Deep-Unfolding Neural-Network Aided Hybrid Beamforming Based on Symbol-Error Probability Minimization

Shuhan Shi, Yunlong Cai, Qiyu Hu, Benoit Champagne, and Lajos Hanzo

***Abstract*—In massive multiple-input multiple-output (MIMO) systems, hybrid analog-digital (AD) beamforming can be used to attain a high directional gain without requiring a dedicated radio frequency (RF) chain for each antenna element, which substantially reduces both the hardware costs and power consumption. While massive MIMO transceiver design typically relies on the conventional mean-square error (MSE) criterion, directly minimizing the symbol error rate (SER) can lead to a superior performance. In this paper, we first mathematically formulate the problem of hybrid transceiver design under the minimum SER (MSER) optimization criterion and then develop an MSER-based iterative gradient descent (GD) algorithm to find the related stationary points. We then propose a deep-unfolding neural network (NN). The iterative GD algorithm is unfolded into a multi-layer structure wherein trainable parameters are introduced to accelerate the convergence and enhance the overall system performance. To implement the training stage, we derive the relationship between adjacent layers' gradients based on the generalized chain rule (GCR). The deep-unfolding NN is developed for both quadrature phase shift keying (QPSK) and M-ary quadrature amplitude modulated (QAM) signals, and its convergence is investigated theoretically. Furthermore, we analyze the transfer capability, computational complexity, and generalization capability of the proposed deep-unfolding NN. Our simulation results show that the latter significantly outperforms its conventional counterpart at a reduced complexity.**

***Index Terms*—Hybrid beamforming, massive MIMO, deep-unfolding, MSER, machine learning.**

I. Introduction

The massive multiple-input multiple-output (MIMO) technology has inspired wide research attention since it is capable of dramatically increasing the system capacity by forming directional beams, hence mitigating the spectrum shortage [1]–[3]. However, these large-scale antenna arrays employed both at the transmitter and receiver have a prohibitive hardware cost and power consumption in fully-digital (FD) beamforming, where each antenna element requires a dedicated radio frequency (RF) chain. Thus, hybrid analog-digital (AD) beamforming has become an important technique of reducing the number of RF chains, while still approaching the performance of fully-digital beamforming [3]–[14].

A. Prior Art

Hybrid AD beamforming conceptually relies on the decomposition of the beamforming operation as the product of a low-dimensional digital beamforming matrix and a high-dimensional analog beamforming matrix. The elements of the analog beamforming matrix obey the unit-modulus constraint imposed by the phase shifters. To reduce the number of RF chains, several authors [4], [5] advocated antenna-selection based hybrid beamforming. In order to further improve their performance, spatially sparse precoding techniques were developed by exploiting the millimeter-wave (mmWave) channel characteristics in [3], [6]. Moreover, the authors of [7] demonstrated that hybrid AD beamforming having twice as many RF chains as data streams approaches the performance of FD beamforming.

Inspired by these findings, sophisticated optimization algorithms were proposed for hybrid beamforming [6], [8]–[14]. In [6], the authors leveraged the channel's sparsity and proposed an orthogonal matching pursuit (OMP) aided algorithm. Based on the OMP concept, the authors of [8]–[10] developed a hybrid transceiver architecture relying on the minimum mean square error (MMSE) criterion. In [11], the authors formulated a matrix factorization problem and developed a manifold optimization (MO) based hybrid beamforming algorithm, where the unit-modulus constraint was considered as a Riemannian manifold. In [12], a codebook-based hybrid beamforming design was conceived for maximizing the system's spectral efficiency. As a further development, the authors of [13] conceived a two-stage optimization algorithm based on the general eigen-decomposition method, and their work evolved further in [14] to the broadband scenario with the aid of MO.

However, most of the existing hybrid beamforming designs in the open literature are based on minimizing the mean square error (MSE), which is not the most appropriate metric from a performance viewpoint in digital communications where the standard measure is the symbol-error-rate (SER). Recently, several powerful beamformers have been designed based on the minimum SER (MSER) criterion [15]–[19]. The authors of [15] and [16] developed an MSER-based adaptive reduced-rank receive beamformer to enhance the performance. In [17], a single-bit direct MSER-optimization based precoder was

The work of Y. Cai was supported in part by the National Natural Science Foundation of China under Grants 61971376 and 61831004, and the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars under Grant LR19F010002. L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/W016605/1 and EP/P003990/1 (COALESCE) as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028). (Corresponding author: Yunlong Cai.)

S. Shi, Y. Cai, and Q. Hu are with the College of Information Science and Electronic Engineering, Zhejiang University, China (e-mail: ssh16@zju.edu.cn; ylcai@zju.edu.cn; qiyhu@zju.edu.cn). B. Champagne is with the Department of Electrical and Computer Engineering, McGill University, Canada (e-mail: benoit.champagne@mcgill.ca). L. Hanzo is with the Department of Electronics and Computer Science, University of Southampton, U.K. (e-mail: lh@ecs.soton.ac.uk).

proposed for simplifying the RF chains. The authors of [18] designed an interference-aided precoder for minimizing the SER of the worst-case user in the system. As a further development, in [19], the authors proposed an MSER-based precoder for K-pair MIMO interference channels by utilizing improper signaling. However, due to the associated complex high-dimensional matrix inversions and decompositions, the existing optimization algorithms suffer from high complexity in practical implementation.

To improve the performance at a reduced complexity, researchers have recently turned their attention to machine learning techniques for solving a variety of problems, such as channel decoding [20], end-to-end communication [21], [22], and channel estimation [23]. Well-trained neural networks (NNs) are capable of learning the mapping between the system inputs and outputs [24]–[29]. One of the early attempts along this avenue appeared in [24], where a NN was trained for performing power allocation based on an iterative weighted MMSE (WMMSE) algorithm. The authors of [25] proposed a two-stage training mechanism for maximizing the sum-rate by applying convolutional neural networks (CNN), and further improving the system performance through unsupervised learning. In [26], the authors adopted an ensemble of fully connected deep neural networks (DNNs) for optimizing the transmit power allocation. In [27], the problem of joint antenna selection and hybrid beamforming was first formulated as a classification problem, and then solved by a deep CNN. Deep learning was also utilized for jointly optimizing the compressive channel and hybrid beamforming matrices in [28]. In [29], the authors designed a novel NN inspired by GoogleNet, which used parallel complex convolutional blocks for hybrid beamforming.

In the above contributions, NNs are generally treated as black-boxes, which does not guarantee optimal performance and leads to limited interpretability, along with limited control. Furthermore, in massive MIMO systems, the training overhead of these methods is expensive due to the multi-dimensional training samples and long training time. A model-driven network, referred to as deep unfolding, has been proposed in [30] to overcome these issues. This method generally unfolds iterative algorithms into multi-layer NN structures and introduces a set of trainable parameters to accelerate convergence and increase the system performance. The authors of [31] and [32] invoked this approach to unfold the gradient descent (GD) algorithm for data detection. In [33], [34], the authors developed a deep-unfolding method inspired by the normalized min-sum algorithm to decode polar codes. In [35], a scheme based on deep-unfolding was put forward for sparse signal recovery. In [36], the authors employed the deep-unfolding method for automatically optimizing the precoders relying on single-bit digital-to-analog converters (DACs). The authors of [37] adopted a deep-unfolding NN based on the conjugate GD algorithm for constant envelope precoding. A finite-alphabet precoder was developed in [38], which unfolded the iterative discrete estimation algorithm into a NN. In [39], the authors derived the generalized chain rule (GCR) in matrix form and proposed a deep-unfolding NN based on the WMMSE precoding design algorithm.

B. Main Contributions

We formulate a joint hybrid AD transceiver design problem based on the MSER criterion in a massive MIMO system context, which is challenging due to the highly nonconvex objective function and unit-modulus constraints imposed on the entries of the analog beamforming matrix. We develop an unconstrained MSER-based iterative GD hybrid beamforming algorithm for solving the investigated problem through the application of kernel density estimation and a parametric representation of the unit-modulus matrix entries. While the proposed iterative GD algorithm can be adopted for finding the stationary solutions, it requires a long time to converge and a large number of iterations. Moreover, by making full use of the structure of the proposed iterative GD algorithm, we propose a novel deep-unfolding NN, in which the iterative GD algorithm is unfolded into a multi-layer structure. To the best of our knowledge, however, the deep-unfolding method has not been investigated in the context of hybrid beamforming based on the MSER criterion. Then, a set of trainable parameters is introduced into the forward propagation (FP) to replace the complex operations, accelerate the convergence and enhance the system performance. In the training stage, the relationship between the gradients of adjacent layers is derived based on the GCR in the back propagation (BP). Then a deep-unfolding NN is developed for both quadrature phase shift keying (QPSK) and M-ary quadrature amplitude modulation (QAM) signals. While deep-unfolding NNs often lack solid theoretical support, we provide a detailed theoretical convergence analysis of the proposed schemes. As a benchmark, we develop a deep learning aided black-box based CNN for hybrid transceiver design. Furthermore, we analyze the transfer ability, computational complexity and generalization capability of the proposed deep-unfolding NN. Our simulation results show that the proposed deep-unfolding NN significantly outperforms the conventional algorithms and approaches the performance of the MSER-based iterative GD algorithm at a much reduced complexity. The main contributions of this work are summarized as follows:

- We propose a deep-unfolding NN based on the MSER-based iterative GD algorithm to expedite convergence while guaranteeing the performance of our hybrid transceiver design, wherein the iterative algorithm is unfolded into a multi-layer structure and a set of trainable parameters are introduced. Specifically, in the FP we introduce 1) the trainable matrix parameters corresponding to learning rates to replace the step sizes of the MSER-based GD algorithm; 2) the trainable parameters corresponding to the constant kernel width in the MSER-based GD algorithm; 3) the trainable offset matrix parameters for the computation of the beamforming matrices to increas the degrees of freedom. Note that the MSER-based iterative GD algorithm has not been unfolded in the literature, and our proposed designs have not been developed in the existing deep-unfolding algorithms.
- Since the existing platforms for implementation and training of NN (e.g. Pytorch or Tensorflow) are not designed to handle loss functions in the form of integrals, we seek to propose a novel method to compute the gradients in closed-form, which is more accurate and efficient. In

the BP, we derive the recursive relationship between the gradients of adjacent layers based on the GCR, and then compute the gradients w.r.t. the trainable parameters.

- We provide a theoretical analysis for the convergence of the proposed deep-unfolding NN. Specifically, we prove that the performance of a single layer in the proposed deep-unfolding NN is capable of approaching that of the MSER-based iterative GD algorithm using several iterations. Thus, the proposed deep-unfolding NN is capable of approaching the performance of the MSER-based iterative GD algorithm.
- We analyze the transfer capability, computational complexity and generalization capability. Through numerical simulations, we show that the proposed deep-unfolding NN substantially outperforms the conventional black-box method and approaches the performance of the MSER-based GD hybrid beamforming algorithm at a significantly reduced number of iterations, which translates into reduced complexity.

C. Organization

The paper is structured as follows. Section II briefly describes the system model. Section III formulates the MSER-based hybrid transceiver design problem for the case of QPSK and develops an MSER-based GD algorithm. The deep-unfolding NN is conceived in Section IV. In Section V, a black-box NN as a benchmark, the analysis of the computational complexity and generalization ability, and the proposed algorithm's extension to the QAM modulation are provided. The simulation results are presented in Section VI, and conclusions are drawn in Section VII.

Notations: Scalars, vectors and matrices are respectively denoted by lower case, boldface lower case and boldface upper case letters. $\mathbf{I}$ represents an identity matrix and $\mathbf{0}$ denotes an all-zero matrix. For a matrix $\mathbf{A}$, $\mathbf{A}^T$, $\mathbf{A}^*$, $\mathbf{A}^H$, $\mathbf{A}^\perp$, $\|\mathbf{A}\|$ and $[\mathbf{A}]_{m,n}$ denote its transpose, conjugate, conjugate transpose, reciprocal by element, Frobenius norm and element at the intersection of row m and column n, respectively. For a vector $\mathbf{a}$, $\|\mathbf{a}\|$ represents its Euclidean norm. $\mathbb{E}\{.\}$ denotes the statistical expectation. $\Re\{.\}$ ($\Im\{.\}$) denotes the real (imaginary) part of a variable. The operator $\mathrm{vec}(\cdot)$ stacks the columns of a matrix in one long column vector. $|\cdot|$ denotes the absolute value of a complex scalar. $\mathbb{C}^{m\times n}$ ($\mathbb{R}^{m\times n}$) denotes the space of $m \times n$ complex (real) matrices. The operator $\angle$ takes the phase angles of the elements in a matrix. The symbol $\circ$ denotes the Hadamard product of two vectors/matrices.

II. System Model

We consider the downlink of a hardware-efficient massive MIMO system as depicted in Fig. 1, which consists of one base station (BS) and K users. The BS is equipped with N_t transmit antennas and $R_t (R_t \ll N_t)$ RF chains. User $k \in \mathcal{K} \triangleq \{1, \dots, K\}$ is equipped with $N_{r,k}$ receive antennas and $R_{r,k} (R_{r,k} \le N_{r,k})$ RF chains, where $N_t \gg \sum_{k=1}^{K} N_{r,k}$ and $R_t \ge \sum_{k=1}^{K} R_{r,k}$. The BS transmits a symbol vector $\mathbf{s} \triangleq [\mathbf{b}_1^T, \mathbf{b}_2^T, \dots, \mathbf{b}_K^T]^T \in \mathbb{C}^{D\times 1}$ to the users, where the transmit symbols are independent and identically distributed (i.i.d.). The symbol vector of user $k \in \mathcal{K}$, $\mathbf{b}_k \triangleq [b_{1,k}, \dots, b_{D_k,k}]^T \in \mathbb{C}^{D_k\times 1}$, has zero-mean and covariance matrix $\mathbb{E}[\mathbf{b}_k\mathbf{b}_k^H] = \mathbf{I}$, where D_k ($D_k \le R_{r,k}$) denotes the number of data streams

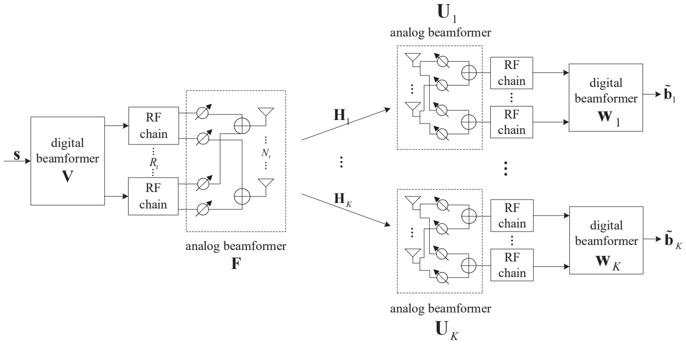

Fig. 1. Hardware-efficient massive MIMO system for downlink transmission.

for user k, and $D = \sum_{k=1}^{K} D_k$ denotes the total number of data streams. In this work, QPSK and M-ary square QAM symbol constellations are adopted, although extensions to other types of constellations are possible. For the QPSK case, the real and imaginary parts of each entry of the signal vector $\mathbf{b}_k$ are uniformly drawn from $\{\pm 1\}$. For the M-QAM modulation, each entry of $\mathbf{b}_k$ is uniformly drawn from $\{F_m + jF_n : 1 \le m, n \le \sqrt{M}\}$, where integer M is a perfect square and we define $F_n = 2n - \sqrt{M} - 1$. The transmitted symbol vector $\mathbf{s}$ is first processed by a digital transmit beamforming matrix $\mathbf{V} \triangleq [\mathbf{P}_1, \dots, \mathbf{P}_K] \in \mathbb{C}^{R_t\times D}$, where $\mathbf{P}_k \in \mathbb{C}^{R_t\times D_k}$ denotes the transmit beamforming matrix for user k. The digital beamformer output is passed through the RF chains and then processed by an analog transmit beamforming matrix $\mathbf{F} \in \mathbb{C}^{N_t\times R_t}$ implemented by means of phase shifters, i.e., $|[\mathbf{F}]_{m,n}| = 1$. The transmit power constraint at the BS is given by

$$\mathbb{E}\{\|\mathbf{FVs}\|^2\} = \|\mathbf{FV}\|^2 = P_T, \tag{1}$$

where P_T denotes the total transmit power budget.

