Cascaded Deep Neural Network Based Adaptive Precoding for Distributed Massive MIMO Systems

Lijun GE [1], *Shixun NIU* [1], *Chenpeng SHI* [1], *Yuchuan GUO* [1], *Gaojie CHEN* [2]

[1] School of Electronics and Information Engineering, Tianjin Key Laboratory of Optoelectronic Detection Technology and System, Tiangong University, Tianjin, 300160, China
[2] Institute for Communications Systems, University of Surrey, Guildford, Surrey GU2 7XH, United Kingdom

gelj_001@hotmail.com, n2369619244@163.com, 2130070843@tiangong.edu.cn, 1310433616@qq.com, gaojie.chen@surrey.ac.uk

Submitted September 7, 2023 / Accepted November 24, 2023 / Online first December 9, 2023

Abstract. *In time-division duplex (TDD) distributed large-scale multiple input multiple output (DM-MIMO) systems, the traditional downlink channel precoding method is used to resist inter-user interference (IUI). However, when the Channel State Information (CSI) is incomplete, the performance loss is serious, not only the bit error rate is high, but also the complexity of the traditional precoding algorithm is high. In order to solve these problems, this paper proposes an adaptive precoding framework based on deep learning (DL) for joint training and split application deployment. First, we train a channel emulator deep neural network (CE-DNN) to learn and simulate the transmission process of the wireless communication channel. Then, we concatenate an untrained precoding DNN (P-DNN) with a trained CE-DNN and retrain the cascaded neural network to converge. The last step is to obtain the P-DNN, namely the adaptive precoding network, by dismantling the joint trained network. Simulation results show that, when CSI is imperfect, the proposed method is compared with Tomlinson-Harashima precoding (THP) and block diagonalization (BD) precoding. The proposed method has a lower mean square error (MSE) and higher spectrum efficiency, as well as a bit error rate (BER) performance close to the THP. The source codes and the neural network codes are available on request.*

Keywords

Distributed multiple-input multiple-output (D-MIMO), deep neural network, downlink precoding, channel state information (CSI)

1. Introduction

Massive multiple-input multiple-output (MIMO) systems are equipped with a large-scale array antenna at the base station (BS), which can sufficiently use the space-dimensional wireless resources and greatly improve the spectrum efficiency of mobile communication [1], [2]. Therefore, massive MIMO has become one of the key enablers for high-speed data transmission [3].

Massive MIMO systems are categorized into centralized massive MIMO (CM-MIMO) systems and distributed massive MIMO (DM-MIMO) systems. In traditional CM-MIMO systems, the central BS is equipped with a large number of antennas [4]. At the cell edge, the signal-to-noise ratio (SNR) for users is lower than that of users at the cell center due to the greater distance between the BS antenna array and the cell-edge users, resulting in reduced signal transmission quality [5]. Moreover, in scenarios characterized by heavy traffic and stringent requirements for mobile communication performance, CM-MIMO systems not only encounter challenges related to inter-cell interference and cell handover but also involve high construction costs, as the BS covers a wide area [6], [7].

In DM-MIMO systems, antennas are distributed across each access point (AP), and these APs are linked to the central unit (CU) through a high-capacity backhaul link, such as optical fiber. The DM-MIMO system is user-centric, with small-sized APs that can be randomly deployed in densely populated areas [8]. As a result, when users are near the APs, the spatial macro-diversity gain is amplified, leading to a substantial reduction in path loss.

Full noun	Abbreviation
massive multiple-input multiple-output	MIMO
time-division duplex	TDD
channel emulator deep neural network	CE-DNN
precoding deep neural network	P-DNN
channel state information	CSI
mean square error	MSE
Tomlinson-Harashima precoding	THP
signal to noise ratio	SNR
block diagonalization	BD
least square	LS
linear minimum mean square error	LMMSE

Tab. 1. Abbreviations used in this paper.

DOI: 10.13164/re.2024.0034

The proliferation of a number of APs leads to reduced multi-user interference, resulting in uniform coverage across the entire region and a significantly improved user experience [9–11]. In comparison to CM-MIMO systems, DM-MIMO systems offer advantages such as low transmit power, high multiplexing gain, high spectrum efficiency, extensive coverage area, and simplified network planning [12]. Consequently, the design of precoding for DM-MIMO has become exceedingly challenging. The authors of [13] investigated the radio frequency (RF) mismatch at the user equipment (UE) in DM-MIMO systems employing block diagonalization (BD) precoding. Despite partial calibration at the radio access units (RAUs), the RF mismatch at the UEs still results in negligible performance loss, and reciprocal calibration was performed at the RAUs and UEs. To mitigate computational complexity, the authors of [14] proposed a joint centralized and distributed precoding method, wherein only a few APs perform centralized precoding while the rest engage in distributed precoding. In [15], the authors addressed the channel estimation and feedback challenge by associating it with distributed precoding and utilizing a joint pilot and deep neural network (DNN) design to map the received pilots to the users, followed by directly mapping all user feedback information into the precoding matrix.

Multiple publications have been devoted to the topic of multi-user precoding, which can be categorized into digital precoding and hybrid precoding techniques. In [16], the authors introduced the BD precoding algorithm, which decomposes multi-user MIMO channels into parallel point-to-point single-user MIMO channels. Non-linear precoding schemes, such as the dirty paper coding (DPC) algorithm proposed in [17] and the approximately optimal Tomlinson Harashima precoding (THP) algorithm proposed in [18], often exhibit high computational complexity and are commonly used as theoretical upper bounds for comparative analysis. The underlying principle of these methods is to employ non-linear processing on the data stream being transmitted to eliminate interference from preceding data streams. THP achieves performance comparable to DPC while significantly reducing computational complexity. In addition to the pure digital precoding methods mentioned earlier, digital-analog hybrid precoding enables the sharing of RF links among antennas, leading to considerable reductions in system cost and the number of RF links. Hybrid precoding strikes a balance between hardware implementation complexity and system performance in massive MIMO systems [19]. In [20], the authors proposed an orthogonal matching pursuit (OMP) algorithm, which formulates the design problem of hybrid precoding as the recovery problem of multiple sparse signals. Furthermore, the authors of [21] presented an algorithm based on iterative minimization of phase extraction. By incorporating the concept of alternate minimization during iterations, this algorithm achieves performance closer to pure digital precoding. However, it is worth noting that the algorithm is more complex and challenging to implement.

Recently, deep learning (DL) has attracted much attention in the field of wireless communication due to its powerful nonlinear mapping ability and has been widely used in massive MIMO precoding [22–27]. The authors of [22] proposed a DL-based beamforming scheme in MIMO systems. However, the approach only achieves performance that is somewhat close to the zero-forcing (ZF) scheme and requires training multiple learners. To facilitate decentralized beamforming and reduce communication costs between the APs and the network controller (NC) for hybrid precoding, a study by [23] proposed completely and partially distributed unsupervised DNN architectures. These architectures achieved near-optimal sum rates. By integrating deep learning with beamforming technology, the authors developed a Beamformer Neural Network (BFNN) capable of learning how to optimize the beamformer for maximizing spectral efficiency while considering hardware limitations and imperfect channel state information (CSI) [24]. When training with auxiliary beams, they utilized a multilayer perceptron (MLP) structure assuming perfect CSI knowledge [25]. In [26], the authors introduced a deep reinforcement learning-based hybrid precoding method. However, the DL-based hybrid precoding method did not consider the output limit requirement of the phase shifters. Based on the analyzed relationship between the input and output of DNN, the authors of [27] proposed a classifier-weighted deep neural network (CW-DNN) to recover data without CSI. The DNN channel equalizer proposed in [27] eliminates the need for channel estimation and directly recovers the transmitted data from the received signal, thereby minimizing the impact of distortion during transmission. Inspired by this, we have developed a DNN precoder and a DNN channel emulator, which are applied to the signal transmitter and transmission process respectively, with the aim of further reducing signal distortion. By employing a more intricate mapping relationship and a unique cascaded network design, we have enhanced the performance of the precoding technique and effectively eliminated inter-user interference.

