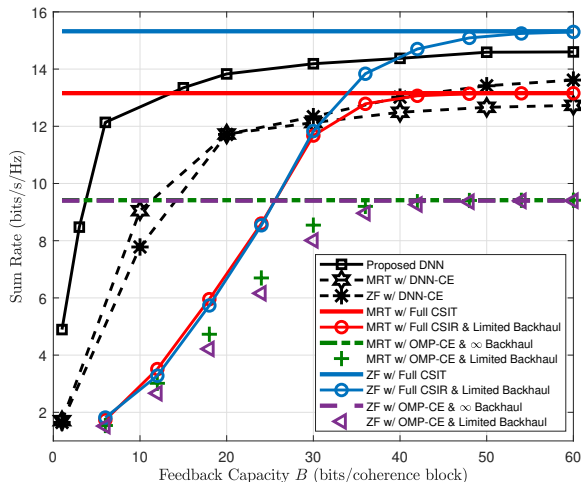
Fig. 4: Sum rate achieved by different methods in a 2-user FDD system with $M = 64$ and $L = 8$.

Fig. 5: Sum rate achieved by different methods in a 2-user FDD system with $M = 64$ and $L = 64$.

the proposed architecture more generalizable to larger system parameters.

As the first experiment, we plot the average sum rate[1] against per-user feedback capacity B for a system in which the pilot length $L = 8$ is much smaller than the number of antennas $M = 64$. Fig. 4 shows that the proposed deep learning-based precoding approach with only 15-bit finite-capacity feedback link already outperforms the MRT precoding even for the scenario where MRT precoder is designed based on the perfect CSI. Note that the MRT precoding approach only seeks to maximize the useful signal power without having a mechanism to mange the interference. So, achieving a better performance by the proposed approach as compared to MRT suggests that the trained DNN has learned a mechanism to reduce the inter-user interference in a multiuser FDD system.

Further, Fig. 4 illustrates that the proposed method significantly outperforms the MRT/ZF precoding schemes with downlink training and conventional OMP channel sparse recovery. This implicitly means that for scenarios that the number of channel observations is very limited (e.g., $L = 8$) and hence, a good sparse recovery may not be feasible, the optimal feedback scheme is quite different from first estimating then quantizing the sparse channel parameters. The numerical results in Fig. 4 indicates that the proposed deep learning-based precoding method is indeed an efficient framework to design a better feedback scheme for such scenarios.

In Fig. 4, we also illustrate the performance of a deep learning-based approach that implements the idea of channel recovery followed by linear MRT/ZF precoding. Fig. 4 shows that in short-pilot-length scenarios such as $L = 8$, a DNN with channel recovery-based approach can outperform the convectional CS-based recovery methods. This is another confirmation showing that the approach of first estimating then quantizing the sparse channel parameters is quite suboptimal for limited-L scenarios.

Moreover, Fig. 4 shows that the proposed DNN which directly optimizes the sum rate and bypasses the channel recovery procedure achieves a higher rate as compared to the DNN that first recovers each user's channels and then applies MRT/ZF at the same feedback capacity constraint. To quantify the gain of the proposed DSC strategy as compared to an optimized separate source coding strategy, we note that if operating at 12 bits/s/Hz sum rate (about 80% of the sum rate achieved with full CSIT), the DSC strategy requires only about 7-bits of feedback, as compared to more than 20-bits of feedback for the separate source coding strategy, whereas both are designed using DNN. Fig. 4 also shows that the proposed precoding method with about 20-bit finite-capacity feedback links can already achieve almost 90% of the sum rate in the ZF precoding method with perfect CSIT.

Finally, to show that the proposed scheme can eventually approach the performance of ZF with perfect CSIT for sufficiently large B and L, in the next experiment, we consider the pilot length $L = M = 64$ in which a near perfect channel recovery at the user side may be possible. Fig. 5 shows that the proposed method always achieves a higher sum rate as compared to the other limited-feedback baseline methods, while it approaches the performance of the ZF with perfect CSIT at feedback rate about $B \geq 50$.

C. Towards Generalizability in L_p

In the previous subsection, we train and test the proposed neural network for the same limited-scattering environment, in which the number of channel paths is assumed to be $L_p = 2$. In this subsection, we are interested to numerically investigate the following two questions. (i) How does the proposed neural network perform if there is a mismatch between the number of paths in the channel realizations of the training data set and that of the test data set? (ii) Can we enhance the robustness of the proposed neural network such that it can perform well for a wide range of channel distributions, i.e., different values of L_p?