The signal received at user k is given by

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{FVs} + \mathbf{n}_k = \mathbf{H}_k\mathbf{F}\sum_{k=1}^{K}\mathbf{P}_k\mathbf{b}_k + \mathbf{n}_k, \tag{2}$$

where $\mathbf{H}_k \in \mathbb{C}^{N_{r,k}\times N_t}$ denotes the massive MIMO channel matrix between the BS and user k. As discussed in [6] and [7], this matrix is conveniently described by a geometry clustered channel model with N_C clusters and N_R rays in each cluster due to the sparse scattering property of massive MIMO channels. Considering a system with a half-wave spaced uniform linear array at both the transmitter and the receiver, we have without loss of generality

$$\mathbf{H}_k = \sqrt{\frac{N_t N_{r,k}}{N_C N_R}}\sum_{i=1}^{N_C}\sum_{j=1}^{N_R}\alpha_{i,j}\mathbf{a}_r(\theta^r_{i,j})\mathbf{a}_t(\theta^t_{i,j})^H, \tag{3}$$

where $\alpha_{i,j}$ denotes the complex propagation gain of the j-th ray in the ith cluster, while $\mathbf{a}_r(\theta^r_{i,j}) \triangleq \frac{1}{\sqrt{N_{r,k}}}[1, e^{j\pi sin\theta^r_{i,j}}, \dots, e^{j\pi(N_r-1)sin\theta^r_{i,j}}]^T$ and $\mathbf{a}_t(\theta^t_{i,j}) \triangleq \frac{1}{\sqrt{N_t}}[1, e^{j\pi sin\theta^t_{i,j}}, \dots, e^{j\pi(N_t-1)sin\theta^t_{i,j}}]^T$ denote the corresponding normalized response vectors of the transmit and receive antenna arrays, with $\theta^r_{i,j}$ and $\theta^t_{i,j}$ denoting the angles of arrival and departure, respectively. The term $\mathbf{n}_k \in \mathbb{C}^{N_{r,k}\times 1}$

in (2) represents the additive noise at user k, modeled as a complex circularly symmetric Gaussian vector with zero-mean and covariance matrix $\mathbb{E}[\mathbf{n}_k\mathbf{n}_k^H]=\sigma_k^2\mathbf{I}$.

Similar to the BS, the hybrid AD receiver at user k is comprised of an analog receive beamforming matrix $\mathbf{U}_k \in \mathbb{C}^{R_{r,k}\times N_r}$, followed by a digital receive beamforming matrix $\mathbf{W}_k \triangleq [\mathbf{w}_{1,k},\ldots,\mathbf{w}_{D_k,k}] \in \mathbb{C}^{R_{r,k}\times D_k}$, where $\mathbf{w}_{i,k} \in \mathbb{C}^{R_{r,k}\times 1}$ and $i \in \mathcal{D}_k \triangleq \{1,\ldots,D_k\}$. Accordingly, the output signal vector $\tilde{\mathbf{b}}_k \triangleq [\tilde{b}_{1,k},\ldots,\tilde{b}_{D_k,k}]^T \in \mathbb{C}^{D_k\times 1}$ of the hybrid receiver at user k can be expressed as

$$\tilde{\mathbf{b}}_k = \mathbf{W}_k^H\mathbf{U}_k\mathbf{y}_k = \mathbf{W}_k^H\mathbf{U}_k(\mathbf{H}_k\mathbf{F}\sum_{k=1}^{K}\mathbf{P}_k\mathbf{b}_k+\mathbf{n}_k), \tag{4}$$

where the entries of the analog beamforming matrices obey the unit-modulus constraints, i.e., $|[\mathbf{U}_k]_{m,n}|=1$, $\forall k\in\mathcal{K}$.

Finally, the estimate of the transmitted symbol vector $\mathbf{b}_k$ at the output of receiver k is

$$\hat{\mathbf{b}}_k = \mathcal{Q}\{\tilde{\mathbf{b}}_k\}, \tag{5}$$

where $\mathcal{Q}\{\cdot\}$ is the quantization operation for the given modulation.

The basic problem in implementing the hybrid AD transceiver scheme is how to effectively design the beamforming matrices $\{\mathbf{P}_k,\mathbf{W}_k,\mathbf{U}_k,\mathbf{F}\}$ to detect the transmitted symbols accurately under the total transmit power constraint at the BS and the unit-modulus constraint imposed on each element of the analog RF beamforming matrices at the BS and user sides.

III. Proposed MSER-Based GD Algorithm for Hybrid AD Transceiver Design

In this section, we focus our attention on the QPSK case to formulate the MSER criterion mathematically. The resultant problem is very difficult to tackle due to the highly nonlinear objective function and constraints. We then develop an MSER-based iterative GD algorithm for finding the stationary points, which will serve as the basis for elaborating on our proposed deep unfolding NN.

A. MSER Criterion

The decisions for the detection of the real and imaginary parts of element $i\in\mathcal{D}_k$ in the k-th user's symbol vector $\mathbf{b}_k=[b_{1,k},...,b_{D_k,k}]^T$ are made as

$$\begin{aligned}\Re\{\hat{b}_{i,k}\} &= \begin{cases}+1, & \text{if } \Re\{\tilde{b}_{i,k}\}\geq 0\\ -1, & \text{if } \Re\{\tilde{b}_{i,k}\}<0,\end{cases}\\ \Im\{\hat{b}_{i,k}\} &= \begin{cases}+1, & \text{if } \Im\{\tilde{b}_{i,k}\}\geq 0\\ -1, & \text{if } \Im\{\tilde{b}_{i,k}\}<0.\end{cases}\end{aligned} \tag{6}$$

For a given desired symbol $b_{i,k}$, there exists $N_b=4^{D-1}$ legitimate combinations of the multi-user interference symbols $\{b_{i,k'}, i\in\mathcal{D}_{k'}, k'\in\mathcal{K}, k'\neq k\}$ and self-interference symbols $\{b_{i',k}, i'\in\mathcal{D}_k, i'\neq i\}$. We define the set of all the possible transmitted symbol vectors as

$$\mathcal{X}\triangleq\{\mathbf{s}^1,\mathbf{s}^2,\ldots,\mathbf{s}^{N_b}\}, \tag{7}$$

where $\mathbf{s}^q=[(\mathbf{b}_1^q)^T,(\mathbf{b}_2^q)^T,\ldots,(\mathbf{b}_K^q)^T]^T$, $q\in\mathcal{N}_b\triangleq\{1,\ldots,N_b\}$, and we assume an equiprobable model for the N_b possible transmit vectors $\mathbf{s}^q$. The noise-free component for the i-th element of the hybrid receiver's output at user k is from the set

$$\mathcal{Y}_{i,k}\triangleq\{\bar{b}_{i,k}^q=\mathbf{w}_{i,k}^H\mathbf{U}_k\mathbf{H}_k\mathbf{F}\mathbf{V}\mathbf{s}^q,\ q\in\mathcal{N}_b\}. \tag{8}$$

Due to the Gaussian distribution for the additive noise at the receiver, and invoking the law of total probability, we can express the probability density function (PDF) for the real part of the hybrid receiver's output, given the symbol $b_{i,k}$, as

$$\begin{aligned}f(x|b_{i,k}) =& \frac{1}{N_b\sqrt{2\pi\mathbf{w}_{i,k}^H\mathbf{U}_k\mathbf{H}_k\mathbf{F}\mathbf{V}\mathbf{V}^H\mathbf{F}^H\mathbf{H}_k^H\mathbf{U}_k^H\mathbf{w}_{i,k}}\sigma_n}\\ &\sum_{q=1}^{N_b} e^{-\frac{\left|x-\Re\{\bar{b}_{i,k}^q\}\right|^2}{2\mathbf{w}_{i,k}^H\mathbf{U}_k\mathbf{H}_k\mathbf{F}\mathbf{V}\mathbf{V}^H\mathbf{F}^H\mathbf{H}_k^H\mathbf{U}_k^H\mathbf{w}_{i,k}\sigma_n^2}},\end{aligned} \tag{9}$$

where $\bar{b}_{i,k}^q\in\mathcal{Y}$. Due to the large value of N_b, it will not be possible to consider the sum over all the possible transmitted symbol vectors. In practice, the PDF of the receiver's output should be approximated based on a block of experimental samples. Specifically, with the aid of kernel density estimation [40], we randomly select J different transmit symbol vectors from the set $\mathcal{X}$ in (7), indexed with q_j for $j\in\{1,...,J\}$, and employ a constant kernel width ϱ to replace the term $\sqrt{\mathbf{w}_{i,k}^H\mathbf{U}_k\mathbf{H}_k\mathbf{F}\mathbf{V}\mathbf{V}^H\mathbf{F}^H\mathbf{H}_k^H\mathbf{U}_k^H\mathbf{w}_{i,k}}\sigma_n$ appearing in (9) for complexity reduction. The parameter ϱ is related to the noise standard deviation and is selected based on separate simulations [40]. The block-data kernel estimate of the true PDF for the real part of the receiver output is

$$f(x|b_{i,k})=\frac{1}{J\sqrt{2\pi}\varrho}\sum_{j=1}^{J}e^{-\frac{|x-\Re\{\bar{b}_{i,k}^{q_j}\}|^2}{2\varrho^2}}. \tag{10}$$

The PDF for the imaginary part of the receiver's output can be obtained in the same way.

Furthermore, the SER of $b_{i,k}$ is given by

$$\mathcal{P}_e(b_{i,k})=\mathcal{P}_e^R(b_{i,k})+\mathcal{P}_e^I(b_{i,k})-\mathcal{P}_e^R(b_{i,k})\mathcal{P}_e^I(b_{i,k}), \tag{11}$$

where $\mathcal{P}_e^R(b_{i,k}) \triangleq \text{Prob}\left\{\Re\{\hat{b}_{i,k}\}\neq\Re\{b_{i,k}\}\right\}$ and $\mathcal{P}_e^I(b_{i,k}) \triangleq \text{Prob}\left\{\Im\{\hat{b}_{i,k}\}\neq\Im\{b_{i,k}\}\right\}$ represent the real-part and imaginary-part SER, respectively. Based on (10), we have

$$\mathcal{P}_e^R(b_{i,k})=\frac{1}{J\sqrt{\pi}}\sum_{j=1}^{J}\int_{-\infty}^{-\frac{\Re\{\bar{b}_{i,k}^{q_j}\}\Re\{b_{i,k}\}}{\sqrt{2}\varrho}} e^{-s^2}ds, \tag{12}$$

$$\mathcal{P}_e^I(b_{i,k})=\frac{1}{J\sqrt{\pi}}\sum_{j=1}^{J}\int_{-\infty}^{-\frac{\Im\{\bar{b}_{i,k}^{q_j}\}\Im\{b_{i,k}\}}{\sqrt{2}\varrho}} e^{-s^2}ds. \tag{13}$$

According to the discussion in [41], we can drop the product term $\mathcal{P}_e^R(b_{i,k})\mathcal{P}_e^I(b_{i,k})$ and focus on the upper bound $\tilde{\mathcal{P}}_e(b_{i,k})\triangleq\mathcal{P}_e^R(b_{i,k})+\mathcal{P}_e^I(b_{i,k})$ for reducing the computational complexity. Indeed, for small values of SER, $\tilde{\mathcal{P}}_e(b_{i,k})$ is very close to the true SER $\mathcal{P}_e(b_{i,k})$, i.e., the bound is tight. We aim to minimize the upper bound of the overall SER over all user data by jointly optimizing the beamforming matrices $\{\mathbf{P}_k,\mathbf{W}_k,\mathbf{U}_k,\mathbf{F}\}$, where index k runs through $\mathcal{K}$. Accordingly, the problem is formulated as

$$\min_{\substack{\{\mathbf{P}_k,\mathbf{W}_k,\\ \mathbf{U}_k,\mathbf{F}\}}} \sum_{k=1}^{K}\sum_{i=1}^{D_k}\tilde{\mathcal{P}}_e(b_{i,k}) \triangleq \sum_{k=1}^{K}\sum_{i=1}^{D_k}(\mathcal{P}_e^R(b_{i,k})+\mathcal{P}_e^I(b_{i,k})) \tag{14a}$$

$$\text{s.t. } \|\mathbf{F}\mathbf{V}\|^2 = P_T, \tag{14b}$$

$$|[\mathbf{F}]_{m,n}| = 1,\ |[\mathbf{U}_k]_{m,n}| = 1, \quad \forall k,m,n, \tag{14c}$$

where (14b) and (14c) are the transmit power and constant modulus constraints, respectively.