In this paper, we present an adaptive precoding framework built on DL for DM-MIMO systems that allowed for split application deployment and joint training. We connect two neural networks in series, train the cascaded neural network, and finally split the cascaded neural network to get a precoding DNN (P-DNN). Compared with conventional counterparts, the proposed method has better mean square error (MSE) performance and spectrum efficiency, even if the CSI is not perfect. The following is a summary of the major contributions.

- We propose a precoding algorithm for Cascaded DNN to replace traditional precoding algorithms. The proposed DNN-based precoding performs well even when using low-precision channel estimation methods. Through offline training, the DNN can acquire a nonlinear mapping from CSI and user data to precoding,

Symbol	Explanation
$(\cdot)^{\mathrm{T}}$	Transpose operator
$(\cdot)^{\mathrm{H}}$	Hermitian operator
$(\cdot)^{-1}$	Inverse operator
$\|\cdot\|$	Frobenius norm operator
$\mathrm{E}[\cdot]$	Expectation operator

Tab. 2. Notation and symbols used in this paper.

which is especially effective in enhancing precoding performance.

- The channel emulator deep neural network (CE-DNN) trained in this paper can be used to cancel the errors of input channel values with different precision so that it has a certain correction ability. CE-DNN provides favorable conditions for subsequent cascade training of P-DNN.

- Specifically, we simulate MSE, bit error rate (BER), and spectral efficiency (SE) to evaluate the performance of precoding. The low-precision CSI obtained by channel estimation is used as the input of P-DNN, and the BER performance of the proposed method is significantly better than that of the traditional method. The proposed method approaches the optimal precoding performance, and the complexity is significantly lower than that of other methods.

Notations: Abbreviations used in this paper are in Tab. 1. Notations and symbols used in this paper are shown in Tab. 2.

In this paper, the content and structure of the arrangement are as follows: DM-MIMO system and conventional precoding methods are presented in Sec. 2. The innovative precoding scheme built on DL is described in Sec. 3. Simulation results and the complexity of the algorithm are analyzed in Sec. 4. Finally, a conclusion is summarized in Sec. 5.

2. System Model

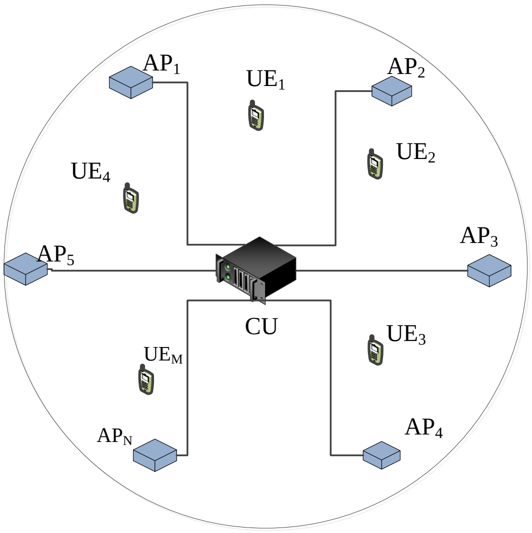

Fig. 1. Distributed massive MIMO systems.

The system model of the downlink single-cell time-division duplex (TDD) DM-MIMO communication system is displayed in Fig. 1, which has N APs with each having K transmitter antennas and M UEs with each having single receiver antennas. A high-capacity backhaul link connects the APs to the CU. Within the range of the DM-MIMO system, each AP and UE are randomly distributed, and they also need that $KN \geq M$. Furthermore, we assume that all APs transmit their data streams to the UEs simultaneously, so the data streams can be written as

$$\mathbf{S} = [s_1, s_2, \ldots, s_m], \quad m = 1, 2, \ldots, M \tag{1}$$

where s_m denotes the data streams to be sent to the UE and the length of s_m is N_{s}, which is normalized so that $\mathrm{E}\left[\mathbf{S}\mathbf{S}^{\mathrm{H}}\right] = \frac{1}{N_{\mathrm{s}}}\mathbf{I}_{N_{\mathrm{s}}}$. Precoding is used across all data streams such that the transmitted signal can be represented as

$$\mathbf{X} = \mathbf{F}\mathbf{S} \tag{2}$$

where $\mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_M] \in \mathbb{C}^{KN \times M}$denotes the precoding matrix of M users, which satisfies the constraints that $\|\mathbf{F}\|^2 \leq N_{\mathrm{s}}$. System modeling of distributed massive MIMO systems is referenced in [28]. The channel model is the same as [26] and fading channels follow a Rayleigh distribution. The APs and the UE downlink channel gains vector is $\mathbf{h}_m$, $\mathbf{h}_m \in \mathbb{C}^{1 \times KN}$, so the total channel matrix can be expressed as

$$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_M] \tag{3}$$

where $\mathbf{H} \in \mathbb{C}^{M \times KN}$. Furthermore, in the data transmission phase, the received signal at the UE can be expressed as

$$\mathbf{y}_m = \sqrt{P}\mathbf{h}_m\mathbf{F}_m\mathbf{s}_m + \sqrt{P}\mathbf{h}_m \sum_{i=1, i\neq m}^{M} \mathbf{F}_i\mathbf{s}_i + \mathbf{n}_m \tag{4}$$

where $\mathbf{n}_m \sim \mathcal{CN}\left(0, \sigma^2\right)$ is the additive white Gaussian noise. The average transmit power on APs is denoted by P. The second term of (4) denotes the interference of the UE from other users. After processing by the receiving end, the final received signal vector can be expressed as

$$\mathbf{y}_m = \sqrt{P}\mathbf{W}_m^{\mathrm{H}}\mathbf{h}_m\mathbf{F}_m\mathbf{s}_m + \sqrt{P}\mathbf{W}_m^{\mathrm{H}}\mathbf{h}_m \sum_{i=1, i\neq m}^{M} \mathbf{F}_i\mathbf{s}_i + \mathbf{W}_m^{\mathrm{H}}\mathbf{n}_m \tag{5}$$

where $\mathbf{W}_m$ denotes the merge matrix of the UE and $\mathbf{W}_m \in \mathbb{C}^{1 \times N_{\mathrm{s}}}$. Therefore, the signal to interference plus noise ratio (SINR) of the UE can be expressed as

$$\mathrm{SINR}_m = \frac{P\left\|\mathbf{W}_m^{\mathrm{H}}\mathbf{h}_m\mathbf{F}_m\right\|^2}{\sigma_n^2 + P\sum_{i\neq m}\left\|\mathbf{W}_m^{\mathrm{H}}\mathbf{h}_m\mathbf{F}_i\right\|^2}. \tag{6}$$

Furthermore, if the bandwidth allocated to the UE is B_m, the total system capacity between APs and multiple UEs can be given as

$$C = \sum_{m=1}^{M} \mathrm{E}\left[B_m \log_2\left(1 + \mathrm{SINR}_m\right)\right]. \tag{7}$$

In a typical multi-user precoding scheme, precoding can be divided into linear digital precoding, non-linear digital precoding, and digital-analog hybrid precoding. The performance of the precoding method is significantly dependent on accurate downlink CSI. In a TDD system, the transmitter can use the channel reciprocity of TDD to quickly obtain downlink CSI by uplink CSI to calculate the precoding matrix. For DM-MIMO systems, these multi-user precoding methods are equally applicable.

