Hybrid Precoding Design Based on Dual-Layer Deep-Unfolding Neural Network

Guangyi Zhang, Xiao Fu, Qiyu Hu, Yunlong Cai, and Guanding Yu
College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China
Email: {zhangguangyi, lemonaddie0909, qiyhu, ylcai, yuguanding}@zju.edu.cn

***Abstract*—Dual-layer iterative algorithms are generally required when solving resource allocation problems in wireless communication systems. Specifically, the spectrum efficiency maximization problem for hybrid precoding architecture is hard to solve by the single-layer iterative algorithm. The dual-layer penalty dual decomposition (PDD) algorithm has been proposed to address the problem. Although the PDD algorithm achieves significant performance, it requires high computational complexity, which hinders its practical applications in real-time systems. To address this issue, we first propose a novel framework for deep-unfolding, where a dual-layer deep-unfolding neural network (DLDUNN) is formulated. We then apply the proposed framework to solve the spectrum efficiency maximization problem for hybrid precoding architecture. An efficient DLDUNN is designed based on unfolding the iterative PDD algorithm into a layer-wise structure. We also introduce some trainable parameters in place of the high-complexity operations. Simulation results show that the DLDUNN presents the performance of the PDD algorithm with remarkably reduced complexity.**

***Index Terms*—Hybrid precoding, dual-layer deep-unfolding, penalty dual decomposition algorithm, massive MIMO, machine learning**

I. Introduction

Multi-user multiple-input multiple-output (MU-MIMO) systems with large-scale antenna arrays have received great interest in wireless communications since they can significantly improve the spectrum efficiency [1]. However, the conventional fully-digital (FD) precoding framework for MU-MIMO systems requires that each antenna element is equipped with a dedicated radio frequency (RF) chain, which is prohibitive from the high power consumption and the high hardware cost. Thus, the hybrid precoding scheme has been introduced to address the issue. To obtain the optimal hybrid precoding design, several algorithms have been proposed [2]–[4]. In [2], the orthogonal matching pursuit (OMP) algorithm achieves significant performance when the number of RF chains is larger than that of data streams. The authors in [3] propose hybrid precoding algorithms for multi-user MIMO systems by designing the analog and digital precoders sequentially. Although the above hybrid precoding methods achieve high spectrum efficiency, they are heuristic algorithms, and there is a significant gap between their achievable spectrum efficiency and the theoretical boundary in some specific cases. Distinguishing from the heuristic algorithms, the iterative penalty dual decomposition (PDD) method has been proposed in [4] to solve the problem from the perspective of mathematical optimization. Although the PDD algorithm guarantees converging a Karush–Kuhn–Tucker (KKT) solution of the studied optimization problem, a large number of iterations and large-dimensional matrix inversions are required, which gets in the way of its applications in real-time systems.

In recent years, deep learning (DL) has played an important role in various fields, and many algorithms based on machine learning have been introduced to address the real-time signal processing issues for communication. The main idea of these methods is to treat the iterative algorithm as a black-box, and then utilize the machine learning tool to find the relationship between input and output. In [5] and [6], the authors aim to approximate the iterative weighted minimum mean-square error (WMMSE) algorithm by applying the convolutional neural network (CNN) and the multi-layer perceptron (MLP), respectively. In [7], a novel joint hybrid processing framework (JHPF) based on deep neural network (DNN) has been proposed to optimize the analog and digital processing matrices at the transceiver. However, the generalization ability and interpretability of the black-box based neural network (NN) are fairly poor, and a large number of training samples are required as well. To address this issue, the deep unfolding method has been proposed [8]–[10], where the iterations of iterative algorithms are unfolded into a layer-wise structure by using the NN scheme. The idea of deep-unfolding has been widely applied in wireless communications. A novel general form of iterative algorithm induced deep-unfolding neural network (IAIDNN) has been developed to maximize the sum-rate of MU-MIMO, where the iterative WMMSE algorithm has been unfolded into a layer-wise structure [11].

As mentioned previously, heuristic algorithms generally suffer from performance loss in many specific cases, and the iteration-based PDD algorithm suffers from high computational complexity. In this paper, we first propose a novel framework, where a general dual-layer deep-unfolding neural network (DLDUNN) is formulated to solve the spectrum efficiency maximization problem for hybrid precoding architecture. The DLDUNN is designed in the forward propagation by unfolding the dual-layer PDD algorithm into a layer-wise structure. Firstly, we utilize DNN to replace the bisection method, which requires a large number of large-dimensional matrix inversions. To further improve the performance of the DLDUNN, we introduce trainable parameters when optimizing some key variables. Moreover, to satisfy the power constraint, we adopt the scale operation at the end of the final layer. We conduct extensive tests to verify the effectiveness of the proposed method and the results show that DLDUNN can approximately reach the high performance of the PDD algorithm with greatly reduced complexity. In addition, our model has a

Guangyi Zhang and Xiao Fu contributed equally to this work.

978-1-7281-7586-7/21/$31.00 ©2021 IEEE

strong generalization ability applicationsto support the of more practical scenarios.