B. MSER-based GD Joint Beamforming Design Algorithm

In the following, we introduce the proposed MSER-based GD joint beamforming design algorithm. To tackle the unit-modulus constraints in (14c), we define the analog beamforming phase matrices $\boldsymbol{\theta}_{U_k} \triangleq \angle\mathbf{U}_k$ and $\boldsymbol{\theta}_F \triangleq \angle\mathbf{F}$. Consequently, $\mathbf{U}_k$ and $\mathbf{F}$ can be obtained by $\mathbf{U}_k = \exp(j\boldsymbol{\theta}_{U_k})$ and $\mathbf{F} = \exp(j\boldsymbol{\theta}_F)$, respectively, where the function $\exp(\cdot)$ is applied element-wise. Furthermore, to guarantee the transmit power constraint, we shall scale the overall digital transmit beamforming matrix $\mathbf{V} = [\mathbf{P}_1,\ldots,\mathbf{P}_K]$ at the end of each iteration of the proposed MSER-based GD algorithm. Therefore, let us consider the following unconstrained problem for simplicity:

$$\min_{\{\mathbf{P}_k,\mathbf{W}_k,\boldsymbol{\theta}_{U_k},\boldsymbol{\theta}_F\}} \sum_{k=1}^{K}\sum_{i=1}^{D_k}\tilde{\mathcal{P}}_e(b_{i,k}) \triangleq \sum_{k=1}^{K}\sum_{i=1}^{D_k}(\mathcal{P}_e^R(b_{i,k})+\mathcal{P}_e^I(b_{i,k})). \tag{15}$$

Based on the objective function of MSER, the gradient of $\tilde{\mathcal{P}}_e(b_{i,k})$ with respect to (w.r.t.) the hybrid beamforming matrices is given by $\nabla\tilde{\mathcal{P}}_e(b_{i,k}) \triangleq \nabla\mathcal{P}_e^R(b_{i,k}) + \nabla\mathcal{P}_e^I(b_{i,k})$, where $\nabla\mathcal{P}_e^R(b_{i,k})$ and $\nabla\mathcal{P}_e^I(b_{i,k})$ denote the gradients w.r.t. the real part and the imaginary part, respectively. Let us first focus on deriving the gradients w.r.t. the real parts of $\{\mathbf{P}_k,\mathbf{W}_k,\mathbf{U}_k,\mathbf{F}\}$. Specifically, computing the gradient of (12), as in [42], we obtain,

$$\begin{aligned}\nabla_{\mathbf{P}_k^*}\mathcal{P}_e^R = &-\frac{1}{J\sqrt{2\pi}\varrho}\sum_{j=1}^{J} e^{-\frac{|\Re\{\bar{b}_{i,k}^{q_j}\}|^2}{2\varrho^2}}\Re\{b_{i,k}\}\\ &\mathbf{F}^H\mathbf{H}_k^H\mathbf{U}_k^H\mathbf{W}_k(\mathbf{b}_k^{q_j})^H,\end{aligned} \tag{16}$$

$$\begin{aligned}\nabla_{\mathbf{w}_{i,k}^*}\mathcal{P}_e^R = &-\frac{1}{J\sqrt{2\pi}\varrho}\sum_{j=1}^{J} e^{-\frac{|\Re\{\bar{b}_{i,k}^{q_j}\}|^2}{2\varrho^2}}\Re\{b_{i,k}\}\\ &\mathbf{U}_k\mathbf{H}_k\mathbf{F}(\mathbf{V}\mathbf{s}^{q_j}),\end{aligned} \tag{17}$$

$$\begin{aligned}\nabla_{\mathbf{U}_k^*}\mathcal{P}_e^R = &-\frac{1}{J\sqrt{2\pi}\varrho}\sum_{j=1}^{J} e^{-\frac{|\Re\{\bar{b}_{i,k}^{q_j}\}|^2}{2\varrho^2}}\Re\{b_{i,k}\}\\ &\mathbf{w}_{i,k}(\mathbf{V}\mathbf{s}^{q_j})^H\mathbf{F}^H\mathbf{H}_k^H,\end{aligned} \tag{18}$$

$$\begin{aligned}\nabla_{\mathbf{F}^*}\mathcal{P}_e^R = &-\frac{1}{J\sqrt{2\pi}\varrho}\sum_{j=1}^{J} e^{-\frac{|\Re\{\bar{b}_{i,k}^{q_j}\}|^2}{2\varrho^2}}\Re\{b_{i,k}\}\\ &\mathbf{H}_k^H\mathbf{U}_k^H\mathbf{w}_{i,k}\,(\mathbf{V}\mathbf{s}^{q_j})^H,\end{aligned} \tag{19}$$

and we define $\nabla_{\mathbf{W}_k^*}\mathcal{P}_e^R \triangleq [\nabla_{\mathbf{w}_{1,k}^*}\mathcal{P}_e^R,\ldots,\nabla_{\mathbf{w}_{D_k,k}^*}\mathcal{P}_e^R]$.

The gradients w.r.t. the analog beamforming phase matrices $\boldsymbol{\theta}_{U_k}$ and $\boldsymbol{\theta}_F$ can be obtained as

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_{U_k}}\mathcal{P}_e^R &= \Re\{\nabla_{\mathbf{U}_k}\tilde{\mathcal{P}}_e \circ j\mathbf{U}_k - \nabla_{\mathbf{U}_k^*}\tilde{\mathcal{P}}_e \circ j\mathbf{U}_k^*\}\\ &= -\nabla_{\mathbf{U}_k}\mathcal{P}_e^R \circ \mathbf{U}_k^I - \nabla_{\mathbf{U}_k}\mathcal{P}_e^I \circ \mathbf{U}_k^R\\ &\quad + \nabla_{\mathbf{U}_k^*}\mathcal{P}_e^R \circ (\mathbf{U}_k^I)^* + \nabla_{\mathbf{U}_k^*}\mathcal{P}_e^I \circ (\mathbf{U}_k^R)^*,\end{aligned} \tag{20}$$

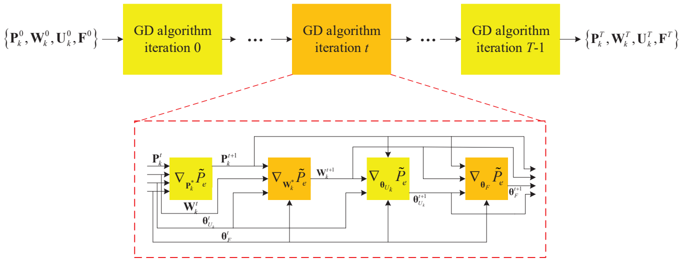
$\{\mathbf{P}_k^0,\mathbf{W}_k^0,\mathbf{U}_k^0,\mathbf{F}^0\}$
GD algorithm iteration 0
GD algorithm iteration t
GD algorithm iteration T-1
$\{\mathbf{P}_k^T,\mathbf{W}_k^T,\mathbf{U}_k^T,\mathbf{F}^T\}$
$\mathbf{P}_k^t$
$\nabla_{\mathbf{P}_k^*}\tilde{P}_e$
$\mathbf{P}_k^{t+1}$
$\mathbf{W}_k^t$
$\nabla_{\mathbf{W}_k^*}\tilde{P}_e$
$\mathbf{W}_k^{t+1}$
$\boldsymbol{\theta}_{U_k}^t$
$\nabla_{\boldsymbol{\theta}_{U_k}}\tilde{P}_e$
$\boldsymbol{\theta}_{U_k}^{t+1}$
$\boldsymbol{\theta}_F^t$
$\nabla_{\boldsymbol{\theta}_F}\tilde{P}_e$
$\boldsymbol{\theta}_F^{t+1}$

Fig. 2. Structure of the proposed MSER-based iterative GD algorithm.

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_F}\mathcal{P}_e^R &= \Re\{\nabla_{\mathbf{F}}\tilde{\mathcal{P}}_e \circ j\mathbf{F} - \nabla_{\mathbf{F}^*}\tilde{\mathcal{P}}_e \circ j\mathbf{F}^*\}\\ &= -\nabla_{\mathbf{F}}\mathcal{P}_e^R \circ \mathbf{F}^I - \nabla_{\mathbf{F}}\mathcal{P}_e^I \circ \mathbf{F}^R\\ &\quad + \nabla_{\mathbf{F}^*}\mathcal{P}_e^R \circ (\mathbf{F}^I)^* + \nabla_{\mathbf{F}^*}\mathcal{P}_e^I \circ (\mathbf{F}^R)^*.\end{aligned} \tag{21}$$

The gradients w.r.t. the imaginary parts of the beamforming matrices can be calculated similarly, thus the details are omitted for brevity.

The hybrid beamforming matrices are jointly optimized based on the MSER criterion. The GD update equations are obtained by substituting the gradients (16)–(21) in the following expressions

$$\mathbf{P}_k^{t+1} = \mathbf{P}_k^t - \mu_P\nabla_{\mathbf{P}_k^*}\tilde{\mathcal{P}}_e, \tag{22}$$

$$\mathbf{W}_k^{t+1} = \mathbf{W}_k^t - \mu_W\nabla_{\mathbf{W}_k^*}\tilde{\mathcal{P}}_e, \tag{23}$$

$$\boldsymbol{\theta}_{U_k}^{t+1} = \boldsymbol{\theta}_{U_k}^t - \mu_{\theta_U}\nabla_{\boldsymbol{\theta}_{U_k}}\tilde{\mathcal{P}}_e, \tag{24}$$

$$\boldsymbol{\theta}_F^{t+1} = \boldsymbol{\theta}_F^t - \mu_{\theta_F}\nabla_{\boldsymbol{\theta}_F}\tilde{\mathcal{P}}_e, \tag{25}$$

where $\{\mu_P,\mu_W,\mu_{\theta_U},\mu_{\theta_F}\}$ denote the step sizes employed in the GD iterations, and $t \in \{0,1,\ldots\}$ is the iteration index. The beamforming matrices are updated alternately until a certain convergence criterion is met. The iterative structure of the MSER-based GD algorithm structure is shown in Fig. 2.

To guarantee the transmit power constraint (14b), at the end of each GD iteration, the overall digital transmit beamforming matrix $\mathbf{V}$ needs to be scaled as follows:

$$\mathbf{V} \leftarrow \frac{\sqrt{P_T}}{\|\mathbf{F}\mathbf{V}\|}\mathbf{V}. \tag{26}$$

The details of the proposed MSER-based GD hybrid beamforming design algorithm are summarized in Algorithm 1. The latter is devised to start its operation in the training mode, where a known training transmit symbol sequence is employed, and then switch to the decision-directed mode, wherein the estimated symbols are used for computation.

IV. Proposed Deep-Unfolding NN for Hybrid AD Transceiver Design

The conventional MSER-based GD algorithms usually provide very slow convergence speed and therefore require a large number of iterations. Moreover, they suffer from performance degradation in the presence of channel state information (CSI) errors. To address these issues, we propose a deep-unfolding NN to jointly design the hybrid AD transceiver, where a small number of NN layers with trainable parameters can be employed while maintaining satisfactory performance. The MSER-based GD algorithm inspires the proposed NN structure. The iterative algorithm is unfolded into a multi-layer structure, and several adjustable step sizes and bias

Algorithm 1 Gradient-descent algorithm for joint hybrid beamforming design

```
1: Set the tolerance of accuracy $\epsilon$, the maximum number of iterations $T$, the size of the random sample $J$, and the step sizes $\{\mu_P, \mu_W, \mu_{\theta_U}, \mu_{\theta_F}\}$. Set the iteration index to $t = 0$.
2: Initialize $\mathbf{P}_k$ to satisfy the power constraint. Initialize $\{\mathbf{W}_k, \boldsymbol{\theta}_{U_k}, \boldsymbol{\theta}_F\}$.
3: repeat
4:     Update $\mathbf{P}_k^{t+1}$ with fixed $\{\mathbf{W}_k^t, \boldsymbol{\theta}_{U_k}^t, \boldsymbol{\theta}_F^t\}$, $\forall k \in \mathcal{K}$, according to (22).
5:     Update $\mathbf{W}_k^{t+1}$ with fixed $\{\mathbf{P}_k^{t+1}, \boldsymbol{\theta}_{U_k}^t, \boldsymbol{\theta}_F^t\}$, $\forall k \in \mathcal{K}$, according to (23).
6:     Update $\boldsymbol{\theta}_{U_k}^{t+1}$ with fixed $\{\mathbf{P}_k^{t+1}, \mathbf{W}_k^{t+1}, \boldsymbol{\theta}_F^t\}$, $\forall k \in \mathcal{K}$, according to (24).
7:     Update $\boldsymbol{\theta}_F^{t+1}$ with fixed $\{\mathbf{P}_k^{t+1}, \mathbf{W}_k^{t+1}, \boldsymbol{\theta}_{U_k}^{t+1}\}$, according to (25).
8:     Scale $\mathbf{P}_k^{t+1}$ based on (26) to meet the transmit power constraint.
9:     Update the iteration index: $t = t + 1$.
10: until The objective function meets chosen convergence criterion or $t > T$.
```

parameters are introduced in the FP. In the training stage, the relationship between the gradients of adjacent layers is derived according to the GCR in the BP. We then calculate the gradients w.r.t. the trainable parameters layer by layer and update these parameters based on the stochastic gradient descent (SGD) algorithm. We perform the FP process based on the trained parameters for computing the AD beamforming matrices in the testing stage. The details of the FP and BP in the proposed NN are presented as follows.

A. Forward Propagation

In this subsection, we describe the structure of the proposed deep-unfolding NN which is induced by the MSER-based GD algorithm developed in Section III. In the iterations of the latter algorithm, the step sizes used for updating the hybrid AD beamforming matrices, i.e., $\{\mu_P, \mu_W, \mu_{\theta_U}, \mu_{\theta_F}\}$ substantially affect the SER performance; furthermore, they are usually determined based on experience and experiments. Therefore, in layer $l \in \{0, \ldots, L-1\}$ of the proposed NN, where L denotes the number of layers, we introduce the trainable matrix parameters $\{\boldsymbol{\alpha}_{P_k}^l, \boldsymbol{\alpha}_{W_k}^l, \boldsymbol{\alpha}_{\theta_{U_k}}^l, \boldsymbol{\alpha}_{\theta_F}^l\}$ as the learning rates to replace the step sizes of the MSER-based GD algorithm. Recall that the term $\sqrt{\mathbf{w}_{i,k}^H \mathbf{U}_k \mathbf{H}_k \mathbf{F} \mathbf{V} \mathbf{V}^H \mathbf{F}^H \mathbf{H}_k^H \mathbf{U}_k^H \mathbf{w}_{i,k}} \sigma_n$ in (9) is set as a constant kernel width ϱ in the MSER-based GD algorithm, which may cause performance loss. Hence, we replace ϱ in (16)–(19) by the set of trainable parameters $\{\rho_{P_k}^l, \rho_{W_k}^l, \rho_{U_k}^l, \rho_F^l\}$. Moreover, to increase the degrees of freedom for the parameters, we introduce the trainable offset matrix parameters $\{\mathbf{O}_{P_k}^l, \mathbf{O}_{W_k}^l, \mathbf{O}_{\theta_{U_k}}^l, \mathbf{O}_{\theta_F}^l\}$ for the computation of the beamforming matrices. The update expressions for the proposed deep-unfolding NN are given by

$$\mathbf{P}_k^{l+1} = \mathbf{P}_k^l - \boldsymbol{\alpha}_{P_k}^l \circ \nabla_{\mathbf{P}_k^*} \tilde{\mathcal{P}}_e^l + \mathbf{O}_{P_k}^l, \tag{27}$$

$$\mathbf{W}_k^{l+1} = \mathbf{W}_k^l - \boldsymbol{\alpha}_{W_k}^l \circ \nabla_{\mathbf{W}_k^*} \tilde{\mathcal{P}}_e^l + \mathbf{O}_{W_k}^l, \tag{28}$$

$$\boldsymbol{\theta}_{U_k}^{l+1} = \boldsymbol{\theta}_{U_k}^l - \boldsymbol{\alpha}_{\theta_{U_k}}^l \circ \nabla_{\boldsymbol{\theta}_{U_k}} \tilde{\mathcal{P}}_e^l + \mathbf{O}_{\theta_{U_k}}^l, \tag{29}$$

$$\boldsymbol{\theta}_F^{l+1} = \boldsymbol{\theta}_F^l - \boldsymbol{\alpha}_{\theta_F}^l \circ \nabla_{\boldsymbol{\theta}_F} \tilde{\mathcal{P}}_e^l + \mathbf{O}_{\theta_F}^l, \tag{30}$$