---

[1]In all numerical simulations, we report the average sum rate performance of different methods over 10^4 channel realizations.
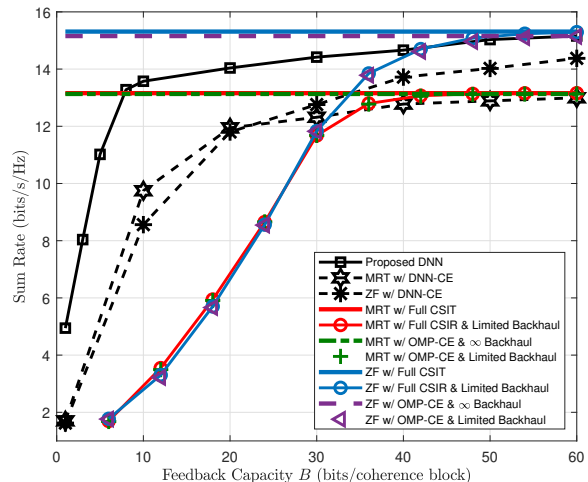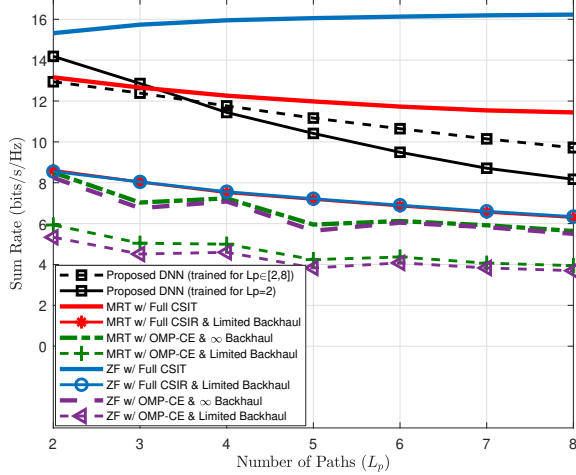
Fig. 6: Sum rate achieved by different methods in a 2-user FDD system with $M = 64$, $B = 30$, and $L = 8$.

Fig. 7: Sum rate achieved by different methods in a 2-user FDD system with $M = 64$, $B = 30$, and $L = 64$.

To numerically investigate the answers for the above questions, we consider a downlink FDD system with $B = 30$ while the other parameters remain the same as in the previous experiments. For the proposed method, we train two instances of the proposed neural network; one of them is trained over the channel realizations with $L_p = 2$, and the other one is trained using samples with $L_p \in \{2, 3, \ldots, 8\}$. For the baselines with imperfect CSIT, we assume that the sparse channel parameters of the two strongest paths are estimated at the users and are fed back to the BS. The achievable sum rate against the number of scattering paths L_p for $L = 8$ and $L = 64$ are respectively illustrated in Fig. 6 and Fig. 7. As it can be seen from these two figures, the performance of the DNN that is trained only with $L_p = 2$ samples is degraded when it is tested against other values of L_p. This is because there is a mismatch between the distribution of the training data set and that of the test data set. This performance degradation is more severe when the training pilot sequence is short, e.g., $L = 8$. This is most probably because when the training resources are limited, the proposed DNN does its best to fully exploit the distribution of the input and to tailor the feedback scheme to that particular distribution. However, when the number of training pilot sequences is not the bottleneck, e.g., $L = 64$, the trained DNN can potentially handle a wider range of the channel distributions.