The remainder of this paper is organized as follows. A general form of dual-layer deep-unfolding based framework is proposed in Section II. The problem formulation and the PDD algorithms are introduced in Section III. Section IV develops the DLDUNN based on the iterative PDD algorithm. The simulation results and performance analysis are presented in Section V. Finally, we conclude this paper in Section VI.

II. Deep-unfolding Based Framework

In this section, we propose a framework for deep-unfolding, where a general dual-layer deep-unfolding structure is developed.

A. Problem Setup

Consider the following general form of the optimization problem

$$\min_{\mathbf{X}} f(\mathbf{X}) \quad s.t. \quad \mathbf{X} \in \mathcal{X}, \tag{1}$$

where $f : \mathbb{C}^{m\times n} \to \mathbb{R}$ is a continuous differentiable objective function, $\mathbf{X} \in \mathbb{C}^{m\times n}$ is the optimization variable set and $\mathcal{X}$ denotes the feasible set.

Algorithm 1 : General dual-layer iterative algorithm

Input: Initialize $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2\}$
Output: Optimized variables $\mathbf{X}^* = \{\mathbf{X}_1^*, \mathbf{X}_2^*\}$
repeat
  repeat
    $\mathbf{X}_1 = F_i(\mathbf{X}_1; \mathbf{X}_2)$
  until inner loop stopping criterion are met
  $\mathbf{X}_2 = F_t(\mathbf{X}_1; \mathbf{X}_2)$
until outer loop stopping criterion are met
return $\mathbf{X_1}, \mathbf{X_2}$

Algorithm 2 : General dual-layer deep-unfolding framework

Input: Initialize $\mathbf{X} = \{\mathbf{X}_1^{0,0}, \mathbf{X}_2^0\}$
Output: Optimized $\mathbf{X}^* = \{\mathbf{X}_1^{T-1,I}, \mathbf{X}_2^T\}$
for t from 0 to $T-1$
  for i from 0 to $I-1$
    $\mathbf{X}_1^{t,i+1} = \mathcal{F}_i(\mathbf{X}_1^{t,i}, \mathbf{X}_2^t; \mathbf{\Theta}^{t,i})$
  end for
  $\mathbf{X}_2^{t+1} = \mathcal{F}_t(\mathbf{X}_1^{t,I}, \mathbf{X}_2^t; \mathbf{\Theta}^t)$
end for
return $\mathbf{X}_1^{T-1,I}, \mathbf{X}_2^T$

The dual-layer iterative algorithm can be designed to solve the problem in (1). Specifically, the variable set $\mathbf{X}$ can be decomposed into $\mathbf{X}_1$ and $\mathbf{X}_2$, where $\mathbf{X}_1$ denotes the variable set iterated in the inner loop and $\mathbf{X}_2$ is the variable set updated in the outer loop. Thus, the iterative algorithm can be formulated as a general dual-layer expression in **Algorithm 1**, where the functions F_i and F_t map the original variables to the updated ones of the inner loop and outer loop, respectively.

B. Forward Propagation

Based on the structure in **Algorithm 1**, we introduce some trainable parameters to reduce the computational complexity of the iterative algorithm and improve its performance. Moreover, the inner and outer loops of the iterative algorithm are replaced with the inner and outer NN layers, respectively. The framework can be illustrated in **Algorithm 2**, where $i \in \mathcal{I} \triangleq \{1, 2, ..., I\}$ represents the index of the inner layer of NN with I being the total number of the inner layers between two adjacent outer layers, $t \in \mathcal{T} \triangleq \{1, 2, ..., T\}$ represents the index of the outer layer of NN with T being the total number of outer layers, $\mathcal{F}_i$ and $\mathcal{F}_t$ respectively denote the structure of the network in the i-th inner layer and the t-th outer layer, $\mathbf{\Theta}$ denotes the trainable parameter set.

III. Problem Formulation and PDD Method

In this section, we introduce the system model and the PDD approach for the studied system.

A. System Model

We consider a downlink multi-user MIMO system, in which the base station (BS) is equipped with N transmit antennas and $N_{RF}(\leq N)$ transmit radio frequency (RF) chains. The BS transmits data to K users, each is equipped with M receive antennas and $M_{RF}(\leq M)$ receive RF chains. The number of data streams sent to each user is d, which satisfies $d \leq \min(N_{RF}/K, M_{RF})$.

For hybrid precoding, the data signal stream is processed sequentially by digital precoder, RF chains, and analog precoder consisting of analog phase shifters. Let $\mathbf{V}_{BB_k} \in \mathbb{C}^{N\times d}$ denote the digital precoder for user k, and let $\mathbf{s}_k$ denote the data stream sent to user k. We assume that $\mathbf{s}_k$ has zero mean and unit variance. Let $\mathbf{V}_{RF} \in \mathbb{C}^{N\times N_{RF}}$ denote the analog precoder of the BS. Hence the signal transmitted by the BS after hybrid precoding is obtained as

$$\boldsymbol{y}_k = \mathbf{H}_k \boldsymbol{x} + \boldsymbol{n}_k, \tag{2}$$

where $\mathbf{H}_k$ denotes the channel from the BS to user k, and $\boldsymbol{n}_k$ represents the additive Gaussian white noise, i.e., $\boldsymbol{n}_k \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$. Thus we obtain the signal received by user k as

$$\boldsymbol{x} = \mathbf{V}_{RF} \sum_{k=1}^{K} \mathbf{V}_{BB_k} \boldsymbol{s}_k. \tag{3}$$