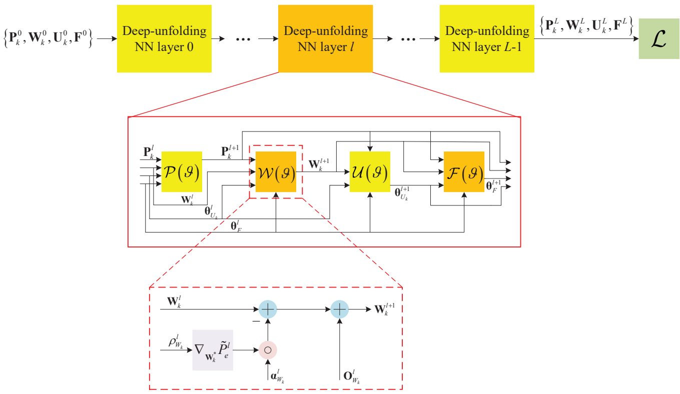
$\{\mathbf{P}_k^0, \mathbf{W}_k^0, \mathbf{U}_k^0, \mathbf{F}^0\}$
Deep-unfolding NN layer 0
Deep-unfolding NN layer l
Deep-unfolding NN layer L-1
$\{\mathbf{P}_k^L, \mathbf{W}_k^L, \mathbf{U}_k^L, \mathbf{F}^L\}$
$\mathcal{L}$
$\mathbf{P}_k^l$
$\mathcal{P}(\vartheta)$
$\mathbf{P}_k^{l+1}$
$\mathcal{W}(\vartheta)$
$\mathbf{W}_k^{l+1}$
$\mathcal{U}(\vartheta)$
$\mathcal{F}(\vartheta)$
$\mathbf{W}_k^l$
$\boldsymbol{\theta}_{U_k}^l$
$\boldsymbol{\theta}_F^l$
$\boldsymbol{\theta}_{U_k}^{l+1}$
$\boldsymbol{\theta}_F^{l+1}$
$\mathbf{W}_k^l$
$\mathbf{W}_k^{l+1}$
$-$
$\rho_{W_k}^l$
$\nabla_{\mathbf{W}_k^*} \tilde{P}_e^l$
$\circ$
$\boldsymbol{\alpha}_{W_k}^l$
$\mathbf{O}_{W_k}^l$

Fig. 3. Structure of the proposed deep-unfolding NN induced by the MSER-based GD algorithm.

where $\{\boldsymbol{\alpha}_{P_k}^l, \mathbf{O}_{P_k}^l\} \in \mathbb{C}^{R_t \times D_k}$, $\{\boldsymbol{\alpha}_{W_k}^l, \mathbf{O}_{W_k}^l\} \in \mathbb{C}^{R_{r,k} \times D_k}$, $\{\boldsymbol{\alpha}_{\theta_{U_k}}^l, \mathbf{O}_{\theta_{U_k}}^l\} \in \mathbb{C}^{R_{r,k} \times N_{r,k}}$, $\{\boldsymbol{\alpha}_{\theta_F}^l, \mathbf{O}_{\theta_F}^l\} \in \mathbb{C}^{N_t \times R_t}$, and $\{\nabla_{\mathbf{P}_k^*} \tilde{\mathcal{P}}_e^l, \nabla_{\mathbf{W}_k^*} \tilde{\mathcal{P}}_e^l, \nabla_{\boldsymbol{\theta}_{U_k}} \tilde{\mathcal{P}}_e^l, \nabla_{\boldsymbol{\theta}_F} \tilde{\mathcal{P}}_e^l\}$ denote the gradients w.r.t. the beamforming matrices in the l-th layer.

The structure of the proposed deep-unfolding NN induced by the MSER-based GD algorithm is illustrated in Fig. 3. Compared with Fig. 2, we can see that this structure is developed by unfolding the iterative GD algorithm into a multi-layer, comprised of L successive layers (top). The enlarged part in the red solid rectangle (middle) presents the common details of each layer in the deep-unfolding NN, where the operations $\mathcal{P}(\vartheta)$, $\mathcal{W}(\vartheta)$, $\mathcal{U}(\vartheta)$, and $\mathcal{F}(\vartheta)$ represent (27)–(30), respectively. Specifically, in layer $l \in \{0, \ldots, L-1\}$, we first update the digital beamforming matrices $\mathbf{P}_k$ and $\mathbf{W}_k$ successively, followed by the analog beamforming phase matrices $\boldsymbol{\theta}_{U_k}$ and $\boldsymbol{\theta}_F$. For each one of these updates, the enlarged diagram in the red dotted rectanlge (bottom) illustrates how the GD updates make use of the additional parameters $\boldsymbol{\alpha}_X^l$, ρ_X^l, and $\mathbf{O}_X^l$, where symbol $X \in \{P_k, W_k, \theta_{U_k}, \theta_F\}$. The beamforming matrices in the last layer are served as the NN outputs and are conveyed to the loss function. Since $\boldsymbol{\theta}_F$ is the last updated parameter matrix, the iteration rule of $\boldsymbol{\theta}_F^L$ adopts the original formula (25) without introducing additional parameters.

Since the channel matrices $\mathbf{H}_k$ are random by nature, the final loss function incorporates an expectation operation $\mathbb{E}_{\mathbf{H}}$ over the ensemble of channel matrices. Hence, the loss function $\mathcal{L}$ in the top-right corner of Fig. 3 is modified as

$$\begin{aligned}\mathcal{L} = \sum_{k=1}^{K} \mathbb{E}_{\mathbf{H}} \Bigg\{ \frac{1}{J\sqrt{\pi}} \sum_{i=1}^{D_k} \sum_{j=1}^{J} \Bigg(\int_{-\infty}^{-\frac{\Re\{\bar{b}_{i,k}^{q_j}\}\Re\{b_{i,k}\}}{\sqrt{2}\rho}} e^{-s^2}\, ds \\ + \int_{-\infty}^{-\frac{\Im\{\bar{b}_{i,k}^{q_j}\}\Im\{b_{i,k}\}}{\sqrt{2}\rho}} e^{-s^2}\, ds \Bigg) \Bigg\}.\end{aligned} \tag{31}$$

As mentioned in Section III, $\mathbf{V} = [\mathbf{P}_1, \ldots, \mathbf{P}_K]$ is scaled based on (26) in each layer to satisfy the transmit power constraint, which also helps avoid gradient explosion.

B. Back Propagation

Since the conventional platforms for implementation and training of NN (e.g. Pytorch or Tensorflow) are not designed to handle loss functions in the form of integrals, we seek to propose a novel method to compute the gradients in closed-form, which is more accurate and efficient. In the BP, we derive the recursive relation between the gradients of adjacent layers based on the GCR which is given in Appendix A, and then compute the gradients w.r.t. the trainable parameters.

Let $\{\mathbf{G}_{P_k}^l, \mathbf{G}_{W_k}^l, \mathbf{G}_{U_k}^l, \mathbf{G}_{U_k^*}^l, \mathbf{G}_F^l, \mathbf{G}_{F^*}^l\}$ denote the gradients w.r.t. the hybrid AD beamforming matrices in the l-th layer. By taking the derivative of (31), we can calculate the gradients w.r.t. $\mathbf{F}^L$ and $(\mathbf{F}^*)^L$ in the last layer as

$$\mathbf{G}_F^L = (\nabla_{\mathbf{F}} \tilde{\mathcal{P}}_e^L)^H, \quad \mathbf{G}_{F^*}^L = (\nabla_{\mathbf{F}^*} \tilde{\mathcal{P}}_e^L)^H. \tag{32}$$

By differentiating on both sides of (20) and (21), it is readily seen that $\mathbf{G}_{\theta_{U_k}}^l$ and $\mathbf{G}_{\theta_F}^l$ can be computed based on $\{\mathbf{G}_{U_k}^l, \mathbf{G}_{U_k^*}^l\}$ and $\{\mathbf{G}_F^l, \mathbf{G}_{F^*}^l\}$, respectively, as

$$\mathbf{G}_{\theta_{U_k}}^l = \mathbf{G}_{U_k}^l \circ j\mathbf{U}_k^T - \mathbf{G}_{U_k^*}^l \circ j\mathbf{U}_k^*, \tag{33}$$

$$\mathbf{G}_{\theta_F}^l = \mathbf{G}_F^l \circ j\mathbf{F}^T - \mathbf{G}_{F^*}^l \circ j\mathbf{F}^*. \tag{34}$$

Next, we derive the recursive relationship between the gradients w.r.t. the hybrid beamforming matrices in the $(l+1)$-th layer and the l-th layer. To this end, we first take the derivative on both sides of the equations (27)–(30) and apply the differential multiplication rules. Let us first expand (27) and provide the relationship between $\mathbf{G}_{P_k}^{l+1}$ and $\mathbf{G}_{P_k}^l$; the corresponding relations for the other matrices can be derived similarly. To simplify the presentation, we omit the index of data stream i, the index of transmit vector q_j, and the index of iteration t. Based on (27) and the GCR shown in Appendix A, we have

$$\begin{aligned}
&\mathrm{Tr}\left\{\mathbf{G}_{P_k}^{l+1} d\mathbf{P}_k^{l+1}\right\} = \frac{1}{J}\sum_{j=1}^{J} \mathrm{Tr}\left\{\mathbf{G}_{P_k}^{l+1} d\mathbf{P}_k^l - b_k^H B\, \mathbf{G}_{P_k}^{l+1} \circ \right. \\
&(\boldsymbol{\alpha}_{P_k}^l)^T \ (\mathbf{F}^l)^H \mathbf{H}_k^H (\mathbf{U}_k^l)^H \mathbf{W}_k^l \circ \big[b_k\, (\mathbf{W}_k^l)^H \mathbf{U}_k^l \mathbf{H}_k \mathbf{F}^l \\
&d\mathbf{P}_k^l + \mathbf{D}^H (\mathbf{U}_k^l)^H d\mathbf{W}_k^l + \mathbf{C}(\mathbf{W}_k^l)^H \mathbf{U}_k^l \mathbf{H}_k\, d\mathbf{F}^l + \mathbf{D} \\
&(\mathbf{W}_k^l)^H d\mathbf{U}_k^l + \mathbf{H}_k^H (\mathbf{U}_k^l)^H \mathbf{W}_k^l \mathbf{C}^H d(\mathbf{F}^H)^l + \mathbf{W}_k^l \mathbf{D}^H \\
&d(\mathbf{U}_k^H)^l\big] + b_k^H A\, \mathbf{G}_{P_k}^{l+1} \circ (\boldsymbol{\alpha}_{P_k}^l)^T (\mathbf{F}^l)^H \mathbf{H}_k^H (\mathbf{U}_k^l)^H \\
&d\mathbf{W}_k^l + b_k^H A\, \mathbf{H}_k^H (\mathbf{U}_k^l)^H \mathbf{W}_k^l \mathbf{G}_{P_k}^{l+1} \circ (\boldsymbol{\alpha}_{P_k}^l)^T d(\mathbf{F}^H)^l \\
&\left. + b_k^H A\, \mathbf{W}_k^l \mathbf{G}_{P_k}^{l+1} \circ (\boldsymbol{\alpha}_{P_k}^l)^T (\mathbf{F}^l)^H \mathbf{H}_k^H d(\mathbf{U}_k^H)^l \right\},
\end{aligned} \tag{35}$$

where $A \triangleq \frac{\Re\{b_k\}}{\sqrt{2\pi}\rho_{P_k}^l} e^{-\frac{|\Re\{\bar{b}_k\}|^2}{2(\rho_{P_k}^l)^2}}$, $B \triangleq A\frac{\Re\{b_k\}}{(\rho_{P_k}^l)^2}$, $\mathbf{C} \triangleq \sum_{i=1}^{D_k} \mathbf{p}_{i,k}^l b_{i,k}$, and $\mathbf{D} \triangleq \mathbf{H}_k \mathbf{F}^l \mathbf{C}$. By proceeding in the same way with (28)–(30), we obtain all the necessary relations linking the various gradients in adjacent layers.

By isolating and rearranging corresponding terms in $d\mathbf{P}_k^l$, we obtain the desired relationship for $\mathbf{G}_{P_k}^l$ as

$$\begin{aligned}
\mathbf{G}_{P_k}^l = &-\frac{1}{J}\sum_{j=1}^{J} \Big(b_k^H B\, \mathbf{G}_{P_k}^{l+1} \circ (\boldsymbol{\alpha}_{P_k}^l)^T (\mathbf{F}^l)^H \mathbf{H}_k^H (\mathbf{U}_k^l)^H \\
&\circ \mathbf{W}_k^l \mathbf{E} - B\, \mathbf{G}_{W_k}^{l+1} \circ (\boldsymbol{\alpha}_{W_k}^l)^T \mathbf{U}_k^l \mathbf{D} \circ \mathbf{E} \\
&+ b_k A\, \left[\mathbf{G}_F^{l+1} \circ (\boldsymbol{\alpha}_{\theta_F}^l)^T \mathbf{H}_k^T (\mathbf{U}_k^{l+1})^T (\mathbf{W}_k^l)^H\right]^T \\
&- b_k A\, \mathbf{G}_{W_k}^{l+1} \circ (\boldsymbol{\alpha}_{W_k}^l)^T \mathbf{U}_k^l \mathbf{H}_k \mathbf{F}^l + b_k A\, [(\mathbf{F}^l)^T \mathbf{H}_k^T \\
&\mathbf{G}_{U_k}^{l+1} \circ (\boldsymbol{\alpha}_{\theta_{U_k}}^l)^T (\mathbf{W}_k^l)^H]^T - B\, \mathbf{D}^T \mathbf{G}_{U_k}^{l+1} \circ (\boldsymbol{\alpha}_{\theta_{U_k}}^l)^T \\
&(\mathbf{W}_k^l)^H \circ \mathbf{E} - B\, \mathbf{C}^H \mathbf{G}_{F^*}^{l+1} \circ (\boldsymbol{\alpha}_{\theta_F}^l)^T \mathbf{H}_k^H (\mathbf{U}_k^{l+1})^H \\
&\mathbf{W}_k^l \circ \mathbf{E} - B\, \mathbf{D}^H \mathbf{G}_{U_k^*}^{l+1} \circ (\boldsymbol{\alpha}_{\theta_{U_k}}^l)^T \mathbf{W}_k^l \circ \mathbf{E} - B\, \mathbf{C}^T \\
&\mathbf{G}_F^{l+1} \circ (\boldsymbol{\alpha}_{\theta_F}^l)^T \mathbf{H}_k^T (\mathbf{U}_k^{l+1})^T (\mathbf{W}_k^l)^H \circ \mathbf{E}\Big) + \mathbf{G}_{P_k}^{l+1},
\end{aligned} \tag{36}$$

where $\mathbf{E} \triangleq b_k\, (\mathbf{W}_k^l)^H \mathbf{U}_k^l \mathbf{H}_k \mathbf{F}^l$. The gradients w.r.t. the other beamforming matrices in the l-th layer can be obtained similarly. Furthermore, the gradients w.r.t. the introduced parameters in each layer are computed based on (27)–(30). The detailed expressions of $\{\nabla_{\boldsymbol{\alpha}_X} \tilde{\mathcal{P}}_e^l, \nabla_{\rho_X} \tilde{\mathcal{P}}_e^l, \nabla_{\mathbf{O}_X} \tilde{\mathcal{P}}_e^l\}$, where $X \in \{P_k, W_k, \theta_{U_k}, \theta_F\}$, are shown in Appendix B.