The BD precoding algorithm, which is the most widely used linear-digital precoding technique, uses singular value decomposition (SVD) to obtain the orthogonal basis of each user concerning other users so that the multi-user MIMO channel can be divided into parallel point-to-point single-user MIMO channels that do not interfere with one another [16]. For m-th UE, the SVD of $\mathbf{h}_m$ can be expressed as

$$\mathbf{h}_m = \mathbf{U}_m \mathbf{\Sigma}_m \mathbf{V}_m^{\mathrm{H}} = \mathbf{U}_m \mathbf{\Sigma}_m \left[\mathbf{V}_m^{(1)} \mathbf{V}_m^{(0)} \right]^{\mathrm{H}} \tag{8}$$

where $\mathbf{U}_m$ and $\mathbf{V}_m$ are unitary matrices, and $\mathbf{\Sigma}_m$ is a diagonal matrix containing the singular values of $\mathbf{h}_m$. $\mathbf{V}_m^{(1)}$ and $\mathbf{V}_m^{(0)}$ are the right singular matrices corresponding to the non-zero and zero singular values of $\mathbf{h}_m$, respectively. Therefore, the precoding matrix $\mathbf{F}_m$ of the m-th single-antenna UE can be designed as the first column of the $\mathbf{V}_m^{(0)}$, which can be expressed as

$$\mathbf{F}^{\mathrm{BD}} = \mathbf{V}_m^{(0)} (:, 1) . \tag{9}$$

There is some performance loss because BD precoding only suppresses interference between multiple users without considering the effect of noise, so the number of UEs it can support is constrained by the number of transmitting antennas of the APs and the number of receiving antennas used by each user. In addition to linear digital precoding schemes, there are some nonlinear digital precoding schemes with better performance, for example, the best DPC algorithm and the approximate optimal THP algorithm. However, these schemes usually have high computational complexity, so they are often used as performance reference algorithms for precoding.

The above precoding design is carried out in the digital domain. As the number of antennas increases, achieving pure digital precoding becomes difficult, necessitating the use of a digital-analog hybrid precoding scheme. In the case of hybrid precoding, the precoding matrix and merge matrix can be expressed as

$$\mathbf{F}_{\mathrm{HY}} = \mathbf{F}_{\mathrm{D}} \cdot \mathbf{F}_{\mathrm{A}}, \tag{10}$$

$$\mathbf{W}_{\mathrm{HY}} = \mathbf{W}_{\mathrm{A}} \cdot \mathbf{W}_{\mathrm{D}}, \tag{11}$$

where $\mathbf{F}_{\mathrm{D}}$ and $\mathbf{F}_{\mathrm{A}}$ denote the digital precoding matrix and analog precoding matrix, respectively. $\mathbf{W}_{\mathrm{A}}$ and $\mathbf{W}_{\mathrm{D}}$ denote the analog merge matrix and digital merge matrix, respectively. It is assumed that the RF link at the transmitting end is N_{RF}, then $\mathbf{F}_{\mathrm{D}} \in \mathbb{C}^{KN \times N_{\mathrm{RF}}}$, $\mathbf{F}_{\mathrm{A}} \in \mathbb{C}^{N_{\mathrm{RF}} \times M}$, $\mathbf{W}_{\mathrm{A}} \in \mathbb{C}^{M \times N_{\mathrm{RF}}}$ and $\mathbf{W}_{\mathrm{D}} \in \mathbb{C}^{N_{\mathrm{RF}} \times M}$.

The performance of the pure digital precoding algorithm is the best, but it also brings high costs and high power consumption, so its implementation is difficult. The analog precoding algorithm is relatively easy to implement, but the performance is not ideal, and it does not support the simultaneous transmission of multiple data streams. The hybrid precoding algorithm achieves a balance between performance and power consumption.

3. Proposed Precoding Based on Deep Learning

DL is a typical learning framework that relies on massive data. It automatically adjusts the model structure through end-to-end optimization and flexible adjustment of parameters. It can replace manual methods to extract features from the original data. In DM-MIMO systems, due to the distributed antenna structure at the transmitting end, the data transmission process becomes more variable, and the preprocessing of transmitted data becomes more challenging. Therefore, the DL-based precoding method has great potential to achieve better performance.

3.1 Proposed Deep Learning Architecture

The most common DL framework, DNN, can be conceptualized as an MLP. DNN is comprised of multiple fully connected (FC) layers or partially connected layers, which provide powerful learning and nonlinear mapping capabilities. Dropout is a method used to optimize training in deep learning. It achieves this by deactivating some neurons' activation values within a hidden layer with a certain probability, effectively generating a subset of the connection layer. This process makes the model more generalized and helps prevent overfitting by reducing reliance on specific local features. Each hidden layer in a DNN contains a number of units, and activation functions enable the production of output in response to the inputs of these units. The rectified linear unit (ReLU) function and the sigmoid function are the two most commonly used nonlinear activation functions for DNN, each specified as

$$\mathrm{ReLU}\,(a) = \max(0, a), \tag{12}$$

$$\mathrm{Sigmoid}\,(a) = \frac{1}{1 + \mathrm{e}^{-a}} \tag{13}$$

where a denotes the input data of the activation function, which is composed of the weights and biases of the same hidden layer in a DNN. Therefore, it can be further represented as

$$a = w^{(l)^{\mathrm{T}}} I^{(l)} + b^{(l)} \tag{14}$$

where $w^{(l)^{\mathrm{T}}}$, $I^{(l)}$, and $b^{(l)}$ represent the connection weight matrix, input value, and bias matrix of the l-th layer of the

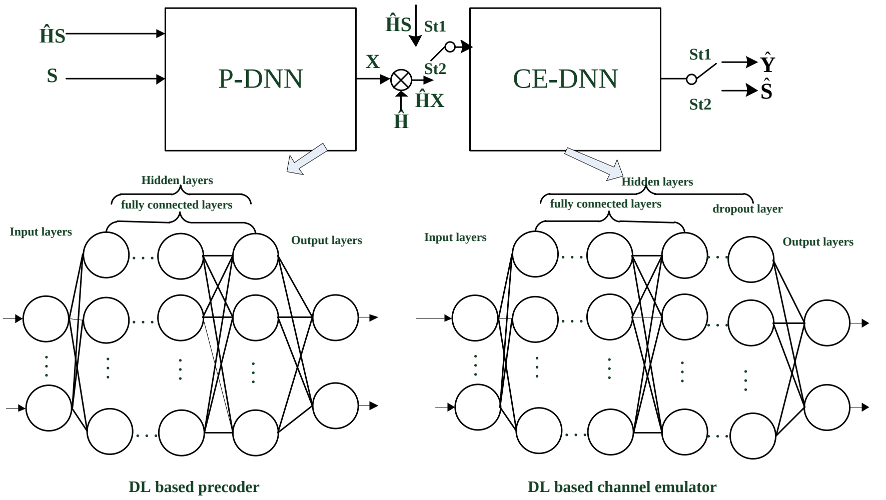

Fig. 2. DNN architecture in the proposed scheme.

network, respectively. I and O represent the input data and output data of the network respectively, and the mapping operation can be expressed as

$$O = f(I, w, b) = f^{(l-1)}\left(f^{(l-2)}\left(\cdots f^{(1)}(I)\right)\right) \tag{15}$$

where $f(\cdot)$ denotes the activation function. We establish a DNN framework, illustrated in Fig. 2, to accomplish adaptive precoding. The whole network is composed of two DNN modules, consisting of a DNN-based channel emulator named CE-DNN and a DNN-based precoder named P-DNN.