Let $\mathbf{U}_{RF_k} \in \mathbb{C}^{M\times M_{RF}}$ and $\mathbf{U}_{BB_k} \in \mathbb{C}^{M_{RF}\times d}$ denote analog combiner and digital combiner at the receiver side, respectively. After the received signal is processed by analog combiner, receive RF chain, and digital combiner sequentially, we obtain the final processed signal as

$$\hat{\mathbf{s}}_k = \mathbf{U}_{BB_k}^H \mathbf{U}_{RF_k}^H \boldsymbol{y}_k. \tag{4}$$

It is assumed that the signals and noises among different users are independent from each other, and we aim at maximzing the overall system spectrum efficiency subject to transmit power constraint and unit modulus constraint. Thus, the problem can be formulated as

$$\max_{\mathbf{V},\mathbf{U}} \sum_{k=1}^{K} \log\det\big(\mathbf{I} + \mathbf{U}_{RF_k}^H \mathbf{H}_k \mathbf{V}_{RF} \mathbf{V}_{BB_k} \times \mathbf{V}_{BB_k}^H \mathbf{V}_{RF}^H \mathbf{H}_k^H \mathbf{U}_{RF_k} \tilde{\mathbf{\Upsilon}}_k^{-1}\big) \tag{5}$$

s.t.

$$\sum_{k=1}^{K} \|\mathbf{V}_{RF}\mathbf{V}_{BB_k}\|^2 \leq P, \tag{5a}$$

$$|\mathbf{V}_{RF}(i,j),| = 1, \forall i,j, \tag{5b}$$

$$|\mathbf{U}_{RF_k}(i,j)| = 1, \forall i,j,k, \tag{5c}$$

where the interference-plus-noise covariance matrix is

$$\tilde{\mathbf{\Upsilon}}_{\mathbf{k}} \triangleq \mathbf{U}_{RF_k}^H \left(\sigma^2\mathbf{I} + \sum_{j\neq k} \mathbf{H}_k\mathbf{V}_{RF}\mathbf{V}_{BB_j} \times \mathbf{V}_{BB_j}^H\mathbf{V}_{RF}^H\mathbf{H}_k^H\right)\mathbf{U}_{RF_k}. \tag{6}$$

In the above, (5a) is the power constraint, and the last two unit modulus constraints (5b) and (5c) must be satisfied due to the fact that both the analog combiner and the analog precoder consist of low-cost phase shifters.

B. PDD Method

The spectrum efficiency maximization problem for hybrid precoding is difficult to solve since it has non-convex coupling constraints. In [4], an efficient PDD algorithm has been proposed to address this problem, and it has been proved that problem (5) can be equivalently converted to the following augmented Lagrangian problem:

$$\max_{\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{X}} \sum_{k=1}^{K}(\log\det(\mathbf{W}_k) - \mathrm{Tr}(\mathbf{W}_k\mathbb{E}_k(\mathbf{U},\mathbf{X})) + d) - \sum_{k=1}^{K}\frac{1}{2\rho}\|\mathbf{X}_k - \mathbf{V}_{RF}\mathbf{V}_{BB_k} + \rho\mathbf{Y}_k\|^2 \tag{7}$$

s.t.

$$\sum_{k=1}^{K} \|\mathbf{X}_k\|^2 \leq P, \tag{7a}$$

$$|\mathbf{V}_{RF}(i,j),| = 1, \forall i,j, \tag{7b}$$

$$|\mathbf{U}_{RF_k}(i,j)| = 1, \forall i,j,k, \tag{7c}$$

where $\mathbf{Y}_k$ is the dual variable introduced for the coupling constraints $\mathbf{X}_k = \mathbf{V}_{RF}\mathbf{V}_{BB_k}$, $\mathbf{W}_k$ is the introduced intermediate variable. The mean-square-error (MSE) matrix is

$$\mathbb{E}_k(\mathbf{U},\mathbf{X}) \triangleq \mathbf{U}_{BB_k}^H\mathbf{\Upsilon}_{\mathbf{k}}\mathbf{U}_{BB_k} + \left(\mathbf{I} - \mathbf{U}_{BB_k}^H\mathbf{U}_{RF_k}^H\mathbf{H}_k\mathbf{X}_k\right)\left(\mathbf{I} - \mathbf{U}_{BB_k}^H\mathbf{U}_{RF_k}^H\mathbf{H}_k\mathbf{X}_k\right)^H. \tag{8}$$

PDD algorithm converges to the approximately global optimal solution via the double loop iterations. In the inner loop, the algorithm updates each variable block in each iteration. To optimize $\mathbf{V}_{RF}$ and $\mathbf{U}_{RF}$, the algorithm adopts a BCD-type method [4] to find the analog precoder and analog combiner matrix that satisfy (5b) and (5c). To optimize $\mathbf{X}$, the algorithm takes the bisection method to find the parameter μ that makes $\mathbf{X}_k$ satisfy (7a). In the outer loop, the coupling constraints $\mathbf{X}_k$ = $\mathbf{V}_{RF}\mathbf{V}_{BB_k}$ is satisfied by updating the penalty parameter $\mathbf{Y}_k$, i.e., the power constraint in the problem (5a) is satisfied simultaneously due to the satifactory of (7a).