We calculate the average gradient in a batch and implement the SGD method to update the trainable parameters, such as in, e.g., $\boldsymbol{\alpha}_{P_k}^{l,t+1} = \boldsymbol{\alpha}_{P_k}^{l,t} - \mu_{\alpha_{P_k}}^t \nabla_{\boldsymbol{\alpha}_{P_k}^t} \tilde{\mathcal{P}}_e^l$, where $\mu_{\alpha_{P_k}}^t$ denotes the step size of the update in SGD of $\boldsymbol{\alpha}_{P_k}^l$ in the t-th iteration, which incorporates an attenuation factor dependent on iteration t. In order to avoid vanishing gradient problem in the BP, normalization is employed after the update of $\{\mathbf{G}_{P_k}^l, \mathbf{G}_{W_k}^l, \mathbf{G}_{U_k}^l, \mathbf{G}_{U_k^*}^l, \mathbf{G}_F^l, \mathbf{G}_{F^*}^l\}$. The specific rule of normalizing $\mathbf{G}_{P_k}^l$ is given as

$$\mathbf{G}_{P_k}^l \leftarrow \frac{K\mathbf{G}_{P_k}^l}{\sum\limits_k \left\|\mathbf{G}_{P_k}^l\right\|}. \tag{37}$$

The gradients w.r.t. the other variables can be computed in the same way. The trainable parameters are initialized randomly and the beamforming matrices $\{\mathbf{P}_k^0, \mathbf{W}_k^0, \boldsymbol{\theta}_{U_k}^0, \boldsymbol{\theta}_F^0\}$ are initialized based on the conventional channel alignment method [6]. The training process of the deep-unfolding NN is shown in Algorithm 2, where $\mathcal{H} \triangleq \{\mathbf{H}_k^1, \mathbf{H}_k^2, \ldots, \mathbf{H}_k^N\}$ and N is determined by simulations.

V. Algorithm Analysis

In this section, we demonstrate that the sequence of iterates generated by the proposed deep-unfolding NN is convergent. Then we develop a black-box CNN as a benchmark to jointly optimize the hybrid AD beamforming matrices. Moreover, we analyze the computational complexity and generalization ability of the proposed schemes. Finally, we extend the deep-unfolding NN to M-QAM signal constellations.

A. Convergence of deep-unfolding NN

In general, no claim of guaranteed convergence can be made for existing deep-unfolding NNs due to the introduction of trainable parameters in the deep-unfolding NN as well as structural differences between the latter and the original iterative optimization algorithm. In the following, we circumvent some of these difficulties and provide novel theoretical insights into the convergence of the deep-unfolding NN. It is difficult to strictly prove its convergence. In the following, we provide some theoretical analysis for the convergence of the deep-unfolding NN.

Algorithm 2 Training process of the proposed deep-unfolding NN induced by GD algorithm

```
1: Generate training data set $\{\mathcal{X},\mathcal{H}\}$. Set tolerance of accuracy $\epsilon$, number of layers $L$, batch size $N$, the maximum number of iterations $I_{\max}$, and the size of the random sample $J$. Set the current iteration index $t=0$. Initialize the beamforming matrices, trainable parameters, and step sizes.
2: repeat
3:     Forward propagation: Randomly select $J$ samples $\{\mathbf{s},\mathbf{H}_k,\forall k\}$ from $\{\mathcal{X},\mathcal{H}\}$. Calculate $\{\boldsymbol{\theta}_F^l,\mathbf{F}^l,l=1,\dots,L-1\}$ and $\{\mathbf{P}_k^l,\mathbf{W}_k^l,\boldsymbol{\theta}_{U_k}^l,\mathbf{U}_k^l,l=1,\dots,L,\forall k\}$ based on (27)–(30).
4:     Calculate $\boldsymbol{\theta}_F^L$ based on (25). Calculate $\mathbf{F}^L$ based on $\boldsymbol{\theta}_F^L$. Substitute $\{\mathbf{P}_k^L,\mathbf{W}_k^L,\mathbf{U}_k^L,\mathbf{F}^L\}$ into (31).
5:     Backward propagation: Firstly, calculate the gradients of $\{\mathbf{F}^L,(\mathbf{F}^L)^*\}$ in the last layer and the gradient of $\boldsymbol{\theta}_F^L$ based on (34). Secondly, calculate the gradients of $\{\mathbf{P}_k^l,\mathbf{W}_k^l,\mathbf{U}_k^l,(\mathbf{U}_k^l)^*,\boldsymbol{\theta}_{U_k}^l,l=L,\dots,0,\forall k\}$ and $\{\mathbf{F}^l,(\mathbf{F}^l)^*,\boldsymbol{\theta}_F^l,l=L-1,\dots,0\}$ based on Appendix A. Finally, calculate the gradients of the trainable parameters based on Appendix B.
6:     Calculate the average gradient in a batch and update the trainable parameters based on the SGD method.
7:     Update the iteration number : $t=t+1$.
8: until The loss function in the validation data set converges or $t>I_{\max}$.
```

Theorem 1 *(Convergence of deep-unfolding NN): The performance of one layer in the deep-unfolding NN can approach that of several iterations in the GD algorithm if the parameters are properly trained. Consequently:*

1) *The performance of several layers in the deep-unfolding NN can approach that of the iterative GD algorithm.*
2) *The deep-unfolding NN converges to a statonary point with much reduced number of layers.*

The proofs of these claim can be provided as follows. As shown in Section III, in the t-th iteration of the GD algorithm, we have the following mapping from $\mathbf{P}_t$ to $\mathbf{P}_{t+2}$, where we omit indices k and i for clarity:

$$\mathbf{P}_{t+2}=\mathbf{P}_t-\mu_P A\sum_{j=1}^{J}e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2}{2\varrho^2}}\Re\{b\}\mathbf{F}_t^H\mathbf{H}^H\mathbf{U}_t^H\mathbf{W}_t(\mathbf{b}^{q_j})^H$$
$$-\mu_P A\sum_{j=1}^{J}e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2}{2\varrho^2}}\Re\{b\}\mathbf{F}_{t+1}^H\mathbf{H}^H\mathbf{U}_{t+1}^H\mathbf{W}_{t+1}(\mathbf{b}^{q_j})^H. \tag{38}$$

Similarly, in the deep-unfolding NN, we have the following mapping from $\mathbf{P}_l$ to $\mathbf{P}_{l+1}$ in the l-th layer:

$$\mathbf{P}_{l+1}=\mathbf{P}_l-\boldsymbol{\alpha}_P^l\circ\Big(A\sum_{j=1}^{J}e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2}{2(\rho_P^l)^2}}\Re\{b\}\mathbf{F}_l^H\mathbf{H}^H\mathbf{U}_l^H\mathbf{W}_l(\mathbf{b}^{q_j})^H\Big)+\mathbf{O}_P^l. \tag{39}$$

On the basis of these relations, we prove that one layer in the deep-unfolding NN can approach two iterations in the GD algorithm. In detail, when the initial values of the two algorithms are identical, i.e., $\mathbf{P}_l=\mathbf{P}_t$, we need to prove that $\mathbf{P}_{l+1}$ approaches $\mathbf{P}_{t+2}$, i.e., $\|\mathbf{P}_{t+2}-\mathbf{P}_{l+1}\|^2<\varepsilon$, for any $\varepsilon>0$. In the following, we provide the analytical justification for two different cases.

1) ***Case 1*** *(Fixed, i.e., deterministic channel):* When the channel matrix $\mathbf{H}$ is fixed or only changes very slowly, we are supposed to demonstrate that there exist trainable parameters $\boldsymbol{\alpha}_X$, ρ_X, and $\mathbf{O}_X$ such that $\|\mathbf{P}_{t+2}-\mathbf{P}_{l+1}\|^2<\varepsilon$ is satisfied for a given $\mathbf{H}$. By comparing (38) with (39), we can make $\mathbf{P}_{t+2}=\mathbf{P}_{l+1}$ satisfied if we set

$$\boldsymbol{\alpha}_P^l=\mu_P\,\mathbf{1}^{R_t\times D_k}, \tag{40a}$$
$$\mathbf{O}_P^l=-\mu_P A\sum_{j=1}^{J}e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2}{2\varrho^2}}\Re\{b\}\mathbf{F}_{t+1}^H\mathbf{H}^H \tag{40b}$$
$$\mathbf{U}_{t+1}^H\mathbf{W}_{t+1}(\mathbf{b}^{q_j})^H,$$
$$\rho_P^l=\varrho, \tag{40c}$$

where $\mathbf{1}^{R_t\times D_k}$ denotes the matrix with dimension $R_t\times D_k$ and all elements equal to 1.

2) ***Case 2*** *(Channel Following Certain Distribution):* When the channel matrix $\mathbf{H}$ conforms to a certain distribution, we have to illustrate the existence of parameters $\boldsymbol{\alpha}_X$, ρ_X, and $\mathbf{O}_X$ such that the following inequality is satisfied:

$$\mathbb{E}_{\mathbf{H}}\Big\{\|\mathbf{P}_{t+2}-\mathbf{P}_{l+1}\|^2\Big\}\le\varepsilon. \tag{41}$$

Setting $\boldsymbol{\alpha}_P^l=\mu_P\,\mathbf{1}^{R_t\times D_k}$ and $\rho_P^l=\varrho$, we need to prove

$$\mathbb{E}_{\mathbf{H}}\Big\{\|\mathbf{O}_P-\mu_P A\sum_{j=1}^{J}e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2}{2\varrho^2}}\Re\{b\}\mathbf{F}_{t+1}^H\mathbf{H}^H\mathbf{U}_{t+1}^H\mathbf{W}_{t+1}(\mathbf{b}^{q_j})^H\|^2\Big\}\le\varepsilon. \tag{42}$$

The variables $\mathbf{F}_{t+1}$, $\mathbf{U}_{t+1}$, and $\mathbf{W}_{t+1}$ are all related to $\mathbf{P}_t$. To simplify the presentation, we only expand $\mathbf{W}_{t+1}$ here but the other variables can be handled similarly. The left side of (42) can be expressed as

$$\mathbb{E}_{\mathbf{H}}\Big\{\|\mathbf{O}_P-\mu_P A\sum_{j=1}^{J}e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2}{2\varrho^2}}\Re\{b\}\mathbf{F}_{t+1}^H\mathbf{H}^H\mathbf{U}_{t+1}^H\mathbf{W}_{t+1}(\mathbf{b}^{q_j})^H\|^2\Big\}$$
$$=\mathbb{E}_{\mathbf{H}}\Big\{\|\mathbf{O}_P-\mu_P A\sum_{j=1}^{J}e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2}{2\varrho^2}}\Re\{b\}\mathbf{F}_{t+1}^H\mathbf{H}^H\mathbf{U}_{t+1}^H\mathbf{W}_t(\mathbf{b}^{q_j})^H\|^2\Big\}$$
$$+\mu_P\mu_W A^2\sum_{j=1}^{J}\sum_{m=1}^{J}e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2+|\Re\{\bar{b}^{q_m}\}|^2}{2\varrho^2}}\mathbf{F}_{t+1}^H\mathbf{H}^H\mathbf{U}_{t+1}^H\mathbf{U}_t\mathbf{H}\mathbf{F}_t\sum_{k'=1}^{K}\big(\mathbf{P}_t$$
$$\mathbf{b}^{q_m}(\mathbf{b}^{q_j})^H\big)-\mu_P^2\mu_W A^3\sum_{j=1}^{J}\sum_{m=1}^{J}\sum_{n=1}^{J}e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2+|\Re\{\bar{b}^{q_m}\}|^2+|\Re\{\bar{b}^{q_n}\}|^2}{2\varrho^2}}\Re\{b\}$$
$$\mathbf{F}_{t+1}^H\mathbf{H}^H\mathbf{U}_{t+1}^H\mathbf{U}_t\mathbf{H}\mathbf{F}_t\mathbf{F}_t^H\mathbf{H}^H\mathbf{U}_t^H\mathbf{W}_t(\mathbf{b}^{q_n})^H\sum_{k'=1}^{K}\mathbf{b}^{q_m}(\mathbf{b}^{q_j})^H\|^2\Big\}$$
$$\le\mathbb{E}_{\mathbf{H}}\Big\{\|\mathbf{O}_P-\mu_P A\sum_{j=1}^{J}e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2}{2\varrho^2}}\Re\{b\}\mathbf{F}_{t+1}^H\mathbf{H}^H\mathbf{U}_{t+1}^H\mathbf{W}_t(\mathbf{b}^{q_j})^H\|^2\Big\}$$
$$+\mathbb{E}_{\mathbf{H}}\Big\{\sum_{j=1}^{J}\sum_{m=1}^{J}\sum_{k'=1}^{K}\|\mu_P\mu_W A^2e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2+|\Re\{\bar{b}^{q_m}\}|^2}{2\varrho^2}}\|^2\|\mathbf{F}_{t+1}^H\mathbf{H}^H\mathbf{U}_{t+1}^H\|^2$$
$$\|\mathbf{U}_t\mathbf{H}\mathbf{F}_t\|^2\|\mathbf{P}_t\mathbf{b}^{q_m}(\mathbf{b}^{q_j})^H\|^2\Big\}+\mathbb{E}_{\mathbf{H}}\Big\{\sum_{j=1}^{J}\sum_{m=1}^{J}\sum_{n=1}^{J}\sum_{k'=1}^{K}\|\mu_P^2\mu_W A^3$$
$$e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2+|\Re\{\bar{b}^{q_m}\}|^2+|\Re\{\bar{b}^{q_n}\}|^2}{2\varrho^2}}\|^2\|\mathbf{F}_{t+1}^H\mathbf{H}^H\mathbf{U}_{t+1}^H\|^2\|\mathbf{U}_t\mathbf{H}\mathbf{F}_t\|^4$$
$$\|\mathbf{W}_t(\mathbf{b}^{q_n})^H\mathbf{b}^{q_m}(\mathbf{b}^{q_j})^H\|^2\Big\},$$

where the first equality is obtained based on (43) and (17), and the second inequality is derived based on *The Absolute Value Inequality*. The term $\mu_P A \sum_{j=1}^{J} e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2}{2\varrho^2}} \Re\{b\}\mathbf{F}_{t+1}^H \mathbf{H}^H \mathbf{U}_{t+1}^H \mathbf{W}_t (\mathbf{b}^{q_j})^H$ is a function of $\mathbf{H}$ and its mean value, denoted as γ can be calculated based on $\mathbb{E}\{\mathbf{H}\}$. According to *The Law of Large Numbers*, when we sample enough $\mathbf{H}$ in the calculation, as long as $\mathbf{O}_P$ is set to γ, (43) can be simplified as

$$\mathbb{E}_{\mathbf{H}}\Big\{||\mathbf{P}_{t+2} - \mathbf{P}_{l+1}||^2\Big\} \leq \mathbb{E}_{\mathbf{H}}\Big\{\sum_{j=1}^{J}\sum_{m=1}^{J}\sum_{k'=1}^{K}\|\mu_P \mu_W A^2$$
$$e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2 + |\Re\{\bar{b}^{q_m}\}|^2}{2\varrho^2}}\|^2 \|\mathbf{F}_{t+1}^H \mathbf{H}^H \mathbf{U}_{t+1}^H\|^2 \| \|\mathbf{U}_t \mathbf{H}\mathbf{F}_t\|^2 \|\mathbf{P}_t \mathbf{b}^{q_m} (\mathbf{b}^{q_j})^H\|^2\Big\}$$
$$+ \mathbb{E}_{\mathbf{H}}\Big\{\sum_{j=1}^{J}\sum_{m=1}^{J}\sum_{n=1}^{J}\sum_{k'=1}^{K}\|\mu_P^2 \mu_W A^3 e^{-\frac{|\Re\{\bar{b}^{q_j}\}|^2 + |\Re\{\bar{b}^{q_m}\}|^2 + |\Re\{\bar{b}^{q_n}\}|^2}{2\varrho^2}}\|^2$$
$$\|\mathbf{F}_{t+1}^H \mathbf{H}^H \mathbf{U}_{t+1}^H\|^2 \| \|\mathbf{U}_t \mathbf{H}\mathbf{F}_t\|^4 \|\mathbf{W}_t (\mathbf{b}^{q_n})^H \mathbf{b}^{q_m} (\mathbf{b}^{q_j})^H\|^2\Big\} = \varepsilon. \tag{44}$$

Through the experiment, we can see that the product terms in the right side of (44) are all less than 1, so ε is a term which is far less than 1. Indeed, ε is a loose boundary between the performance of the deep-unfolding NN and the iterative GD algorithm.