The CE-DNN network comprises six layers, with the input layer being an FC layer consisting of 128 neurons. The length of each training sequence depends on the network's dimension. The subsequent three hidden layers, each containing 500 neurons, are also FC layers designed to capture channel characteristics. A dropout layer with 256 neurons randomly deactivates some neurons to enhance data depth and prevent overfitting during training. Additionally, the output layer consists of 128 neurons, configured to produce the expected output signals.

In the P-DNN, the network structure is similar to that of the DL-based channel emulator, except for the noise layer and the dropout layer. The input layer consists of 256 neurons, and the output layer consists of 128 neurons. The hidden layers consist of 4 FC layers, each with 500 neurons. Additionally, the ReLU function is used as the activation function for both the input layer and the hidden layers. However, assuming that the data stream of each UE is a single data, to enforce the power constraint in the output layer of the DL-based precoder, an exclusive activation is intended as

$$f(I) = \min\left(\max(I, 0), N_\mathrm{s}\right). \tag{16}$$

3.2 Learning Policy

There are training sets and testing sets for the experimental sample sets. MATLAB R2021B is used to generate the channel matrix $\mathbf{H}$ and the user data stream $\mathbf{S}$, and then the received signal $\mathbf{Y}$ of the UEs is obtained through the channel emulator. Therefore, the sample set can be expressed as $(\mathbf{S}, \mathbf{H}, \mathbf{Y})$.

The network training process is divided into two stages, as demonstrated in Fig. 2.

Stage 1: CE-DNN is trained as an independent network, with $\mathbf{S}$ and $\hat{\mathbf{H}}$ as inputs, and $\mathbf{Y}$ as outputs. The stochastic gradient descent (SGD) algorithm is used to reduce the value of the loss function during training, which is expressed as

$$loss_\mathrm{ce} = \frac{1}{2}\sum_{i=1}^{N_\mathrm{b}}\left(\hat{\mathbf{Y}}_i - \mathbf{Y}_i\right)^2 \tag{17}$$

where $\hat{\mathbf{Y}}_i$ denotes the output of the CE-DNN, and N_b denotes the number of samples used in one training. When the training phase is completed, assuming that the channel matrix is obtained through uplink channel estimation by utilizing the inherent channel reciprocity of TDD, such as the (LS) algorithm, linear minimum mean square error (LMMSE) algorithm, and OMP algorithm, so the network can get the mapping function of the CE-DNN as $g_\mathrm{ce}(.)$, which can be defined as

$$\hat{\mathbf{Y}} = g_\mathrm{ce}\left(\mathbf{S}, \hat{\mathbf{H}}\right) \tag{18}$$

where $\hat{\mathbf{Y}}$ and $\hat{\mathbf{H}}$ denote the estimated value of the received signal generated by the CE-DNN based channel emulator and the channel matrix obtained by channel estimation algorithms, respectively.

MSE is used to measure the performance of the CE-DNN. The MSE of the estimated amount of each SNR value under various channel estimation algorithms can be approximated as

$$MSE_\mathrm{CE} = \frac{1}{MN_\mathrm{p}}\sum_{i=1}^{N_\mathrm{p}}\left(\widehat{\mathbf{Y}}_i - \mathbf{Y}_i\right)^2 \tag{19}$$

where N_p denotes the total number of samples.

Stage 2: Two DNNs are linked together to create a cascaded network for joint training. The output of the P-DNN serves as the input for the CE-DNN, and the parameters of the CE-DNN remain fixed. The parameters of the P-DNN in the cascaded network are adjusted through training. The input and output data of the cascaded network are denoted as $(\mathbf{S}, \mathbf{H})$ and $\hat{\mathbf{S}}$, respectively. Therefore, the mapping function of the cascaded network, denoted as $g_{\text{ca}}(.)$, includes the mapping of the P-DNN, denoted as $g_{\text{p}}(.)$, and the mapping function of the CE-DNN, denoted as $g_{\text{ce}}(.)$. So it can be written as

$$\hat{\mathbf{S}} = g_{\text{ca}}\left(\mathbf{S}, \hat{\mathbf{H}}\right) = g_{\text{ce}}\left(g_{\text{p}}\left(\mathbf{S}, \hat{\mathbf{H}}\right), \hat{\mathbf{H}}\right). \tag{20}$$

In Stage 2, the cascaded network undergoes training using the same process as Stage 1. Once the training of the cascaded network is completed, it is disassembled, and the P-DNN becomes an independent network. The output of P-DNN is precoded data, and the mapping function of P-DNN can be expressed as

$$\mathbf{X} = g_{\text{p}}\left(\mathbf{S}, \hat{\mathbf{H}}\right). \tag{21}$$

When utilizing the test set to gauge how well the suggested precoding framework performs, we expect that after the precoded data $\mathbf{X}$ through the actual channel, the received data $\hat{\mathbf{S}}$ obtained in the UEs can be as close as possible to the transmitted user data $\mathbf{S}$ in the APs. The process can be shown as

$$\hat{\mathbf{S}} = \mathbf{HX} + \mathbf{N} = \mathbf{H}g_{\text{p}}\left(\mathbf{S}, \hat{\mathbf{H}}\right) + \mathbf{N}. \tag{22}$$

$\mathbf{S}$ and $\mathbf{H}$ are used as the input of P-DNN. After the input signal goes through the P-DNN forward propagation, the output of P-DNN, which is the precoded signal, can be obtained. The received data obtained by each UE only contains its own data stream and does not include other UEs' data streams, thus eliminating IUI. The entire procedure for training the precoding algorithm is displayed in Algorithm 1.

Additionally, we introduce the MSE and the BER to examine the precoding performance of the DL-based precoding approach, which can be provided as

$$MSE_{\text{P}} = \frac{1}{MN_{\text{p}}}\sum_{i=1}^{N_{\text{p}}}\left(\widehat{S}_i - S_i\right)^2, \tag{23}$$

$$BER = \frac{1}{B}\sum_{k=1}^{B}\left|\hat{b}_k - b_k\right| \tag{24}$$

where $\widehat{S}_i$ and S_i represent the received data of the UEs and the original user data of the UEs, respectively. The total amount of bits transmitted across the channel is indicated by the value B, b_k, and $\hat{b}_k$ are the transmitted bit streams of APs and the received bit streams from the UEs, respectively.

4. Simulation Results and Complexity Analysis

4.1 Simulation Results

The performance analysis of CE-DNN is presented under various channel estimation algorithms, encompassing the LS algorithm, LMMSE algorithm, and true channel value. The established distributed massive MIMO system has 16 APs with 4 transmitting antennas and 64 UEs with a single receiving antenna. The number of transmitting antennas and receiving antennas is both 64. Rayleigh fading channel is used as the channel. The transmitted signal is multiplied by the Rayleigh channel coefficient [29], and then a white Gaussian noise signal with different SNR powers is added to the signal. The simulation data is generated in MATLAB R2021B. Neural network training is performed using TensorFlow, Pycharm, and Python. The experiments are conducted on a computer with an Intel Core i7 CPU (1.5 GHz). The digital modulation method is 64 QAM or 32 QAM. We set each AP to transmit only one symbol per antenna during the coherence time. System parameters are listed in Tab. 3.

We developed our DL-based solution using Python language in the Pycharm environment. We used a well-known DL-based network using the basic DNN network structure, which is the most general structure in the deep learning framework and can be thought of as the MLP. Through linear accumulation and nonlinear activation of multiple hidden layers, the prediction result is obtained from the output layer. In this paper, both CE-DNN and P-DNN are represented by six-layer DNN models, each consisting of an input layer, four hidden layers, and an output layer. The dropout layer in CE-DNN has a neuron rejection rate of 0.1. Furthermore, the output layer of the DNN model uses the linear function, while the activation function for the hidden layers is ReLU. We generated 200,000 sample sets using MATLAB, with 80% allocated for training and 20% for testing. The batch size for neural network training is 100. The optimization algorithm of the neural network uses SGD. For the neural network to achieve convergence the learning rate is set to 0.001. Multiple iterations were necessary to obtain all simulation results.