More specifically, the details of PDD algorithm are shown in **Algorithm 3**, and the supplements for **Algorithm 3** are shown in Table I.

Algorithm 3 Penalty dual decomposition method

Input: Initialize $\mathbf{V}_{RF}$, $\mathbf{V}_{BB_k}$, $\mathbf{U}_{RF_k}$ and $\mathbf{X}_k$ to satisfy $\mathbf{X}_k = \mathbf{V}_{RF}\mathbf{V}_{BB_k}$, $|\mathbf{V}_{RF}(i,j)| = 1$, $|\mathbf{U}_{RF_k}(i,j)| = 1$, $\sum_{k=1}^{K}\|\mathbf{V}_{RF}\mathbf{V}_{BB_k}\|^2 \leq P$. Set the constraint violation parameter$\{\epsilon,\eta_0\}$, the penalty parameter ρ, the control parameter c.

Output: optimal $\{\mathbf{V}_{RF},\mathbf{V}_{BB_k},\mathbf{U}_{RF_k},\mathbf{U}_{BB_k}\}$

repeat
  repeat
    $\mathbf{U}_{BB_k} = \left(\mathbf{U}_{RF_k}^H\mathbf{A}_k\mathbf{U}_{RF_k}\right)^{\dagger}\left(\mathbf{U}_{RF_k}^H\mathbf{H}_k\mathbf{X}_k\right)$
    $\mathbf{W}_k$=$\left(\mathbf{I} - \mathbf{U}_{BB_k}^H\left(\mathbf{U}_{RF_k}^H\mathbf{H}_k\mathbf{X}_k\right)\right)^{-1}$
    $\mathbf{V}_{RF} = \mathbf{BCD}\left(\mathbf{I},\mathbf{V}_{RF},\sum_{k=1}^{K}\mathbf{V}_{BB_k}\mathbf{V}_{BB_k}^H, \sum_{k=1}^{K}\left(\mathbf{X}_k + \rho\mathbf{Y}_k\right)\mathbf{V}_{BB_k}^H\right)$
    $\mathbf{U}_{RF_k} = \mathbf{BCD}\left(\mathbf{A}_k,\mathbf{U}_{RF_k},\mathbf{U}_{BB_k}\mathbf{W}_k\mathbf{U}_{BB_k}^H, \mathbf{H}_k\mathbf{X}_k\mathbf{W}_k\mathbf{U}_{BB_k}^H\right)$
    $\mathbf{V}_{BB_k} = \left(\mathbf{V}_{RF}\right)^{\dagger}\left(\mathbf{X}_k + \rho\mathbf{Y}_k\right)$
    $\mathbf{X}_k = \left(\mathbf{A}_\rho + \mu\mathbf{I}\right)^{-1}\mathbf{B}_{\rho,k}$
  until $\frac{\left|\mathcal{L}_k\left(\mathbf{x}^t\right) - \mathcal{L}_k\left(\mathbf{x}^{t-1}\right)\right|}{\left|\mathcal{L}_k\left(\mathbf{x}^{t-1}\right)\right|} \leq \epsilon$
  if $\max_k\|\mathbf{X}_k - \mathbf{V}_{RF}\mathbf{V}_{BB_k}\|_\infty \leq \eta_t$ then
    $\mathbf{Y}_k = \mathbf{Y}_k + \frac{1}{\rho}\left(\mathbf{X}_k - \mathbf{V}_{RF}\mathbf{V}_{BB_k}\right)$
  else
    $\rho = c\rho$
  end if $t = t+1$
  η_t=0.9 $\max_k\|\mathbf{X}_k - \mathbf{V}_{RF}\mathbf{V}_{BB_k}\|_\infty$, $\epsilon_k = c\epsilon_k$
until $\max_k\|\mathbf{X}_k - \mathbf{V}_{RF}\mathbf{V}_{BB_k}\|_\infty \leq \epsilon$