In the above, we have proved that the performance of one layer in the deep-unfolding NN can approach that of two iterations in the GD algorithm. It then naturally follows that the performance of one layer in the deep-unfolding NN with more complex structure can approach that of several iterations in the iterative GD algorithm. Thus, the proposed deep-unfolding NN can approach the merits of the GD algorithm. Moreover, since the GD algorithm is known to converge to a stationary point of its objective function, it follows that the proposed deep-unfolding NN also converges to a stationary point.

B. Benchmark CNN

Recently, the use of NNs to emulate the end-to-end signal transmission in communication systems has drawn considerable attention [21]–[26]. In this work, based on the NN structure developed in [22], we employ a composite structure comprised of 4 CNNs for the hybrid AD transceiver design, as shown in Fig. 4. Specifically, the two CNNs labeled "P_NN" and "θ_F_NN" in the top dotted rectangle at the transmitter, implement the digital transmit beamforming matrices $\mathbf{P}_k$ and the analog transmit beamforming phase matrix $\boldsymbol{\theta}_F$, respectively, where the original symbol vectors $\mathbf{b}_k$ are converted into the precoded signals. The CNNs labeled "W_NN" and "θ_U_NN" in the bottom dotted rectangle at the receiver implement the digital receive beamforming matrices $\mathbf{W}_k$ and the analog receive beamforming phase matrices $\boldsymbol{\theta}_{U_k}$, respectively, which use the received signals $\mathbf{y}_k$ to generate $\hat{\mathbf{b}}_k$. In the training stage, the SER is obtained by comparing the original symbol vector $\mathbf{b}_k$ with the detected symbol vector $\hat{\mathbf{b}}_k$ and the SER function (15) is defined as the loss function. Since standard CNN implementations (e.g. Tensorflow) cannot process complex-number matrices directly, the channel matrix $\mathbf{H}_k \in \mathbb{C}^{N_r \times N_t}$ is transformed into a $2 \times N_r \times N_t$ dimensional real-number tensor $\tilde{\mathbf{H}}_k$. The inputs of "θ_F_NN" and "θ_U_NN" are $\tilde{\mathbf{H}}_k$ and $\mathbf{b}_k$, and the outputs are used to generate a low-dimensional equivalent channel $\mathbf{H}_{eq}$, which are served as the inputs of "P_NN" and "W_NN".

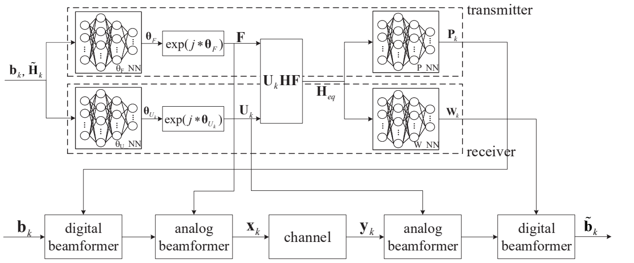

Fig. 4. Structure of the benchmark CNN for transceiver design.

Each CNN consists of the convolutional, pooling, fully-connected and batch normalization layers, the latter being implemented to avoid the gradient explosion. To reach a trade-off between the training overhead and system performance, we employ a relatively simple network structure, where the number of layers of each CNN is set to 17. In order to satisfy the transmit power constraint, the output of the "P_NN" needs to be normalized in the same way as in (26).

In the training stage, the adjustable weight parameters of the CNNs are updated by the SGD method (available in Tensorflow) while in the testing stage, only the FP is involved.

C. Dimension of Parameter Space

We first discuss the parameter space dimension of the proposed deep-unfolding NN followed by the benchmark CNN. The number of the parameters introduced in each layer of the deep-unfolding NN is $2(3K + KN_tD + KN_{r,k}D + KR_{r,k}N_{r,k} + R_tN_t)$. Since the parameters $\{\boldsymbol{\alpha}_{\theta_F}^L, \rho_F^L, \rho_{F^H}^L, \mathbf{O}_{\theta_F}^L\}$ for the analog transmit beamforming matrix $\mathbf{F}$ are not used in the last layer, i.e., L-th layer, the total number of parameter dimension in the network is $2KL(3 + N_tD + N_{r,k}D + R_{r,k}N_{r,k}) + 2(L-1)R_tN_t$.

In the benchmark CNN, model parameters are needed to implement the interconnection of the convolutional layers and the fully-connected layers. The total number of parameters involved in the convolutional layers is given by $\sum_{l=1}^{L_c} K_l C_{l-1} C_l$, where L_c denotes the number of convolutional layers, K_l denotes the size of convolution kernel, and C_l denotes the number of paths in the l-th convolutional layer. The parameter dimension of the fully-connected layers is given by $K^2 N_t N_{r,k} F_{in} F_{out}$, where F_{in} and F_{out} denote the input and output sizes of the fully-connected layers, respectively.

D. Computational Complexity

The computational complexity of the MSER-based iterative GD algorithm is given by $\mathcal{O}(TJS(KN_tD + KN_{r,k}D + KR_{r,k}N_{r,k} + R_tN_t))$, where T denotes the number of iterations and S denotes the size of the testing data set. As for the black-box CNN, the computational complexity is given by $\mathcal{O}(\sum_{l=2}^{L_c} L_c^2 K_l C_{l-1} C_l + K^2 N_t N_{r,k} F_{in} F_{out} + KN_tD + KN_{r,k}D + KR_{r,k}N_{r,k} + R_tN_t)$, where $L_c \triangleq (C_{in} - K_l + 2P)/S_t + 1$, C_{in} denotes the input size of the convolutional layers, P denotes the padding size, and S_t denotes the stride. The computational complexity of the proposed deep-unfolding

NN in the testing stage is $\mathcal{O}(LJS(KN_tD + KN_{r,k}D + KR_{r,k}N_{r,k} + R_tN_t))$, where L is the number of layers. In practice, for a comparable level of performance - as shown in Section VI - we find $L \ll T$; which means that the deep-unfolding NN can effectively reduce the computational complexity compared to its iterative GD counterpart. Also, the parameter space dimension and computational complexity of the deep-unfolding NN is substantially reduced compared to the benchmark CNN.

E. Generalization Capability

If a large-scale deep-unfolding NN has been trained, it can be employed to implement a network with smaller values of $(N_t, N_{r,k}, K)$ directly instead of training a new one. Suppose that the original large-scale NN is trained with $(N_{t0}, N_{r,k0}, K_0)$ and the smaller system is characterized by $(N_{t1}, N_{r,k1}, K_1)$, where $N_{t0} \leq N_{t1}$, $N_{r,k0} \leq N_{r,k1}$, and $K_0 \leq K_1$. In the testing stage, we only need to input $\{\mathbf{H}_k, k \leq K_1\}$ and set $\{\mathbf{H}_k = \mathbf{0}, K_1 < k \leq K_0\}$. Meanwhile, the corresponding column and row vectors in $\mathbf{H}_k$ should be set to $\mathbf{0}$.

F. Extension to M-QAM Modulation

The proposed deep-unfolding NN can also be extended to the higher-order modulation scheme, i.e., the M-QAM signals. In this case, the symbol is detected as

$$\Re\{\hat{b}_{i,k}\} = \begin{cases} F_1, & \text{if } \Re\{\tilde{b}_{i,k}\} \leq c_{i,k}(F_1+1) \\ F_m, & \text{if } c_{i,k}(F_m-1) < \Re\{\tilde{b}_{i,k}\} \leq c_{i,k}(F_m+1), \\ F_{\sqrt{M}}, & \text{if } \Re\{\tilde{b}_{i,k}\} > c_{i,k}(F_{\sqrt{M}-1}), \end{cases}$$
$$\Im\{\hat{b}_{i,k}\} = \begin{cases} F_1, & \text{if } \Im\{\tilde{b}_{i,k}\} \leq c_{i,k}(F_1+1) \\ F_n, & \text{if } c_{i,k}(F_n-1) < \Im\{\tilde{b}_{i,k}\} \leq c_{i,k}(F_n+1), \\ F_{\sqrt{M}}, & \text{if } \Im\{\tilde{b}_{i,k}\} > c_{i,k}(F_{\sqrt{M}-1}), \end{cases} \tag{45}$$

where $2 \leq m, n \leq \sqrt{M} - 1$ and $c_{i,k} \triangleq \mathbf{w}_{i,k}^H \mathbf{U}_k \mathbf{H}_k \mathbf{F} \mathbf{p}_{i,k}$. Generally, $c_{i,k}$ is a complex value and hence, the following phase rotations are utilized to guarantee that $c_{i,k}$ is real and positive:

$$\begin{aligned} \mathbf{p}_{i,k} &\leftarrow \frac{c_{i,k}}{|c_{i,k}|}\mathbf{p}_{i,k}, \quad \mathbf{w}_{i,k} \leftarrow \frac{c_{i,k}}{|c_{i,k}|}\mathbf{w}_{i,k}, \\ \mathbf{U}_k &\leftarrow \frac{c_{i,k}}{|c_{i,k}|}\mathbf{U}_k, \quad \mathbf{F} \leftarrow \frac{c_{i,k}}{|c_{i,k}|}\mathbf{F}, \end{aligned} \tag{46}$$

which make the detection rule (45) effective.

The number of the legitimate sequences of the transmitted signal $\mathbf{s}$ in the case of M-QAM modulation is $N_b = M^{D-1}$. Hence, similar to the case of QPSK modulation, we choose J different transmit symbol vectors and define the noise-free part of $\tilde{b}_{i,k}$ as $\bar{b}_{i,k}$. We then obtain the following expressions for the SER,

$$\mathcal{P}_e^R = \frac{\varphi}{J\sqrt{\pi}} \sum_{j=1}^{J} \int_{-\infty}^{-\frac{c_{i,k}\left(\Re\left\{b_{i,k}\right\}-1\right)-\Re\left\{\bar{b}_{i,k}^{q_j}\right\}}{\sqrt{2}\varrho}} e^{-s^2}\, ds, \tag{47}$$

$$\mathcal{P}_e^I = \frac{\varphi}{J\sqrt{\pi}} \sum_{j=1}^{J} \int_{-\infty}^{-\frac{c_{i,k}\left(\Im\left\{b_{i,k}\right\}-1\right)-\Im\left\{\bar{b}_{i,k}^{q_j}\right\}}{\sqrt{2}\varrho}} e^{-s^2}\, ds, \tag{48}$$

where $\varphi \triangleq \frac{2\sqrt{M}-2}{\sqrt{M}}$ [41].

The derivation of the MSER-based iterative GD algorithm for the M-QAM modulation follows similar steps as for the QPSK modulation, leading to similar updates as in (22)–(25) but where the expressions of the gradient vectors are modified accordingly. The deep-unfolding NN can be also developed in the same way, where the parameters introduced have the same function as in the QPSK case. The details of the two schemes are omitted for brevity. The feasibility of the deep-unfolding NN for the M-QAM signals is verified by the simulation results in Section VI.

VI. Simulation Results

In this section, we investigate the performance of the proposed MSER-based GD and deep-unfolding NN algorithms. We first present the simulation methodology, followed by the investigation of the convergence in training of the deep-unfolding NN. We then present the comparative performance results of these algorithms to the benchmark CNN and other approaches from the literature for both QPSK and M-QAM constellations in the context of massive MIMO transmissions.

A. Methodology

We consider downlink transmission in a multi-user MIMO system as illustrated in Fig. 1. The BS is equipped with $N_t = 64$ transmit antenna elements and $R_t = 8$ RF chains. We consider $K = 2$ users, each of which is equipped with $N_{r,k} = 8$ receive antennas and $R_{r,k} = 4$ RF chains. The BS transmits $D_k = 3$ data streams for each user, i.e., $\forall k \in \mathcal{K}$, consisting of either QPSK or M-QAM symbols. The channel matrix between the BS and the users is generated according to the model presented in (3).

The data set for training the deep unfolding NN is obtained by generating 500 channel matrices and using the transmitted and received symbols as true and target data. In this stage, the average of the loss function in (15) is used to approximate the expectation in (31). We set the batch size as $N = 20$ and the number of layers as $L = 15$ in the proposed deep-unfolding NN.

For the comparative performance evaluation, we generate 5000 channel matrices from which we obtain the testing data. In the case of the GD algorithm, we run the GD algorithm 50 times with randomly selected initial values and retain the best result as its performance, which can be treated as an upper bound. We consider the following algorithms for comparison:

- **GD**: The beamforming matrices are alternately optimized based on the GD algorithm, as developed in Section III.
- **OMP**: The near-optimal hybrid beamforming matrices are designed to approximate the optimal unconstrained beamforming matrices based on the OMP method in [6].
- **MO**: The hybrid beamforming matrices are alternately optimized based on the MO method in [11].
- **Benchmark CNN**: The hybrid beamforming matrices are optimized jointly by the proposed black-box CNN presented in Section V-B which is designed based on [22].
- **Proposed deep-unfolding NN**: The hybrid beamforming matrices are optimized jointly based on the proposed deep-unfolding NN developed in Section IV.

B. Convergence of Deep Unfolding NN

Fig. 5 illustrates the effects of the batch size on the convergence speed of the SER for the proposed deep-unfolding NN when SNR = 20 dB. As shown, a smaller batch size yields

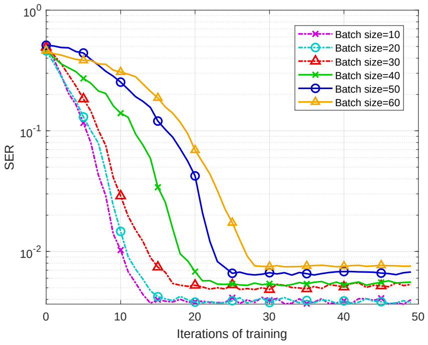

Fig. 5. SER of deep-unfolding NN versus iteration number for different batch sizes.