Parameters	Values
Channel type	AWGN, Rayleigh fading
The number of AP	16
The number of UE	64
The number of antennas per AP	4
The number of antennas per UE	1
Modulation	32, 64 QAM
Batch size	100
Neuron rejection rate	0.1
Learning rate	0.001

Tab. 3. Simulation parameters of the distributed massive MIMO system.

In Fig. 3, the two-channel estimation methods, LS and LMMSE, are compared with the true channel value under various SNR conditions. The results demonstrate that at low SNRs, the MSE performance of the LS algorithm is significantly lower than that of the LMMSE algorithm. It is due to the performance loss caused by ignoring the influence of noise. However, when the SNR is 30 dB, the NMSE performance of the LS method is superior to that of the LMMSE method because the LMMSE uses statistics instead of specific parameters, resulting in errors. Figure 4 provides an MSE comparison of the proposed CE-DNN using three different accuracies of the channel value in Fig. 3. With an increase in SNR, the MSE value of the channel emulator decreases, as shown in Fig. 4. When CE-DNN inputs channel values of different accuracy, the MSE performance will also change with the accuracy of the channel value. However, CE-DNN has similar MSE performance in Fig. 4, despite the difference in MSE performance of the channel value in Fig. 3. We can conclude that CE-DNN, as a channel emulator, does not heavily depend on the accuracy of the input channel values. It exhibits strong generalization and correction capabilities.

In Fig. 5, the convergence effectiveness of the CE-DNN algorithm is evaluated with a learning rate of 0.001 and 64 QAM modulation. The MSE demonstrates how well the proposed channel emulator converges as CE-DNN is trained for various durations. It is illustrated in Fig. 5 that adding training epochs improves the convergence of MSE, stabilizing after approximately 180 iterations in CE-DNN training. This figure depicts the convergence behavior of the proposed CE-DNN.

Figure 6 shows the performance of P-DNN as a precoder after being jointly trained with CE-DNN. The MSE is used to reflect the similarity between the original user data signal and the signal received by the user after the precoded signal generated by the P-DNN passes through the actual channel. The outcome of Fig. 6 demonstrates that as the SNR increases, the received signal generated by the precoding signal through the actual channel gets closer to the original user data signal. Even though the inputs of P-DNN are channel values with different precisions, P-DNN exhibits similar MSE performance. When the inputs of P-DNN are

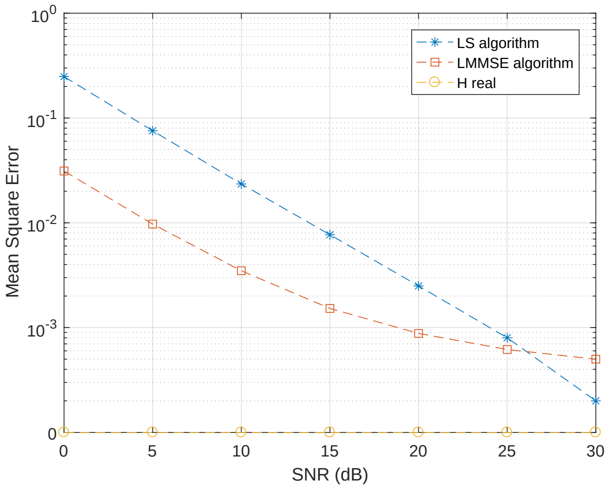

Fig. 3. NMSE performance for varying channel estimation algorithms.

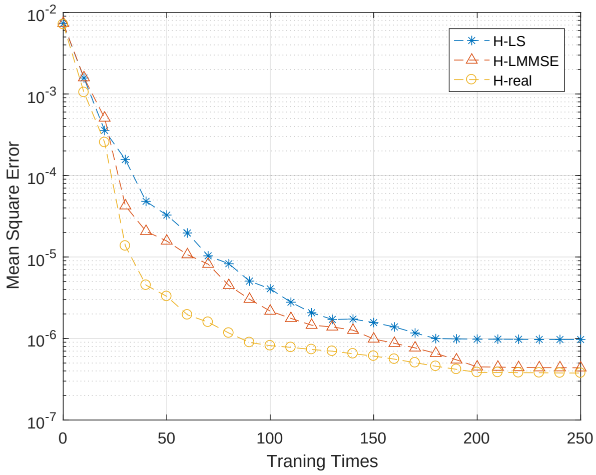

Fig. 5. Convergence performance of the proposed CE-DNN for varying channel estimation in the case of "64 QAM, learning rate = 0.001".

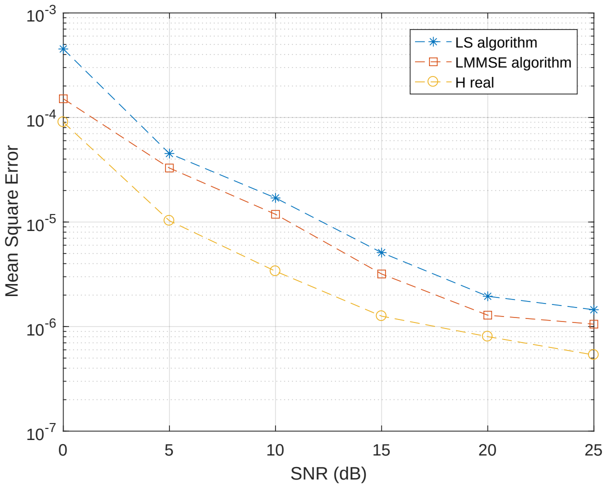

Fig. 4. MSE performance of the proposed CE-DNN for varying channel estimation algorithms at 64 QAM.

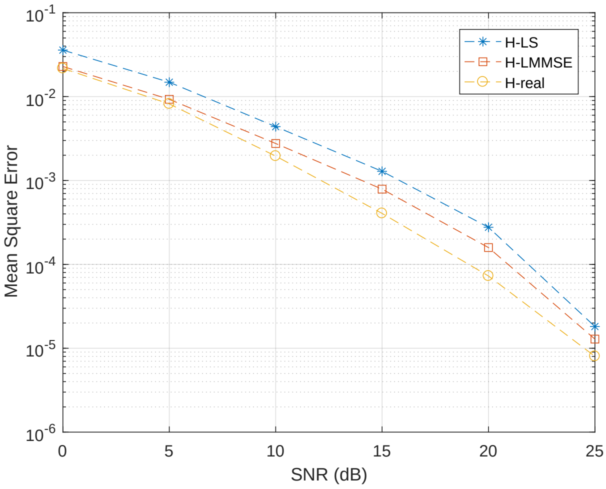

Fig. 6. MSE performance of the proposed P-DNN for varying channel estimation algorithms at 64 QAM.

channel values based on the LS channel estimation algorithm, LMMSE channel estimation algorithm, and real channel, under SNR is 25 dB, the MSE performance of P-DNN can achieve 1.97×10^{-5}, 1.08×10^{-5}, and 8.01×10^{-6}, respectively. Compared with 32 QAM, 64 QAM is a high-order modulation mode, which transmits more bits, and it also requires higher channel conditions. Therefore, Figure 7 shows that the MSE of the proposed P-DNN is smaller for different channel estimation algorithms at 32 QAM.

Figure 8 shows the MSE performance of P-DNN obtained through multiple rounds of training to demonstrate its convergence performance. Compared with CE-DNN, P-DNN requires a deeper exploration of the non-linear mapping relationship between user data, CSI, and precoded signals, thus necessitating more extensive training. As depicted in the figure, the P-DNN begins to stabilize approximately around the 800th training iteration.