TABLE I
SUPPLEMENTS FOR **ALGORITHM 1**

$\mathbf{A}_k \triangleq \sigma^2\mathbf{I} + \sum_{j=1}^{K}\mathbf{H}_k\mathbf{X}_j\mathbf{X}_j^H\mathbf{H}_k^H$
BCD-type algorithm can be found in [4]
$\mathbf{A}_\rho \triangleq \sum_{j=1}^{K}(\mathbf{H}_j^H\mathbf{U}_{RF_j}\mathbf{U}_{BB_j}\mathbf{W}_j\mathbf{U}_{BB_j}^H\mathbf{U}_{RF_j}^H\mathbf{H}_j) + \frac{1}{2\rho}\mathbf{I}$
$\mathbf{B}_{\rho,k} \triangleq \mathbf{H}_k^H\mathbf{U}_{RF_k}\mathbf{U}_{BB_k}\mathbf{W}_k + \frac{1}{2}(\frac{1}{\rho}\mathbf{V}_{RF_k}\mathbf{V}_{BB_k} - \mathbf{Y}_k)$
μ satisfies $\sum_{k=1}^{K}\mathrm{Tr}(\mathbf{B}_{\rho,k}^H\left(\mathbf{A}_\rho + \mu\mathbf{I}\right)^{-2}\mathbf{B}_{\rho,k}) = P$ and is obtained by applying bisection method.
$\mathcal{L}_k = \sum_{k=1}^{K}\left(\log\det\left(\mathbf{W}_\mathbf{k}\right) - \mathrm{Tr}\left(\mathbf{W}_k\mathbf{E}_k(\mathbf{U},\mathbf{X})\right) + d\right) - \sum_{k=1}^{K}\frac{1}{2\rho}\|\mathbf{X}_k - \mathbf{V}_{RF}\mathbf{V}_{BB_k} + \rho\mathbf{Y}_k\|^2$

IV. THE PROPOSED DLDUNN FOR HYBRID PRECODING

In this section, we introduce the architecture of DLDUNN based on the iterative PDD algorithm and analyze the computational complexity.

A. The Architecture of DLDUNN

The spectrum efficiency maximization problem for hybrid precoding design can be solved well by the PDD method. However, the PDD method has considerable computational complexity since it requires a large number of iterations to converge. Moreover, many complicated operations, such as matrix inversion and iterative BCD-type algorithm, are involved in the PDD algorithm.

To address the above issues, we design the architecture of DLDUNN as shown in Fig. 1. Let D_{in} and D_{out} represent

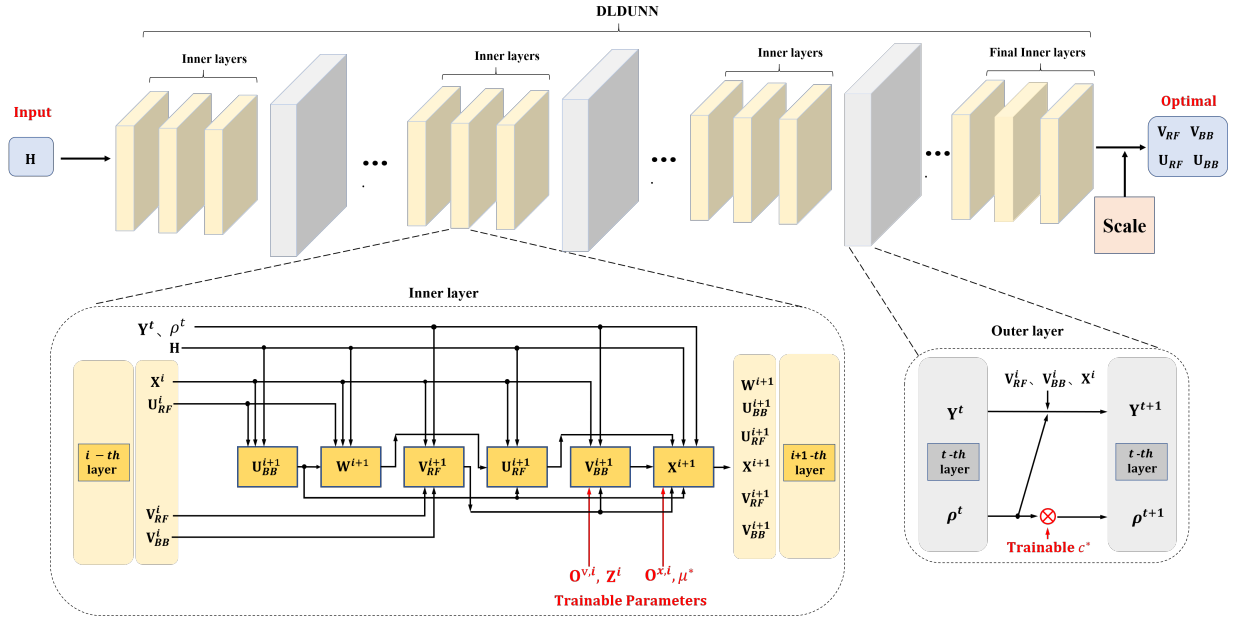
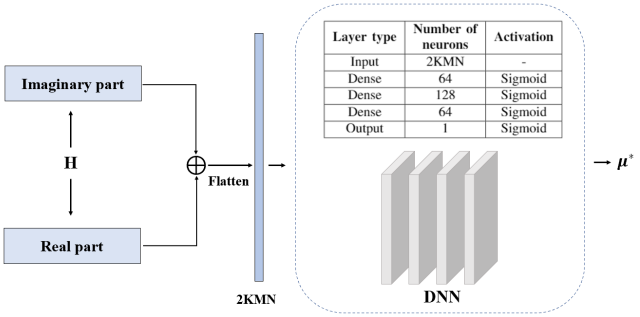


Fig. 1. The architecture of DLDUNN. The index i, $i+1$ denote the current inner layer and the next inner layer, repectively. The index t, $t+1$ denote the current outer layer and the next outer layer, repectively. The scale operation is placed after the final layer.

the numbers of DLDUNN's inner layers and outer layers, respectively. Let P_{in} and P_{out} denote the numbers of PDD's inner iterations and outer iterations, respectively. Moreover, let L_{bcd} denote the number of iterations of BCD-type algorithm. Thus, the main structures of our model can be described as follows.