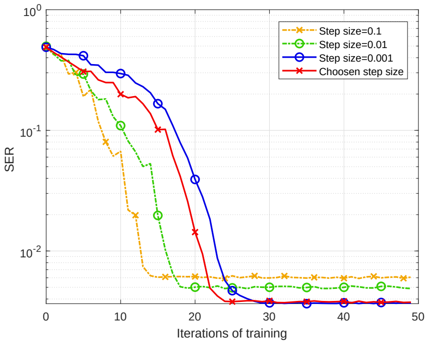

Fig. 6. SER of deep-unfolding NN versus iteration number for different step sizes.

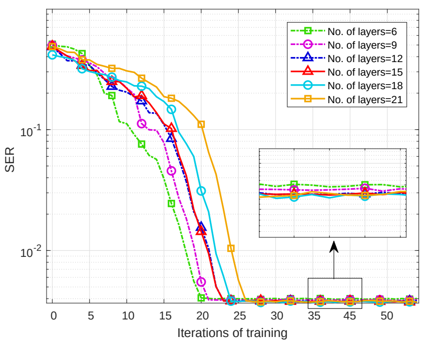

Fig. 7. SER of deep unfolding NN versus iteration number for different numbers of layers.

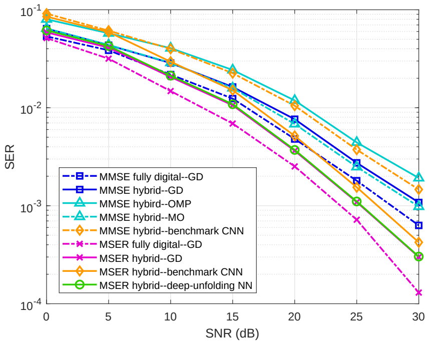

Fig. 8. SER performance versus SNR for the different algorithms ($R_{r,k} = 4$, $D = 3$).

a better SER performance in terms of both the convergence speed and residual SER after convergence. By comparison, a larger batch size has a smaller fluctuation of SER performance after convergence. Indeed, a batch size reduction introduces additional randomness in the gradient vector, which prevents the NN from being trapped in local optima. To strike an attractive tradeoff, we choose $N = 20$ as the batch size in the sequel. Fig. 6 shows the effects of the step size used in the deep-unfolding NN, i.e., $\{\mu_{\alpha_X}, \mu_{\rho_X}, \mu_{O_X}\}$, on the SER performance when SNR = 20 dB. The deep-unfolding NN with a larger step size requires fewer iterations to converge, but this comes at the price of increased residual SER after convergence. In this work, we opted for the step size of $0.02 \times 0.5^{\lfloor \frac{t}{10} \rfloor}$ for striking an attractive system performance and convergence speed tradeoff. Fig. 7 presents the effects of the number of layers L on the SER performance of the deep-unfolding NN. Increasing the number of layers slows down convergence (as expected) but also affects the residual SER after convergence. As L is increased from 6 to 18, the residual SER decreases until it reaches a lower bound but then starts to increase. Indeed, as L increases, the propagation of gradient errors and the effect of gradient disappearance also become more pronounced.

C. SER Performance for QPSK Signals

Fig. 8 shows the SER performance versus SNR in downlink MIMO transmission for the proposed deep-unfolding hybrid beamforming and benchmark algorithms. The results show that the proposed MSER-based algorithms significantly outperform the existing MMSE-based algorithms in terms of SER. Moreover, by making full use of the structure of the MSER-based hybrid GD beamforming algorithm and introducing the useful trainable parameters, the performance of the proposed deep-unfolding NN outperforms that of the black-box CNN benchmark algorithm and approaches that of the MSER-based hybrid GD algorithm. In addition, the MSER hybrid-benchmark CNN algorithm shows worse performance than the MMSE hybrid-MO algorithm in the low-SNR region, while the opposite holds in the high-SNR region. The reason is that in the low-SNR region, since the data is highly affected by the noise, the parameters in the CNN cannot be trained accurately, thus the MSER hybrid-benchmark CNN algorithm shows worse performance than the MMSE hybrid-MO algorithm. However, in the high-SNR region, the network parameters can be well trained and the hybrid-benchmark CNN is designed based on the MSER criterion, thus it can outperform the MMSE hybrid-MO algorithm in terms of SER. The parameters of the CNN are well tuned. The number of layers in the CNN is set to 17. We set the learning rate of the CNN to 0.005 in the early stage and change it to 0.001 for good performance.

Fig. 9 shows the convergence performance of the proposed deep-unfolding and MSER-based GD hybrid beamforming algorithms. We can see that the deep-unfolding hybrid beamforming algorithm significantly improves the convergence of the GD algorithm. The parameters of the proposed deep-

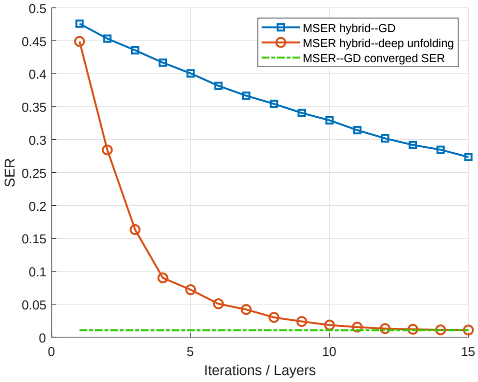

Fig. 9. Convergence behavior of the analyzed algorithms at SNR = 15 dB.

TABLE I
SER VERSUS SNR

	Hybrid GD 500 iterations	Benchmark CNN 17 layers	Deep-unfolding NN 15 layers
SNR = 5 dB	4.11×10^{-2}	5.92×10^{-2}	4.27×10^{-2}
SNR = 15 dB	1.06×10^{-2}	1.51×10^{-2}	1.09×10^{-2}
SNR = 25 dB	1.09×10^{-3}	1.54×10^{-3}	1.11×10^{-3}

unfolding NN have been optimized by considering the trade-off between complexity and performance, where we select 15 layers.

Table I presents the steady-state SER versus SNR for the proposed MSER-based deep-unfolding NN, hybrid GD beamforming algorithm and benchmark CNN, where the number of iterations/layers is emphasized for comparison. We find that the proposed deep-unfolding NN requires a smaller number of layers (15 layers) than the black-box CNN (17 layers) to achieve a similar (or better) steady-state performance, while a much larger number of iterations (500 iterations) is required by the GD algorithm to reach that steady-state performance level. Consequently, in the testing stage, the two NN schemes exhibit a much reduced computational complexity compared to the iterative GD scheme (see below).

Table II indicates the SER for different configurations of the MIMO system, i.e., different number of users K and number of receiving RF chains $R_{r,k}$, in the case of SNR = 20 dB. The percentages of the deep-unfolding NN in the table are calculated via dividing the SER of the iterative GD algorithm by those of the deep-unfolding NN (the percentages in the tables below are the same). The percentages of the benchmark CNN are calculated in the same way. We observe that the performance gap between the proposed deep-unfolding NN and the MSER-based GD algorithm slightly increases with K. Moreover, the gap between the SER of the general benchmark CNN and that of the deep-unfolding NN becomes larger with K. This could be the result of increased multi-user interference, which hinders the ability of the NN to learn from the training data. However, the proposed deep-unfolding NN always provides performance close to the MSER-based GD algorithm. Table III shows the SER performance of the NN schemes versus the number of training data samples for SNR = 20 dB. We observe that the deep-unfolding NN needs much fewer training data samples compared to the benchmark CNN since its structure is developed based on that of the MSER-based GD algorithm.

TABLE II
SER FOR DIFFERENT MIMO SYSTEM CONFIGURATIONS AT SNR = 20 DB

$(K,\, R_{r,k})$	(1, 4)	(2, 4)	(3, 2)	(4, 2)
Hybrid GD	$1.15{\times}10^{-3}$	$3.64{\times}10^{-3}$	$6.83{\times}10^{-3}$	$1.56{\times}10^{-2}$
Deep-unfolding NN	98.43%	97.85%	97.07%	95.86%
Benchmark CNN	73.14%	70.48%	68.52%	65.02%

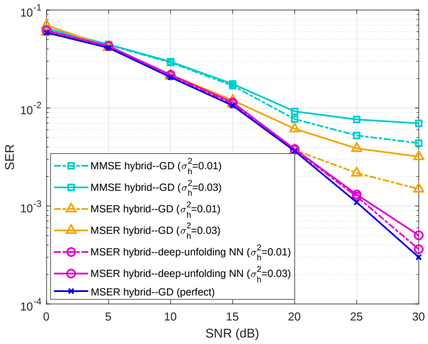

Fig. 10. SER performance of different schemes versus SNR in the presence of imperfect CSI at SNR = 20 dB.

Fig. 10 illustrates the SER performance versus SNR of the various hybrid AD MIMO transceiver designs in the presence of imperfect CSI. The channel estimation errors are represented by $\mathbf{H}_k = \bar{\mathbf{H}}_k + \sigma_h \Delta\mathbf{K}_k, \forall k \in \mathcal{K}$, where $\mathbf{H}_k$ is the true (synthesized) channel matrix, $\bar{\mathbf{H}}_k$ denotes its estimates, and the term $\sigma_h \Delta\mathbf{K}_k$ is the estimation error. In this latter term, $\Delta\mathbf{K}_k$ is a random matrix with zero-mean, unit variance uncorrelated elements following a circular complex Gaussian distribution, while σ_h^2 provides the estimation error variance, assumed identical $\forall k$ for simplicity. We can see that the SER performance degrades with the error variance σ_h^2. Moreover, the best performance is achieved by the proposed deep-unfolding NN, followed by the MSER-based GD beamforming algorithm and the MMSE-based GD beamforming algorithm. These results show that the proposed deep-unfolding NN has stronger robustness against the channel uncertainties compared to the other schemes.

Fig. 11 illustrates the transfer ability of the proposed schemes, by plotting their SER performance versus SNR for different channel scenarios. Specifically, in this experiment, we first train a network with data based on the considered mmWave channel characteristics, and then transfer the model to the Gaussian channel model. When retraining the model, the parameters of the first 10 layers are fixed and only the parameters of the remaining layers are trainable. From the result, we can see that the gap between the transferred deep-unfolding NN and the MSER-based hybrid GD algorithm with perfect CSI is much smaller compared to that of the transferred

TABLE III
SER versus the Number of Training Samples for SNR = 20 dB

Training samples	5000	10000	15000	20000	25000	30000	35000	40000
Benchmark CNN	66.03%	70.48%	72.55%	73.61%	74.13%	74.21%	74.36%	74.36%

Training samples	100	200	300	400	500	600	700	800
Deep-unfolding NN	92.83%	95.04%	96.21%	97.37%	97.85%	97.94%	98.02%	98.02%

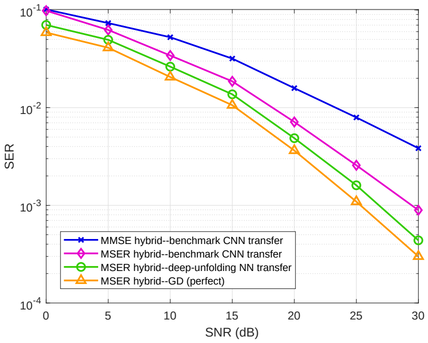

Fig. 11. SER performance of different schemes versus SNR under transfer conditions.

benchmark CNNs under the MSER and MMSE criteria. It shows that the proposed deep-unfolding NN has a better performance in transfer learning as it can significantly apply the knowledge gained from an older scenario and quickly adapt to a new one with less training time.

Table IV presents the computational complexity of the proposed schemes for different system configurations, as measured by the CPU running time of the training and testing stages. From the results, we can see that the CPU time of both training and testing stages increases with N_t and K. Clearly, the proposed deep-unfolding NN requires less training time compared to the CNN, which proves that the former structure can effectively accelerate convergence and reduce the training overhead. Since the deep-unfolding NN requires a much smaller number of trainable parameters and the gradients are computed in close-form in the BP, this makes it more efficient compared to the CNN which employs the platform “tensorflow”. Moreover, the gap in CPU time between the CNN and the deep-unfolding NN increases with N_t and K, where the training time is much larger than the testing time. The reason is that there are more complex computations in the training stage but only the FP process in the testing stage.

Table V aims to show the generalization ability of the proposed deep-unfolding NN. To this end, we first train a large-scale network with $N_t = 128$, $R_t = 32$, $N_r = 16$, $R_r = 4$, $K = 8$, and $D_k = 3$ and then implement it to test the performance of a MIMO system with smaller number of users K and transmit antennas N_t. From the results, we can see that the performance loss of applying this large-scale network to test the scenarios with smaller K and N_t is slight, which indicates the strong generalization ability of the proposed deep-unfolding NN.

TABLE V
Generalization Ability of the Proposed Deep-Unfolding NN

K / N_t	8	6	4	2
128	98.04%	98.45%	98.93%	99.37%
64	—	—	96.94%	97.52%

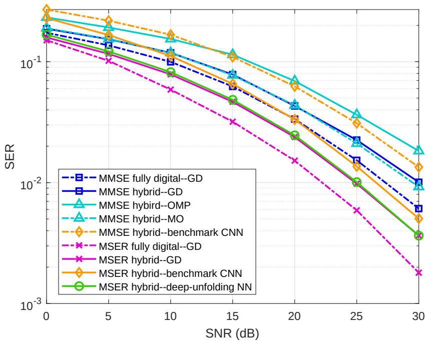

Fig. 12. SER performance of the different schemes versus SNR for 16-QAM signals.

D. SER Performance for 16-QAM Signals

Fig. 12 presents the SER performance of the different schemes versus SNR in the case of 16-QAM modulation. Similar to the QPSK modulation, the MSER-based algorithms achieve better performance than the MMSE-based algorithms. Meanwhile, the performance of the proposed deep-unfolding NN approaches that of the MSER-based hybrid GD beamforming algorithm although the performance gap between these two schemes decreases with SNR. In addition, we observe that the proposed deep-unfolding NN significantly outperforms the benchmark CNN at lower SNR. Table VI provides the SER values along with the number of the iterations/layers for SNR = 25 dB. It demonstrates the ability of the deep-unfolding NN to achieve nearly the same performance as the hybrid GD method while significantly reducing computational complexity. The above results show that the proposed deep-unfolding NN still works well for other higher-order modulation schemes.