We compared the bit error rate (BER) performance of the P-DNN-based precoding scheme to that of the linear fully digital BD-based precoding scheme [16] and the non-linear fully digital TH-based precoding scheme [18] to assess the superiority of the proposed approach. When 64 QAM is chosen as the digital modulation mode of the baseband signal, the BER of the system is larger than that of 32 QAM. This is because high-order digital modulation has higher requirements for the accuracy of signal recovery. As shown in Fig. 9 and Fig. 10, when the channel estimation value is obtained using the LS channel estimation algorithm, the proposed method outperforms the conventional linear fully digital precoding scheme, and its BER performance is comparable to the fully digital non-linear TH-based precoding scheme. THP adopts the idea of DPC and has approximately optimal precoding performance, but THP has high complexity. The simulation results prove that the P-DNN-based precoding has better performance, which is due to the superior representation capabilities of DL. This means that the precoding in distributed massive MIMO can be solved with DL.

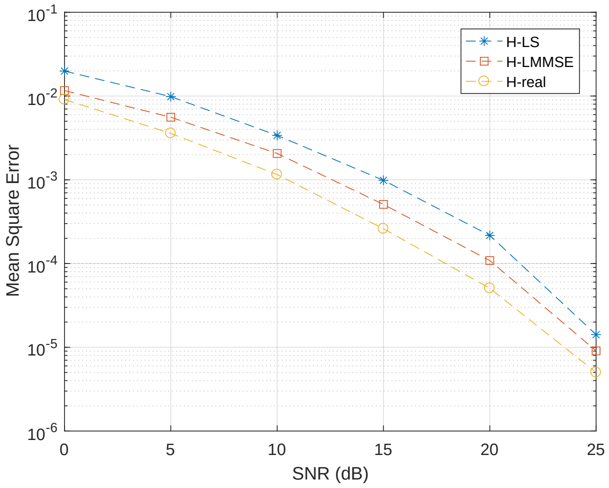

Fig. 7. MSE performance of the proposed P-DNN for varying channel estimation algorithms at 32 QAM.

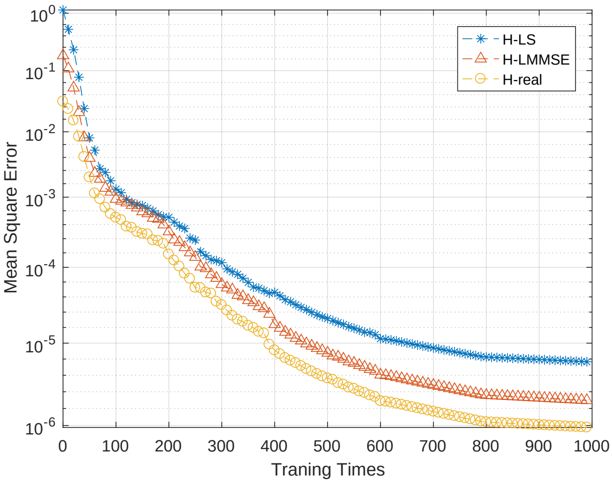

Fig. 8. Convergence performance of the proposed P-DNN for varying channel estimation in the case of "64 QAM, learning rate = 0.001".

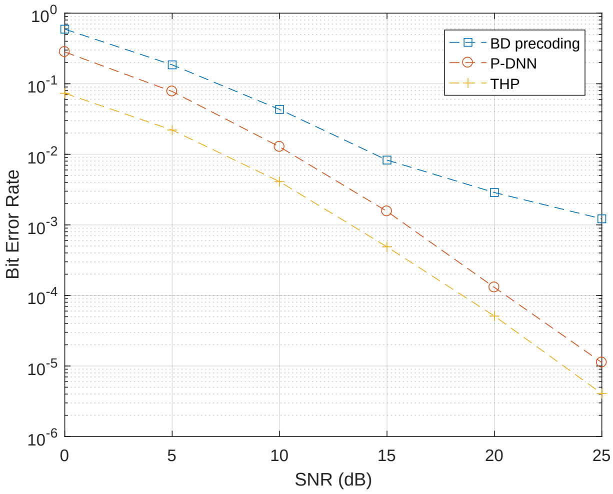

Fig. 9. BER performance for varying precoding schemes at 32 QAM.

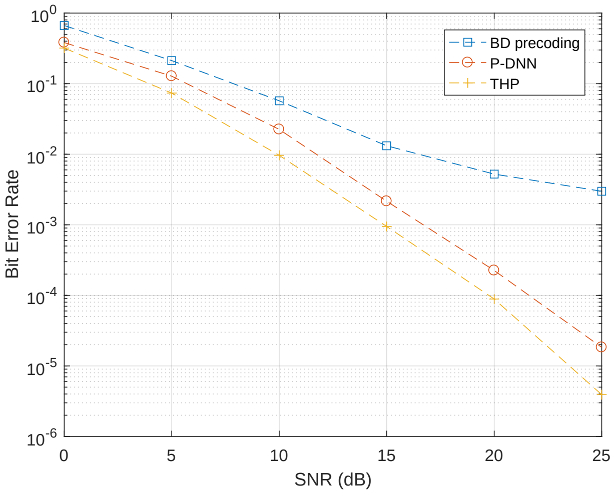

Fig. 10. BER performance for varying precoding schemes at 64 QAM.

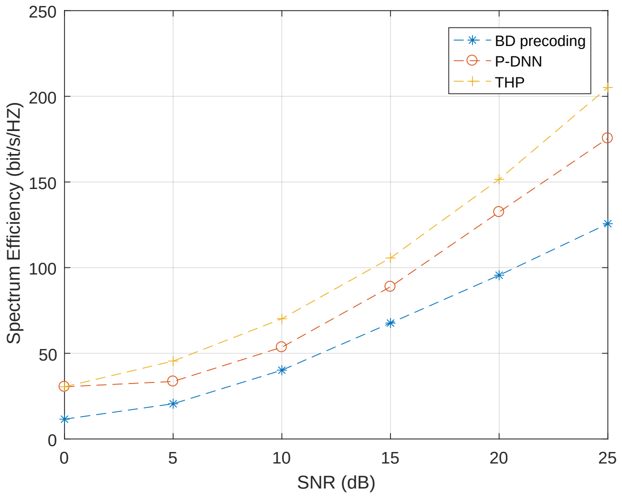

Fig. 11. Spectrum efficiency of different algorithms at different SNRs.

As demonstrated in Fig. 11, we display the spectrum efficiency performance of the P-DNN-based precoding technique against the SNR, the linear fully digital BD-based precoding scheme, and the non-linear fully digital TH-based precoding scheme. In all the schemes, it is evident from Fig. 11 that the spectrum efficiency increases as the SNR rises. Furthermore, it is clear from Fig. 11 that the proposed P-DNN-based precoding scheme outperforms other methods by a wide margin and exhibits a spectrum efficiency comparable to the TH-based precoding scheme, thanks to the outstanding mapping and learning capabilities of DL. Additionally, the performance disparity between the DL-based system and other approaches widens as the SNR increases. This improved performance highlights the value of the suggested precoding technique.

4.2 Complexity Analysis

For a more quantitative analysis, let us consider the number of multiply-accumulate operations of the DNN. This can be easily observed to be proportional to the number of channels. In the offline training phase, the computational complexity was not significant since the required time was usually not strictly bounded. Therefore, the computational complexity of the proposed Cascaded DNN does not need to consider the complexity of CE-DNN and cascaded DNN. The computational complexity of the proposed method was the complexity of P-DNN. The number of multiplications and additions of complex numbers was used to define the computing complexity in terms of the necessary floating-point operations (FLOPs). The complexity of P-DNN is $O(KN(KN+M))$. It was compared with the computational complexity of the traditional nonlinear precoding algorithm THP algorithm and BD algorithm [30], as shown in Tab. 4. In a multiuser configuration, the amount of transmit antennas is represented by KN, and the amount of receive antennas is represented by M.