1) The number of iterations for DLDUNN's inner and outer layers is fixed.
2) The conditional branch statements, i.e., if-else, may cause gradient interruption in the backpropagation. Thus, we keep the dual variable $\mathbf{Y}_k$ and the penalty parameter ρ updated in each outer layer, which are originally selectively updated in step 10-13 of **Algorithm 3**. Besides, the original iterative BCD-type algorithm applied in step 5-6 of **Algorithm 3** also terminates under specific conditions. Hence we design a simple-BCD layer by fixing the iteration number of the iterative BCD-type algorithm to 1.
3) To find the appropriate penalty parameter ρ for each inner layer, we introduce trainable parameter c^* to replace the original scalar c in the outer layer as
$$\rho^{t+1} = c^* \rho^t. \tag{9}$$
4) Trainable parameters $\mathbf{Z}$ and $\mathbf{O}^v$ are introduced to address the performance loss caused by fixing $L_{bcd} = 1$. Notice that $\mathbf{V}_{RF}$ is directly multiplied with $\mathbf{V}_{BB_k}$ in the objective function (5). Thus, the structure of $\mathbf{V}_{BB}$'s layer can be designed as
$$\mathbf{V}_{BB_k}^{i+1} = \mathbf{Z}_k^i \left(\mathbf{V}_{RF}^i\right)^\dagger \left(\mathbf{X}_k^i + \rho^t \mathbf{Y}_k^t\right) + \mathbf{O}_k^{v,i}. \tag{10}$$
5) We introduce the DNN to replace the bisection method in $\mathbf{X}$'s layer. The NN takes the channel matrix $\mathbf{H}$ as the input to learn μ^*. Moreover, an extra trainable parameter $\mathbf{O}^x$ is introduced to compensate for the transformation. Thus, the structure of $\mathbf{X}$'s layer can be designed as
$$\mathbf{X}_k^{i+1} = \left(\mathbf{A}_\rho^i + \mu^* \mathbf{I}\right)^{-1} \mathbf{B}_{\rho,k}^i + \mathbf{O}_k^{x,i}. \tag{11}$$
6) The power constraint (5a) cannot be satisfied by the calculated μ^*. Thus the scale operation is introduced at the end of the architecture, as
$$\mathbf{V}_{BB_k} = \mathbf{V}_{BB_k}^* \frac{P}{\left\|\mathbf{V}_{RF}\mathbf{V}_{BB_k}^*\right\|^2}, \forall k, \tag{12}$$
where $\mathbf{V}_{BB_k}^*$ denotes the digital precoder before being scaled.



Layer type	Number of neurons	Activation
Input	2KMN	-
Dense	64	Sigmoid
Dense	128	Sigmoid
Dense	64	Sigmoid
Output	1	Sigmoid

Fig. 2. The architecture of DNN for μ^*

Finally, the objective function (5) can be regarded as the loss function of the DLDUNN.

B. Computational Complexity Analysis

We define the total number of iterations as the product of the number of inner layers and the number of outer layers.

Therefore, DLDUNN has much lower computational complexity compared to PDD algorithm mainly in three aspects:

1) The total number of iterations in DLDUNN is much smaller, i.e. $D_{in} \times D_{out}$ (70 in this simulation) $\ll P_{in} \times P_{out}$ (generally larger than 10000).
2) The main computations of the PDD algorithm and our proposed DLDUNN lie in each inner iteration and each inner layer, respectively. For PDD, the BCD-type algorithm requires to iterate for many times when optimizing $\mathbf{V}_{RF}$ and $\mathbf{U}_{RF}$ with complexity $\mathcal{O}\left(N^2 N_{RF}^2\right)$ and $\mathcal{O}\left(KM^2 M_{RF}^2\right)$, respectively, while in DLDUNN, we iterate it only once. In step 8 of **Algorithm 3**, the bisection method requires nearly 100 times matrix inversion operations ($\mathcal{O}\left(N^3\right)$), and the value of N is large, while this step is converted into matrix multiplications ($\mathcal{O}\left(n^{2.37}\right)$, n depends on the structure of the DNN) in DLDUNN.
3) Since DLDUNN has a fixed number of iterations, there is no need to calculate the convergence variable $\mathcal{L}_k\,(\mathrm{x})$ (in Table I).