From Fig. 6 and Fig. 7, we also see that the proposed DNN can achieve a more robust performance when it is trained with $L_p \in \{2, 3, \ldots, 8\}$ samples. This suggests that training the DNN on a wider range of channel parameters can help design more robust systems when perfect prior knowledge about those parameters are not available. Nevertheless, Fig. 6 and Fig. 7 show that the proposed DNN still significantly outperforms the other baselines with limited feedback, even when there is a mismatch between the channel parameters in the training set and those in the test set.
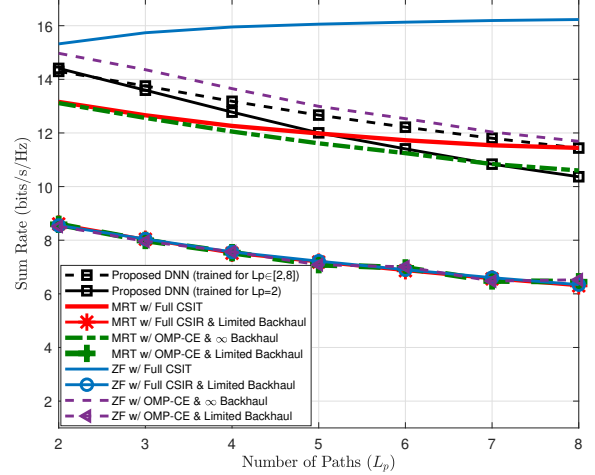
D. Towards Generalizability in B

In Section VI-B we have presented the performance of the proposed deep learning-based approach against the capacity of the feedback links B. To do so, we propose to train a different instance of the proposed neural network in Fig. 2 for each value of B. In practice, it would be more desirable to train a single DNN which can operate for different B's. Accordingly, we now illustrate the performance of the proposed two-step training approach, presented in Section IV-B, which trains a common DNN for different values of B. In this experiment, we consider the same setup as in the experiment of Fig. 4. As explained in Section IV-B, in the first step, we train a modified DNN architecture where the $\tanh$ layer with S soft outputs replaces the binary layer. In Fig. 8, the empirical PDF of the $\tanh$ layer output of the trained modified network is plotted. Interestingly, we observe that this empirical PDF is well approximated as a Gaussian distribution. Next, we obtain the optimal scalar quantizer for this distribution using the Lloyd-Max method. The optimized quantization regions and their corresponding representation points for $Q = 3$ bits are shown in Fig. 8. Finally, we train the parameters of the BS-side DNN while fixing the user-side DNN. The performance of this two-step training approach is shown in Fig. 9. It can be seen that there is only marginal performance degradation in adopting this training approach, which provides a common DNN that can handle different values of B, as compared to using the earlier approach, which requires to train a separate DNN for each feedback capacity value. This shows that the proposed two-step approach can indeed help improve the generalizability of the proposed neural network with respect to the feedback capacity, with the caveat that only integer values of S can be supported in this two-step training approach.

E. Towards Generalizability in K

In all previous experiments, we have evaluated the performance of the proposed DNN for $K = 2$. In this last experiment, we investigate whether or not the proposed approach
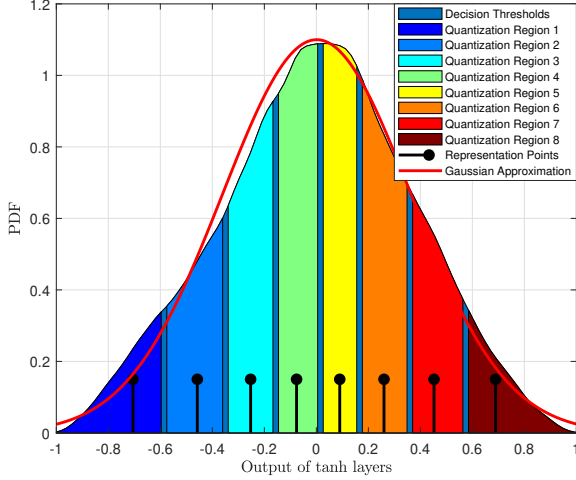
Fig. 8: The empirical PDF of the tanh layer output in a proposed modified neural network which has been trained for a system with $M = 64$, $K = 2$, and $L = 8$. This figure also indicates the quantization regions and the corresponding representation points for the optimal 3-bit quantizer.
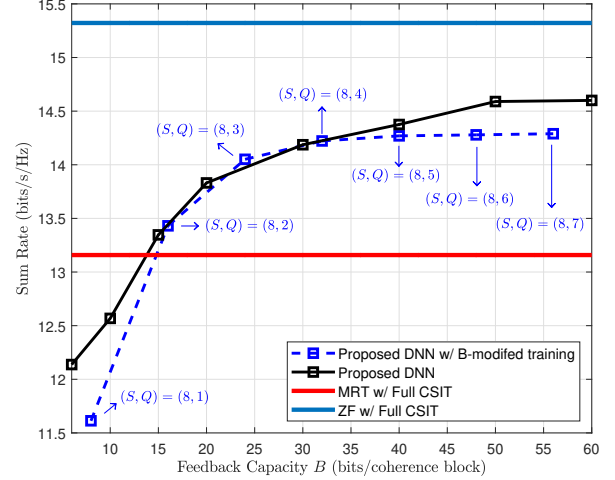

Fig. 9: Sum rate achieved by different methods in a 2-user FDD system with $M = 64$ and $L = 8$.