Fig. 13 illustrates the transfer ability of the proposed shemes for 16-QAM signals. We can see that the gap between the deep-unfolding NN and the MSER-based hybrid GD algorithm is much smaller compared to that of the benchmark CNNs for changed channel characteristics. The results show that the proposed deep-unfolding NN has a better performance in

TABLE IV
CPU Running Time of the Proposed Algorithm

$(N_t,\ K,\ R_r)$	CPU time of training stage (min)		CPU time of testing stage (s)	
	Deep-unfolding NN	Benchmark CNN	Deep-unfolding NN	Benchmark CNN
(64, 2, 4)	5.35	8.73	0.01	0.01
(64, 3, 2)	12.46	21.58	0.01	0.1
(128, 4, 8)	84.45	130.79	0.07	0.11
(128, 5, 6)	95.73	200.46	0.10	0.13
(128, 6, 5)	109.48	242.35	0.14	0.17
(128, 7, 4)	129.63	295.43	0.17	0.22
(128, 8, 4)	162.86	360.15	0.24	0.28

TABLE VI
SER versus the Iterations/Layers for SNR = 25 dB

	Hybrid GD	Benchmark CNN	Deep-unfolding NN
	500 iterations	17 layers	15 layers
SNR = 25 dB	9.8×10^{-3}	1.36×10^{-2}	1.01×10^{-2}

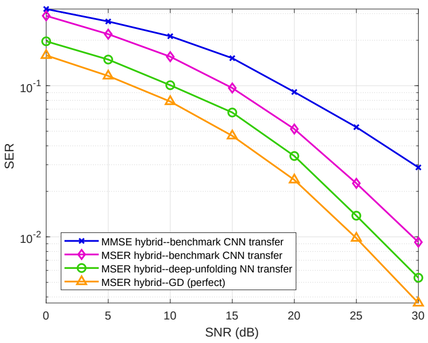

Fig. 13. SER performance of different schemes versus SNR under transfer conditions.

transfer learning compared to the benchmark CNNs.

VII. Conclusion

In this work, we addressed the problem of hybrid AD transceiver design for massive-MIMO systems based on the MSER criterion. Following the mathematical formulation of the problem as a constrained optimization, we first developed an MSER-based iterative GD algorithm to find stationary points. We then unfolded the iterative GD algorithm into a multi-layer structure and proposed a deep-unfolding NN, where a set of trainable parameters are introduced to reduce the computational complexity and improve the system performance. For the purpose of training, we obtained the relationship linking the gradients between two adjacent layers based on the GCR. The deep-unfolding NN was proposed for both QPSK and M-QAM signal constellations and its convergence was investigated through theoretical analysis; besides, we developed a black-box NN based on a recently proposed approach, for use as a benchmark. Simulation results showed that the proposed deep-unfolding NN provides better SER performance compared to the black-box CNN and approaches the performance of the MSER-based iterative GD algorithm with much lower complexity. Extensions of the deep-unfolding schemes developed in this paper to nonlinear modulation techniques or other types of beamforming algorithms are interesting avenues for future work.

Appendix A
GCR in Matrix Form

Based on [39], the general form of the optimization problem can be formulated as

$$\min_{\mathbf{X}} \ g(\mathbf{X};\mathbf{Z}) \quad \text{s.t. } \mathbf{X} \in \mathcal{S}, \tag{49}$$

where $g : \mathbb{C}^{m\times n} \to \mathbb{R}$ denotes a continuous objective function, $\mathbf{X} \in \mathbb{C}^{m\times n}$ denotes the variable, $\mathcal{S}$ is the feasible region, and $\mathbf{Z}$ is the random parameter in the problem. To solve the problem, an iterative algorithm is developed as

$$\mathbf{X}^{t+1} = G_t(\mathbf{X}^t;\mathbf{Z}), \tag{50}$$

where function $G_t(\cdot)$ maps the variable $\mathbf{X}^t$ to the variable $\mathbf{X}^{t+1}$ in the t-th iteration. To reduce the computational complexity, trainable parameter $\boldsymbol{\lambda}$ is introduced. Since $\mathbf{Z}$ is a random variable, we need to take the expectation of $\mathbf{Z}$ and the problem is transformed into

$$\min_{\mathbf{X}} \ \mathbb{E}_{\mathbf{Z}}\{g(\mathbf{X};\boldsymbol{\lambda},\mathbf{Z})\} \quad \text{s.t. } \mathbf{X} \in \mathcal{S}. \tag{51}$$

A deep-unfolding NN can be developed for the above problem as follows,

$$\mathbf{X}^{l+1} = \mathcal{G}_l(\mathbf{X}^l;\boldsymbol{\lambda}^l,\mathbf{Z}), \tag{52}$$

where $\mathcal{G}_l(\cdot)$ denotes the update function of the NN in the l-th layer, $\mathbf{X}^l$ and $\mathbf{X}^{l+1}$ denote the input and output of the l-th layer, respectively, $\mathbf{Z}$ represents the fixed parameter or input of the NN, and $\boldsymbol{\lambda}^l$ is the trainable parameter in the l-th layer. For such a NN, in order to train the parameter $\boldsymbol{\lambda}^l$, we need to derive the relationship between the gradients of adjacent layers and then calculate the gradient w.r.t. $\boldsymbol{\lambda}^l$. To apply the GCR which leads to an expression of the following type

$$\mathrm{Tr}\left(\mathbf{G}^{l+1}\, d\mathbf{X}^{l+1}\right) = \mathrm{Tr}\left(\mathbf{G}^{l+1} \circ \mathcal{J}(\mathbf{X}^l;\boldsymbol{\lambda}^l,\mathbf{Z})\, d\mathbf{X}^l\right), \tag{53}$$

where $\mathbf{G}^{l+1}$ and $\mathbf{G}^l$ are the gradients w.r.t. $\mathbf{X}^{l+1}$ and $\mathbf{X}^l$, respectively, $\mathcal{J}(\mathbf{X}^l;\boldsymbol{\lambda}^l,\mathbf{Z})$ denotes some matrix functions of $\mathbf{X}^l$, $\boldsymbol{\lambda}^l$, and $\mathbf{Z}$, which are related to the update function $\mathcal{G}_l(\cdot)$. Then, we obtain the desired gradient relationship as

$$\mathbf{G}^l = \mathbf{G}^{l+1} \circ \mathcal{J}(\mathbf{X}^l;\boldsymbol{\lambda}^l,\mathbf{Z}). \tag{54}$$

In this work, when $\mathbf{X}$ denotes $\mathbf{P}_k$, we find that $\mathcal{J}(\cdot) = \mathcal{J}_1(\cdot) \circ \mathcal{J}_2(\cdot)$, where $\mathcal{J}_1(\cdot) = \mathbf{E}^{\perp} - b_k^H B(\boldsymbol{\alpha}^l_{P_k})^T(\mathbf{F}^l)^H\mathbf{H}_k^H (\mathbf{U}_k^l)^H\mathbf{W}_k^l$ and $\mathcal{J}_2(\cdot) = \mathbf{E}$, where $\mathbf{E}$ is given in Section IV-B.

Appendix B
Gradients of Trainable Parameters

According to the update rules in (27)–(30), the gradients w.r.t. the introduced trainable parameters in each layer are acquired as follows, where the index of iteration t is omitted for simplicity.

$$\begin{aligned}
&\nabla_{\boldsymbol{\alpha}_{P_k}} \tilde{\mathcal{P}}_e^l = -(\mathbf{G}_{P_k}^{l+1})^H \circ (\nabla_{\mathbf{P}_k^H} \tilde{\mathcal{P}}_e^l)^H, \nabla_{\mathbf{O}_{P_k}} \tilde{\mathcal{P}}_e^l = (\mathbf{G}_{P_k}^{l+1})^H, \\
&\nabla_{\boldsymbol{\alpha}_{\theta_F}} \tilde{\mathcal{P}}_e^l = -(\mathbf{G}_F^{l+1})^H \circ j\mathbf{F}^* \circ (\nabla_{\mathbf{F}} \tilde{\mathcal{P}}_e^l)^H + (\mathbf{G}_{F^*}^{l+1})^H \circ j\mathbf{F} \circ (\nabla_{\mathbf{F}^*} \tilde{\mathcal{P}}_e^l)^H, \\
&\nabla_{\mathbf{O}_{\theta_F}} \tilde{\mathcal{P}}_e^l = (\mathbf{G}_F^{l+1})^H \circ j\mathbf{F}^* - (\mathbf{G}_{F^*}^{l+1})^H \circ j\mathbf{F}, \\
&\nabla_{\rho_{P_k}} \tilde{\mathcal{P}}_e^l = \frac{1}{J\sqrt{2\pi}(\rho_{P_k}^l)^2} \sum_{j=1}^{J} \sum_{i=1}^{D_k} e^{-\frac{|\bar{b}_{i,k}^{q_j}|^2}{2(\rho_{P_k}^l)^2}} \Big(1 - \frac{|\bar{b}_{i,k}^{q_j}|^2}{2(\rho_{P_k}^l)^2}\Big) \mathbf{G}_{P_k}^{l+1} \\
&\qquad \circ (\boldsymbol{\alpha}_{P_k}^l)^T \mathbf{F}^H \mathbf{H}_k^H \mathbf{U}_k^H \mathbf{w}_{i,k}.
\end{aligned} \tag{55}$$

References

[1] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 101-108, Jun. 2011.

[2] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335-349, May 2013.

[3] T. Kim, J. Park, J.-Y. Seol, S. Jeong, J. Cho, and W. Roh, "Tens of Gbps support with mmWave beamforming systems for next generation communications," in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2013, pp. 3685-3690.

[4] X. Zhang, A. F. Molisch, and S.-Y. Kung, "Variable-phase-shift-based RF-baseband codesign for MIMO antenna selection," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4091-4103, Oct. 2005.

[5] P. Sudarshan, N. B. Mehta, A. F. Molisch, and J. Zhang, "Channel statistics-based RF pre-processing with antenna selection," *IEEE Trans. Wireless Commun.*, vol. 5, no. 12, pp. 3501-3511, Dec. 2006.

[6] 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.

[7] 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.

[8] M. Kim and Y. H. Lee, "MSE-based hybrid RF/baseband processing for millimeter-wave communication systems in MIMO interference channels," *IEEE Trans. Veh. Technol.*, vol. 64, no. 6, pp. 2714-2720, Jun. 2015.

[9] D. H. N. Nguyen, L. B. Le, T. Le-Ngoc, and R. W. Heath, "Hybrid MMSE precoding and combining designs for mmWave multiuser systems," *IEEE Access*, vol. 5, pp. 19167-19181, 2017.

[10] D. H. N. Nguyen, L. B. Le, and T. Le-Ngoc, "Hybrid MMSE precoding for mmWave multiuser MIMO systems," in *IEEE Int. Conf. on Communications (ICC)*, May 2016, pp. 1-6.

[11] X. Yu, J. Shen, J. Zhang, and K. B. Letaief, "Alternating minimization algorithms for hybrid precoding in millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 485-500, Apr. 2016.

[12] S. He, J. Wang, Y. Huang, B. Ottersten, and W. Hong, "Codebook-based hybrid precoding for millimeter wave multiuser systems," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5289-5304, Oct. 2017.

[13] J. Cong, X. Li, and Y. Zhu, "Hybrid precoding for multi-user mmWave systems based on MMSE criterion," in *23rd Asia-Pacific Conference on Communications (APCC)*, Dec. 2017, pp. 1-5.

[14] T. Lin, J. Cong, Y. Zhu, J. Zhang, and K. Ben Letaief, "Hybrid beamforming for millimeter wave systems using the MMSE criterion," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3693-3708, May 2019.

[15] Y. Cai and R. C. de Lamare, "Adaptive linear minimum BER reduced-rank interference suppression algorithms based on joint and iterative optimization of filters," *IEEE Commun. Lett.*, vol. 17, no. 4, pp. 633-636, Apr. 2013.

[16] Y. Cai, R. C. de Lamare, B. Champagne, B. Qin, and M. Zhao, "Adaptive reduced-rank receive processing based on minimum symbol-error-rate criterion for large-scale multiple-antenna systems," *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 4185-4201, Nov. 2015.

[17] M. Shao, Q. Li, and W. Ma, "One-bit massive MIMO precoding via a minimum symbol-error probability design," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 3579-3583.

[18] K. L. Law and C. Masouros, "Symbol error rate minimization precoding for interference exploitation," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5718-5731, Nov. 2018.

[19] H. D. Nguyen, R. Zhang, and S. Sun, "Improper signaling for symbol error rate minimization in K-user interference channel," *IEEE Trans. Commun.*, vol. 63, no. 3, pp. 857-869, Mar. 2015.

[20] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563-575, Dec. 2017.

[21] S. Dörner, S. Cammerer, J. Hoydis, and S. T. Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132-143, Feb. 2018.

[22] 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, 2020.

[23] D. Neumann, T. Wiese, and W. Utschick, "Learning the MMSE channel estimator," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2905-2917, Jun. 2018.

[24] 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.

[25] 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.

[26] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1760-1776, Mar. 2020.

[27] A. M. Elbir and K. V. Mishra, "Deep learning design for joint antenna selection and hybrid beamforming in massive MIMO," in *IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, Jul. 2019, pp. 1585-1586.

[28] X. Li and A. Alkhateeb, "Deep learning for direct hybrid precoding in millimeter wave massive MIMO systems," in *53rd Asilomar Conference on Signals, Systems, and Computers*, Nov. 2019, pp. 800-805.

[29] C. Sidharth, S. M. Hiremath, and S. K. Patra, "Deep learning based hybrid precoding for mmWave massive MIMO system using comcepNet," in *International Conference on Communication and Signal Processing (ICCSP)*, Jul. 2020, pp. 1317-1321.

[30] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.

[31] V. Corlay, J. J. Boutros, P. Ciblat, and L. Brunel, "Multilevel MIMO detection with deep learning," in *52nd Asilomar Conference on Signals, Systems, and Computers*, Oct. 2018, pp. 1805-1809.

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

[33] W. Xu, Z. Wu, Y. Ueng, X. You, and C. Zhang, "Improved polar decoder based on deep learning," in *IEEE International Workshop on Signal Processing Systems (SiPS)*, Oct. 2017, pp. 1-6.

[34] W. Xu, X. You, C. Zhang, and Y. Beéry, "Polar decoding on sparse graphs with deep learning," in *52nd Asilomar Conference on Signals, Systems, and Computers*, Oct. 2018, pp. 599-603.

[35] D. Ito, S. Takabe, and T. Wadayama, "Trainable ISTA for sparse signal recovery," *IEEE Trans. Signal Process.*, vol. 67, no. 12, pp. 3113-3125, Jun. 2019.

[36] A. Balatsoukas-Stimming, O. Castañeda, S. Jacobsson, G. Durisi, and C. Studer, "Neural-network optimized 1-bit precoding for massive MU-MIMO," in *IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Jul. 2019, pp. 1-5.

[37] Y. He, H. He, C. -K. Wen, and S. Jin, "Model-driven deep learning for massive multiuser MIMO constant envelope precoding," *IEEE Wireless Commun. Lett.*, vol. 9, no. 11, pp. 1835-1839, Nov. 2020.

[38] H. He, M. Zhang, S. Jin, C. -K. Wen, and G. Y. Li, "Model-driven deep learning for massive MU-MIMO with finite-alphabet precoding," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2216-2220, Oct. 2020.

[39] 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.

[40] S. Chen, A. K. Samingan, B. Mulgrew, and L. Hanzo, "Adaptive minimum-BER linear multiuser detection for DS-CDMA signals in multipath channels," *IEEE Trans. Signal Process.*, vol. 49, no. 6, pp. 1240-1247, Jun. 2001.

[41] S. Chen, A. Livingstone, H. -Q. Du, and L. Hanzo, "Adaptive minimum symbol error rate beamforming assisted detection for quadrature amplitude modulation," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1140-1145, Apr. 2008.

[42] X. Zhang, *Matrix Analysis and Applications*, U.K., Cambridge Univ. Press, 2017.