V. Simulation Result

In our simulation, we adopt the complex Gaussian MIMO channel, i.e., the element $h_{i,j}$ in $\mathbf{H}_k$ satisfies $h_{i,j} \sim \mathcal{CN}(0,1)$. We assume that all users are equipped with $M = 4$ receive antennas and the number of signal streams received by each user is $d = 2$. We set the number of receive RF chains $M_{RF} = d$. We assume that the BS is equipped with $N = 24$ transmit antennas and $N_{RF} = 2Kd$ transmit RF chains to satisfy $N_{RF} \geq Kd$. Moreover we set $D_{in} = 10$, $D_{out} = 7$ as the number of layers in our proposed DLDUNN. In each simulation, we run the channel matrix 100 times and take the average loss function as our estimate of the spectrum efficiency. We take PDD's average results of 100 randomly initialized channel as the approximate global optimal solution. In addition, to compare the performance between the PDD algorithm and our proposed DLDUNN under the same number of iterations, we design the PDD with the same layers (PSL) by limiting the maximum number of inner layer's iteration to $P_{in} = 10$ and that of outer layer's iteration to $P_{out} = 7$.

A. Spectrum Efficiency

In the following experiments, we compare the spectrum efficiency achieved by our proposed DLDUNN with that achieved by the PDD algorithm [4], the PSL method, the black-box network [7], and the heuristic RBD method [3]. Moreover, the FD precoding performance is regarded as the performance upper bound.

Fig. 3 depicts the spectrum efficiency achieved under various methods with different values of SNR ($10\log_{10}(P/\sigma^2)$). The performance achieved by the PDD algorithm and our proposed DLDUNN is close to the FD precoding performance. The gap between PDD and DLDUNN does not increase with SNR. In contrast, the RBD method suffers from severe performance loss as compared to PDD and DLDUNN. Moreover, the gap between PSL's performance and that of DLDUNN increases with SNR, which shows that the PDD algorithm requires more iterations to maintain a good performance at large SNR.

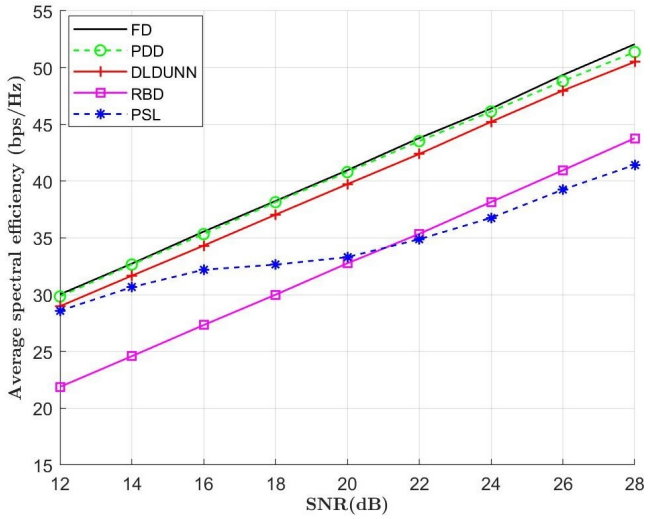


Fig. 3. Spectrum efficiency achieved by different methods, $N_{RF} = 12$, $K = 3$, $d = 2$.

TABLE II
Spectrum efficiency versus multi-users

# of user (K)	2	3	4	5
FD(bps/Hz)	28.985	40.964	51.880	62.038
PDD(bps/Hz)	28.932	40.801	51.643	61.513
Achievable performance (PDD-based)				
DLDUNN	97.82%	97.49%	97.57%	97.35%
PSL	77.91%	81.64%	86.77%	91.63%
RBD	75.88%	79.90%	83.86%	87.56%

Table II shows the spectrum efficiency (bps/Hz) versus different numbers of users. Our proposed DLDUNN achieves more than 97% spectrum performance of the PDD algorithm on average, which outperforms both the RBD and the PSL methods.

Fig. 4 depicts the spectrum efficiency (bps/Hz) versus different numbers of RF chains. It has been proved that the performance of the FD structure can be perfectly achieved by the hybrid precoding structure when the number of transmit RF chains is equal to or larger than twice the total number of data streams, i.e., $N_{RF} \geq 2Kd$ [12]. Meanwhile, the number of transmit RF chains must be larger than or equal to the total number of data streams for successful symbol detection, i.e., $N_{RF} \geq Kd$. Thus the N_{RF} starts from $Kd = 4$ in this test. The FD's result (28.985bps/Hz) is regarded as an upper bound since N_{RF} is equal to N in the FD precoding structure. Results show that the proposed DLDUNN can approach the performance of PDD at various scenarios with different numbers of RF chains. Specifically, the proposed DLDUNN is able to perform well even in the more general scenarios ($N_{RF} < 2Kd$), which is of great significance compared with other existing precoding methods. We can also see that the spectrum performance of the RBD method gradually approaches the performance of PDD and DLDUNN as the number of the transmit RF chains increases. It is mainly due to the fact that with the increase of N_{RF}, the system is gradually closer to the FD precoding structure,