Fig. 10: Sum rate achieved by different methods in a K-user FDD system with M=64, B=30, and L=8.

can handle the scenarios in which the BS serves more users. In this experiment, we consider a system with downlink training resources of $L = 8$ and $B = 30$. Further, to show that the proposed method can handle a larger range of AoDs, in this experiment we model the AoDs as $\theta_{\ell,k} \sim \mathcal{U}(-60°, +60°)$. The other system parameters remain the same as in the previous subsections. We remark that since the input dimension of the decoding DNN is KB, for larger values of K we need to increase the capacity of the BS's DNN in order to fully process the input signals. In the simulations, we still employ a 4-layer DNN at the BS but this time with $[\ell'_1, \ell'_2, \ell'_3, \ell'_4] = [2048, 1024, 512, 2MK]$ number of neurons per layer.

Fig. 10 plots the sum rate against the number of users in the network, i.e., K. Fig. 10 shows that similar to the previous experiments with $K = 2$, the proposed deep learning framework achieves a much higher rate as compared to all the other conventional baselines with limited downlink training resources. However, in the numerical simulations, we observe that for larger values of K the training procedure becomes slower since many parameters have to be jointly designed. To get around with this problem, we also examine the performance of the two-step training procedure proposed in Section IV-C. It can be seen from Fig. 10 that the performance of the training approach of Section IV-C is very similar to that of the end-to-end training approach in which all the system parameters are jointly designed. We remark that the good performance of the two-step training approach is possible because the different users all have i.i.d. channels. Investigating the cases where the channels are correlated or the channel distributions of different users are different can be considered as an interesting direction for future work.

VII. CONCLUSION

This paper proposes a deep learning approach to design a downlink FDD massive MIMO system with limited feedback
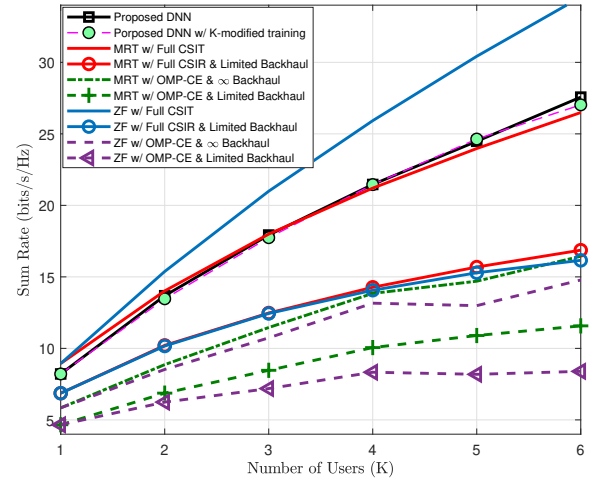
which is formulated to a DSC problem. In particular, this paper presents an end-to-end FDD downlink precoding system, including the downlink training phase, the uplink feedback phase, and the downlink precoding phase, using a user-side DNN and a BS-side DNN. We propose a machine learning framework to jointly design the pilots in the downlink training phase, the channel estimation and feedback strategy adopted at the users, and the multiuser precoding scheme at the BS. This paper also investigates how to make the proposed DNN architecture more generalizable to different system parameters. Numerical results show that the proposed DSC strategy for FDD precoding, which bypasses explicit channel estimation, can achieve an outstanding performance especially for the scenarios in which the downlink training pilot length and/or the feedback capacity are very limited.

REFERENCES

[1] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, May 2019.

[2] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, Mar. 2019.

[3] F. Sohrabi and W. Yu, "One-bit precoding constellation design via autoencoder-based deep learning," in *IEEE Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 2019, pp. 754–758.

[4] F. Sohrabi, H. V. Cheng, and W. Yu, "Robust symbol-level precoding via autoencoder-based deep learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Barcelona, Spain, May 2020.

[5] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1248–1261, June 2019.

[6] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[7] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[8] E. Björnson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: Ten myths and one critical question," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 114–123, Feb. 2016.