Precoding method	Complexity
BD	$\frac{4}{3}M^3+\sum_{k=1}^{n} O\left(N^6\right)$
THP	$O\left(N^4\right)$
P-DNN	$O(KN(KN+M))$

Tab. 4. Computational complexity of methods.

5. Conclusions

In this paper, we propose an adaptive precoding framework based on DL for distributed massive MIMO systems. Firstly, the CE-DNN is trained to learn and simulate the transmission process of the wireless communication channel, enabling it to effectively correct the input channel values. Secondly, the untrained P-DNN and CE-DNN are cascaded and then trained, resulting in the P-DNN with precoding ability obtained through splitting. Simulation results show that P-DNN can achieve adaptive precoding even when the CSI obtained by channel estimation is poor. Compared with traditional precoding technology, P-DNN not only reduces the BER and achieves nearly optimal precoding performance, but also reduces algorithm complexity. The DNN network used in this paper is a more general and basic neural network. More complex neural networks can be employed to enhance the performance of digital precoding in the future. Another promising direction is to implement hybrid precoding based on the neural network architecture presented in this paper, aiming to reduce RF links overhead in massive MIMO systems.

Acknowledgments

This work has supported by the National Natural Science Foundation of China (61302062, 61601325, 61901298), the Natural Science Foundation of Tianjin (13JCQNJC00900, 20JCQNJC00300) in China.

References

[1] MARZETTA, T. L. Noncooperative cellular wireless with unlimited numbers of base station antennas. *IEEE Transactions on Wireless Communications*, 2010, vol. 9, no. 11, p. 3590–3600. DOI: 10.1109/TWC.2010.092810.091092

[2] RUSEK, F., PERSSON, D., LAU, B. K., et al. Scaling up MIMO: Opportunities and challenges with very large arrays. *IEEE Signal Processing Magazine*, 2013, vol. 30, no. 1, p. 40–60. DOI: 10.1109/MSP.2011.2178495

[3] AGIWAL, M., ROY, A., SAXENA, N. Next generation 5G wireless networks: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 2016, vol. 18, no. 3, p. 1617–1655. DOI: 10.1109/COMST.2016.2532458

[4] KAMGA, G. N., XIA, M., AISSA, S. Spectral-efficiency analysis of massive MIMO systems in centralized and distributed schemes. *IEEE Transactions on Communications*, 2016, vol. 64, no. 5, p. 1930–1941. DOI: 10.1109/TCOMM.2016.2519513

[5] SOOMRO, H., HABIB, A. Impact of remote radio head positions on the performance of distributed massive MIMO system with user mobility. In *15th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. Islamabad (Pakistan), 2018, p. 789–794. DOI: 10.1109/IBCAST.2018.8312313

[6] LU, L., LI, G. Y., SWINDLEHURST, A. L., et al. An overview of massive MIMO: Benefits and challenges. *IEEE Journal of Selected Topics in Signal Processing*, 2014, vol. 8, no. 5, p. 742–758. DOI: 10.1109/JSTSP.2014.2317671

[7] GUPTA, A., JHA, R. K. A survey of 5G network: Architecture and emerging technologies. *IEEE Access*, 2015, vol. 3, p. 1206–1232. DOI: 10.1109/ACCESS.2015.2461602

[8] BUZZI, S., DANDREA, C., ZAPPONE, A., et al. User-centric 5G cellular networks: Resource allocation and comparison with the cell-free massive MIMO approach. *IEEE Transactions on Wireless Communications*, 2020, vol. 19, no. 2, p. 1250–1264. DOI: 10.1109/TWC.2019.2952117

[9] REN, H., LIU, N., PAN, C., et al. Energy efficiency optimization for MIMO distributed antenna systems. *IEEE Transactions on Vehicular Technology*, 2017, vol. 66, no. 3, p. 2276–2288. DOI: 10.1109/TVT.2016.2574899

[10] LI, Y., FAN, P., LIU, L., et al. Distributed MIMO precoding for in-band full-duplex wireless backhaul in heterogeneous networks. *IEEE Transactions on Vehicular Technology*, 2018, vol. 67, no. 3, p. 2064–2076. DOI: 10.1109/TVT.2017.2713413

[11] YANG, T. Distributed MIMO broadcasting: Reverse compute-and-forward and signal-space alignment. *IEEE Transactions on Wireless Communications*, 2017, vol. 16, no. 1, p. 581–593. DOI: 10.1109/TWC.2016.2626360

[12] BRANDT, R., BENGTSSON, M. Distributed CSI acquisition and coordinated precoding for TDD multicell MIMO systems. *IEEE Transactions on Vehicular Technology*, 2016, vol. 65, no. 5, p. 2890–2906. DOI: 10.1109/TVT.2015.2432051

[13] YANG, M., WEI, H., WANG, D. Analysis of BD precoding for distributed large-scale MIMO systems with RF mismatches at UEs. In *IEEE/CIC International Conference on Communications in China - Workshops (CIC/ICCC)*. Shenzhen (China), 2015, p. 48–51. DOI: 10.1109/ICCChinaW.2015.7961578

[14] KIM, M., CHOI, I. K., HONG, S. E., et al. Joint centralized and distributed precoding in scalable cell-free massive MIMO systems. In *13th International Conference on Information and Communication Technology Convergence (ICTC)*. Jeju Island (Korea), 2022, p. 1254–1257. DOI: 10.1109/ICTC55196.2022.9952405

[15] SOHRABI, F., ATTIAH, K., YU, W. Deep learning for distributed channel feedback and multiuser precoding in FDD massive MIMO. *IEEE Transactions on Wireless Communications*, 2021, vol. 20, no. 7, p. 4044–4057. DOI: 10.1109/TWC.2021.3055202

[16] KHAN, M. H. A., CHO, K. M., LEE, M. H., et al. A simple block diagonal precoding for multi-user MIMO broadcast channels. *Eurasip Journal on Wireless Communications and Networking*, 2014, vol. 2014, no. 95, p. 1–8. DOI: 10.1186/1687-1499-2014-95

[17] ELLIOTT, R. C., KRZYMIEN, W. A. Downlink scheduling via genetic algorithms for multiuser single-carrier and multicarrier MIMO systems with dirty paper coding. *IEEE Transactions on Vehicular Technology*, 2009, vol. 58, no. 7, p. 3247–3262. DOI: 10.1109/TVT.2008.2009059

[18] TSENG, F. S., WANG, Y. C. Codebook size design for RVQ-based Tomlinson-Harashima precoded MIMO broadcast channels. *IEEE Transactions on Vehicular Technology*, 2015, vol. 64, no. 10, p. 4876–4881. DOI: 10.1109/TVT.2014.2367577

[19] LIANG, L., XU, W., DONG, X. Low-complexity hybrid precoding in massive multiuser MIMO systems. *IEEE Wireless Communications Letters*, 2014, vol. 3, no. 6, p. 653–656. DOI: 10.1109/LWC.2014.2363831

[20] AYACH, O. E., RAJAGOPAL, S., ABU-SURRA, S., et al. Spatially sparse precoding in millimeter wave MIMO systems. *IEEE Transactions on Wireless Communications*, 2014, vol. 13, no. 3, p. 1499–1513. DOI: 10.1109/TWC.2014.011714.130846

[21] YU, X., SHEN, J. C., ZHANG, J., et al. Alternating minimization algorithms for hybrid precoding in millimeter wave MIMO systems. *IEEE Journal of Selected Topics in Signal Processing*, 2016, vol. 10, no. 3, p. 485–500. DOI: 10.1109/JSTSP.2016.2523903