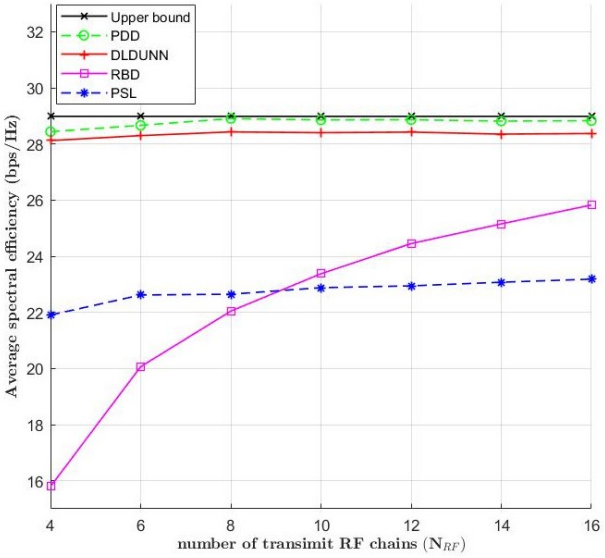


Fig. 4. Spectrum efficiency versus RF chains, $K = 2$, $SNR = 20dB$.

which can be well solved by the channel matching strategies.

We introduce the DNN-based black-box network as the benchmark in the simulation. The results are presented in Table III, and the percentages inside are calculated via dividing the values of spectrum efficiency achieved by the black-box or DLDUNN by those of the PDD. We can see that the black-box has relatively good performance for single user ($K = 1$) systems. However, our additional tests show that the black-box network suffers from serious performance loss in multi-user cases. In contrast, our proposed DLDUNN still maintains a good performance in multi-user scenarios by taking full advantage of the PDD algorithm's structure.

TABLE III
PERFORMANCE COMPARISON OF DLDUNN AND BLACK-BOX

SNR/dB	12	14	16	18
DLDUNN	97.34%	97.77%	97.94%	97.78%
Black-box	84.50%	84.83%	87.23%	87.55%
SNR/dB	20	22	24	26
DLDUNN	97.84%	97.38%	97.62%	96.90%
Black-box	89.08%	89.02%	90.45%	90.64%

B. Generalization Analysis

NN-based models are generally required to have a good generalization property to support applications in various complex scenarios. We evaluate the generalization ability of DLUDNN in two aspects.

1) After being trained with a larger user number (K), the model can be directly applied to the system of smaller users without being retrained.
2) In the training process of the model, fixed SNR is used as inputs, then we apply the trained model with $SNR = 20dB$ to various SNR values for experiments. The results are presented in Table IV, which shows that the model trained with fixed SNR can work efficiently with different values of SNR.

TABLE IV
GENERALIZATION PERFORMANCE OF DLDUNN

SNR/dB	12	14	16	18
PDD (bps/Hz)	21.766	23.561	25.339	27.171
DLDUNN	97.41%	97.61%	97.43%	97.40%
SNR/dB	20	22	24	26
PDD (bps/Hz)	28.906	30.813	32.539	34.263
DLDUNN	97.82%	97.96%	97.77%	96.77%

VI. CONCLUSION

This paper proposed a deep-unfolding framework for the dual-layer iterative algorithm. To design the hybrid precoding scheme under multi-user MIMO, we designed a DLDUNN based on the PDD algorithm. The PDD algorithm has been unfolded into a layer-wise structure, where we introduced learning parameters to replace the large number of complex computational operations. The experimental results showed that our proposed DLDUNN can achieve high performance of the PDD algorithm after being trained and with a significantly reduced computational complexity. In the future, our method can also be applied to more practical scenarios.

REFERENCES

[1] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436-1449, Apr. 2013.
[2] O. E. Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi, and R. W. Heath, "Spatially sparse precoding in millimeter wave MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1499-1513, Mar. 2014.
[3] J. Zhang, M. Haardt, I. Soloveychik, and A. Wiesel, "A channel matching based hybrid analog-digital strategy for massive multi-user MIMO downlink systems," 2016 *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Jul 2016, pp. 1–5.
[4] Q. Shi and M. Hong, "Spectral efficiency optimization for millimeter wave multiuser MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 3, pp. 455-468, Jun. 2018.
[5] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438-5453, Oct. 2018.
[6] W. Lee, M. Kim, and D. Cho, "Deep power control: Transmit power control scheme based on convolutional neural network," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1276-1279, Jun. 2018.
[7] P. Dong, H. Zhang, and G. Y. Li, "Framework on deep learning-based joint hybrid processing for mmWave massive MIMO systems," *IEEE Access*, vol. 8, pp. 106023-106035, Jun. 2020.
[8] J. Chien and C. Lee, "Deep unfolding for topic models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 318-331, Feb. 2018.
[9] R. Liu, S. Cheng, L. Ma, X. Fan, and Z. Luo, "Deep proximal unrolling: Algorithmic framework, convergence analysis and applications," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5013-5026, Oct. 2019.
[10] L. Zhang, G. Wang, and G. B. Giannakis, "Real-time power system state estimation and forecasting via deep unrolled neural networks," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 4069-4077, Aug. 2019.
[11] Q. Hu, Y. Cai, Q. Shi, K. Xu, G. Yu, and Z. Ding, "Iterative algorithm induced deep-unfolding neural networks: Precoding design for multiuser MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1394-1410, Feb. 2021.
[12] 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, Apr. 2016.