[9] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 101–107, June 2011.

[10] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[11] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.

[12] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 80–94, Sept. 2004.

[13] A. El Gamal and Y.-H. Kim, *Network information theory*. Cambridge University Press, 2011.

[14] J. Korner and K. Marton, "How to encode the modulo-two sum of binary sources (corresp.)," *IEEE Trans. Inf. Theory*, vol. 25, no. 2, pp. 219–221, Mar. 1979.

[15] O. A. Hanna, Y. H. Ezzeldin, T. Sadjadpour, C. Fragouli, and S. Diggavi, "On distributed quantization for classification," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 237–249, May 2020.

[16] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498–3516, Oct. 2007.

[17] G. E. Hinton, "Neural networks for machine learning," *Coursera lectures*.

[18] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," Aug. 2013. [Online]. Available: https://arxiv.org/abs/1308.3432

[19] J. Chung, S. Ahn, and Y. Bengio, "Hierarchical multiscale recurrent neural networks," Sept. 2016. [Online]. Available: https://arxiv.org/abs/1609.01704

[20] D. J. Love, R. W. Heath, V. K. N. Lau, D. Gesbert, B. D. Rao, and M. Andrews, "An overview of limited feedback in wireless communication systems," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 8, pp. 1341–1365, Oct. 2008.

[21] Z. Gao, L. Dai, Z. Wang, and S. Chen, "Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6169–6183, Dec. 2015.

[22] X. Rao and V. K. N. Lau, "Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3261–3271, June 2014.

[23] P. Kuo, H. T. Kung, and P. Ting, "Compressive sensing based channel feedback protocols for spatially-correlated massive antenna arrays," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Shanghai, China, Apr. 2012, pp. 492–497.

[24] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acd. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.

[25] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[26] R. Ghanaatian, V. Jamali, A. Burg, and R. Schober, "Feedback-aware precoding for millimeter wave massive MIMO systems," in *Proc. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Istanbul, Turkey, Sept. 2019, pp. 1–7.

[27] S. S. Nair and S. Bhashyam, "Hybrid beamforming in MU-MIMO using partial interfering beam feedback," *IEEE Commun. Lett.*, (early access) 2020.

[28] M. R. Castellanos, V. Raghavan, J. H. Ryu, O. H. Koymen, J. Li, D. J. Love, and B. Peleato, "Channel-reconstruction-based hybrid precoding for millimeter-wave multi-user MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 2, pp. 383–398, May 2018.

[29] A. Alkhateeb, G. Leus, and R. W. Heath, "Limited feedback hybrid precoding for multi-user millimeter wave systems," *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 6481–6494, Nov. 2015.

[30] G. Dietl and G. Bauch, "Linear precoding in the downlink of limited feedback multiuser MIMO systems," in *IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, Dec. 2007, pp. 4359–4364.

[31] J. Jang, H. Lee, S. Hwang, H. Ren, and I. Lee, "Deep learning-based limited feedback designs for MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 4, pp. 558–561, Apr. 2020.

[32] C. Wen, W. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748–751, Oct. 2018.

[33] C. Lu, W. Xu, H. Shen, J. Zhu, and K. Wang, "MIMO channel information feedback using deep recurrent network," *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 188–191, Jan. 2019.

[34] J. Guo, X. Yang, C.-K. Wen, S. Jin, and G. Y. Li, "DL-based CSI feedback and cooperative recovery in massive MIMO," Mar. 2020. [Online]. Available: https://arxiv.org/abs/2003.03303

[35] Q. Yang, M. B. Mashhadi, and D. Gündüz, "Distributed deep convolutional compression for massive MIMO CSI feedback," Mar. 2020. [Online]. Available: https://arxiv.org/abs/2003.04684

[36] G. Caire, N. Jindal, M. Kobayashi, and N. Ravindran, "Multiuser MIMO achievable rates with downlink training and channel state feedback," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2845–2866, June 2010.

[37] F. Sohrabi and W. Yu, "Hybrid digital and analog beamforming design for large-scale antenna arrays," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 501–513, Jan. 2016.

[38] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," July 2012. [Online]. Available: https://arxiv.org/abs/1207.0580

[39] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," Mar. 2016. [Online]. Available: https://arxiv.org/abs/1603.04467

[40] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," Mar. 2015. [Online]. Available: https://arxiv.org/abs/1502.03167