[22] KERRET, P. D., GESBERT, D. Robust decentralized joint precoding using team deep neural network. In *15th International Symposium on Wireless Communication Systems (ISWCS)*. Lisbon (Portugal), 2018, p. 1–5. DOI: 10.1109/ISWCS.2018.8491209

[23] HOJATIAN, H., NADAL, J., FRIGON, J. F., et al. Decentralized beamforming for cell-free massive MIMO with unsupervised learning. *IEEE Communications Letters*, 2022, vol. 26, no. 5, p. 1042–1046. DOI: 10.1109/LCOMM.2022.3157161

[24] LIN, T., ZHU, Y. Beamforming design for large-scale antenna arrays using deep learning. *IEEE Wireless Communications Letters*, 2020, vol. 9, no. 1, p. 103–107. DOI: 10.1109/LWC.2019.2943466

[25] ELBIR, A. M., PAPAZAFEIROPOULOS, A. K. Hybrid precoding for multiuser millimeter wave massive MIMO systems: A deep learning approach. *IEEE Transactions on Vehicular Technology*, 2020, vol. 69, no. 1, p. 552–563. DOI: 10.1109/TVT.2019.2951501

[26] WANG, Q., FENG, K., LI, X., et al. PrecoderNet: Hybrid beamforming for millimeter wave systems with deep reinforcement learning. *IEEE Wireless Communications Letters*, 2020, vol. 9, no. 10, p. 1677–1681. DOI: 10.1109/LWC.2020.3001121

[27] GE, L., QI, C., GUO, Y., et al. Classification weighted deep neural network based channel equalization for massive MIMO-OFDM systems. *Radioengineering*, 2022, vol. 31, no. 3, p. 346–356. DOI: 10.13164/re.2022.0346

[28] MINASIAN, A., SHAHBAZPANAHI, S., ADVE, R. S. Distributed massive MIMO systems with non-reciprocal channels: Impacts and robust beamforming. *IEEE Transactions on Communications*, 2018, vol. 66, no. 11, p. 5261–5277. DOI: 10.1109/TCOMM.2018.2859937

[29] YUN, S., KANG, J., KIM, I., et al. Deep artificial noise: Deep learning-based precoding optimization for artificial noise scheme. *IEEE Transactions on Vehicular Technology*, 2020, vol. 69, no. 3, p. 3465–3469. DOI: 10.1109/TVT.2020.2965959

[30] ALBREEM, M. A., HABBASH, A. H. A., IKKI, S. S., et al. Overview of precoding techniques for massive MIMO. *IEEE Access*, 2021, vol. 9, p. 60764–60801. DOI: 10.1109/ACCESS.2021.3073325

About the Authors . . .

Lijun GE received the B.E. degree in the major of Electronics Science and Technology in 2006 from Nankai University, Tianjin, China, where subsequently, he did graduate study in the major of Communication and Information Systems, and the Ph.D. degree in the major of Communication and Information Systems after five-year graduate study from Nankai University, in 2011. From 2008 to 2010, he was a Teaching

Assistant with Nankai University, teaching communication-related experiment courses. In 2011, he joined the Department of Communication Engineering, School of Electrical and Electronic Engineering, Tiangong University, Tianjin, China, as a Lecturer in Communication and Information Systems. Since 2013, he has been an Associate Professor in the same academic field, and he has been a Professor and Doctoral Supervisor since 2021. His research interests include OFDM, UWB, massive MIMO wireless communication technologies and wearable Internet of Things, FPGA technologies and applications, and the development of communication and information systems. During the past years, he was in charge of eight projects supported by the nation or the city or some companies, and published more than 40 academic papers.

Shixun NIU received the B.S. degree in Communication Engineering in 2017 from the School of Information Science and Engineering, Henan University of Technology, Zhengzhou, China. He is currently studying toward the M.S. degree in the major of Information and Communication Engineering from the School of Electrical and Electronic Engineering, Tiangong University, Tianjin, China, since 2021. His research interests include massive MIMO systems, deep learning-based wireless communication algorithms, and simulation of communication systems.

Chenpeng SHI received the B.S. degree in Communication Engineering in 2017 from the School of Electrical and Electronic Engineering, Tiangong University, Tianjin, China. He is currently studying toward the M.S. degree in the major of Information and Communication Engineering from the School of Electrical and Electronic Engineering, Tiangong University, Tianjin, China, since 2021. His research interests include massive MIMO systems, deep learning based wireless communication algorithms, and simulation of communication systems.

Yuchuan GUO received the B.S. degree in Communication Engineering in 2015 from the School of Electrical and Electronic Engineering, Tiangong University, Tianjin, China. He is currently studying toward the M.S. degree in the major of Information and Communication Engineering from the School of Electrical and Electronic Engineering, Tiangong University, Tianjin, China, since 2019. His research interests include massive MIMO systems, deep learning based wireless communication algorithms, and simulation of communication systems.

Gaojie CHEN received the B.E. degrees in Electrical Information Engineering and International Economics and Trades from Northwest University, China, in 2006, and the M.Sc. (Hons.) and Ph.D. degrees in Electrical and Electronic Engineering from Loughborough University, Loughborough, U.K., in 2008 and 2012, respectively. After graduation, he took up academic and research positions at DT Mobile, Loughborough University, University of Surrey, University of Oxford and University of Leicester, U.K. He is currently an Assistant Professor with the Institute for Communication Systems, 5GIC & 6GIC, University of Surrey, U.K., and a Visiting Research Collaborator with the Information and Network Science Lab, University of Oxford, U.K. His current research interests include information theory, wireless communications, satellite communications, cognitive radio, the Internet of Things, secrecy communications, and random geometric networks. He served as an Associate Editor for the IEEE Journal on Selected Areas in Communications - Machine Learning in Communications from 2021–2022. He serves as an Associate Editor for the IEEE Communications Letters, IEEE Wireless Communications Letters, and Electronics Letters (IET).

Appendix A: Algorithm 1

```
Algorithm 1: Precoding Algorithm.
  Input: The data streams and channel estimation values for CE-DNN in
  Stage 1 and P-DNN in Stage 2.
  Output: The received data of UEs for CE-DNN in Stage 1. The data
  streams for CE-DNN in Stage 2. The precoding data for P-DNN in
  Stage 2.
  Initialization: Initialize the weight $w$ and the bias $b$ of the hidden and
  output layers randomly.
Training:
Stage 1
  1: Input the training set into the CE-DNN.
  2: Determine the outputs of the CE-DNN in accordance with (15) and
     (18).
  3: Calculate and update the weights and biases according to the SGD
     algorithm.
  4: Repeat steps 1–3, until the stopping standard is met (the number of
     iterations is directly limited or the error difference between adjacent
     two times is very minimal).
Stage 2
  5: Input the training set into the cascaded DNN include CE-DNN and
     P-DNN.
  6: Calculate the outputs of the P-DNN according to (15) and (20). The
     outputs of P-DNN and channel estimation values are used as the input
     of the CE-DNN.
  7: Keep the weight and bias of the CE-DNN unchanged. Calculate and
     update the weights and biases of the cascaded DNN according to the
     SGD algorithm.
  8: Separate the cascaded DNN and make P-DNN an independent net-
     work.
Testing:
Stage 1
  9: Enter the testing set into the CE-DNN, then compute the outputs of the
     CE-DNN and evalute the performance of DL based channel emulator
     according to (19).
Stage 2
 10: Input the testing set into the P-DNN, calculate the outputs of the P-
     DNN and evaluate the performance of DL-based precoder according
     to (23) and (24).
```