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 a MSER-based gradient descent (GD) iterative algorithm to find the related stationary points. We then propose a deep-unfolding neural network (NN), in which the iterative GD algorithm is unfolded into a multi-layer structure wherein a set of trainable parameters are introduced for accelerating the convergence and enhancing the overall system performance. To implement the training stage, the relationship between the gradients of adjacent layers is derived based on the generalized chain rule (GCR). The deep-unfolding NN is developed for both quadrature 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, hence mitigating the spectrum shortage [1]–[3]. This is achieved by forming directional beams. 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 the minimization of 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, a number of powerful beamformers have been designed based on the minimum SER (MSER) criterion [15]–[19]. The authors of [15] and [16] developed a MSER-based adaptive reduced-rank receive beamformer for enhancing the performance. In [17], a single-bit direct MSER-optimization based precoder was 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 a 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 improved 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. To overcome these issues, a model-driven network, which is referred to as deep-unfolding, has been proposed in [30]. This method generally unfolds iterative algorithms into multi-layer NN structures and introduces a set of trainable parameters to accelerate convergence and to 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], the authors developed a deep-unfolding method inspired by the normalized min-sum algorithm for the decoding of polar codes. In [34], a scheme based on deep-unfolding was put forward for sparse signal recovery. In [35], 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

The MSER-based hybrid beamforming design problem in massive MIMO systems is very challenging due to the highly nonconvex objective function and unit-modulus constraints. While the GD algorithm is a common technique that can be adopted for finding the stationary points, it requires a long time to converge and a large number of iterations. Since the deep-unfolding method makes full use of the structure of the iterative algorithm and only replaces some complex operations with trainable parameters, it can significantly reduce the computational complexity with only a slight performance loss. 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. Hence we first mathematically formulate the problem of hybrid AD transceiver design under the direct MSER criterion in a hardware-efficient massive MIMO system. We then develop a MSER-based GD iterative algorithm for finding the stationary points. Specifically, we propose a deep-unfolding NN, in which the iterative GD algorithm is unfolded into a multi-layer structure and then a set of trainable parameters are introduced into the forward propagation (FP) for accelerating the convergence and enhancing 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 herein 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 GD iterative algorithm at a much reduced complexity. The main contributions of this work are summarized as follows:

- We formulate the joint hybrid AD transceiver design problem based on the MSER criterion in a massive MIMO system context, which is very challenging to tackle due to the highly nonconvex objective function and unit-modulus constraints imposed on the entries of the analog beamforming matrix. Then an unconstrained MSER-based GD iterative algorithm is developed for finding the stationary points through the application of kernel density estimation and a parametric representation of the unit-modulus matrix entries.
- We propose a deep-unfolding NN based on the MSER-based GD iterative 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.
- We provide a theoretical analysis for the convergence of the proposed deep-unfolding NNs and also analyze their transfer ability, computational complexity and generalization capability.
- Through numerical simulation, we show that the proposed deep-unfolding NN substantially outperforms the conventional black-box method and approaches the performance of the MSER-based GD 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 a 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 extension of the proposed algorithm to the QAM modulation are provided. The simulation results are presented in Section VI and conclusions are drawn in Section VII.
Fig. 1: Hardware-efficient massive MIMO system for downlink transmission.

Notations: Scalars, vectors and matrices are respectively denoted by lower case, boldface lower case and boldface upper case letters. I represents an identity matrix and 0 represents an all-zero matrix. For a matrix $A$, $A^T$, $A^*$, $A^H$, $A^\perp$, $\|A\|$ and $[A]_{m,n}$ 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 $a$, $\|a\|$ represents its Euclidean norm. $E\{\cdot\}$ denotes the statistical expectation. $\mathbb{R}\{\cdot\}$ denotes the real (imaginary) part of a variable. The operator vec$(\cdot)$ stacks the columns of a matrix in one long column vector. $\{\cdot\}$ denotes the absolute value of a complex scalar. $\mathcal{C}^m\times n(\mathbb{R}^m\times n)$ denotes the space of $m \times n$ complex (real) matrices. The operator $\otimes$ 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 $k \in K = \{1, \ldots, K\}$ is equipped with $N_r,k$ receive antennas and $R_r (R_r \ll N_r,k)$ RF chains, where $N_t \gg \sum_{k=1}^K N_r,k$ and $R_t \geq \sum_{k=1}^K R_r,k$. The BS transmits a symbol vector $s \triangleq [b_1^T, b_2^T, \ldots, b_K^T]^T \in \mathbb{C}^{D_r \times 1}$ to the users, where the transmit symbols are independent and identically distributed (i.i.d.). The symbol vector of user $k \in K$, $b_k \triangleq [b_{1,k}, \ldots, b_{D_k,k}]^T \in \mathbb{C}^{D_k \times 1}$, has zero-mean and covariance matrix $E[b_k b_k^H] = I$, where $D_k$ ($D_k \leq R_r,k$) denotes the number of data streams 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 symbol vector $b_k$ are uniformly drawn from $\{\pm 1\}$. For the $M$-QAM modulation, each entry of $b_{k,m}$ is uniformly drawn from $\{F_{m,j} + jF_{m,j} : 1 \leq m, n \leq \sqrt{M}\}$, where integer $M$ is a perfect square and we define $F_{m} = 2n - \sqrt{M} - 1$. The transmitted symbol vector $s$ is first processed by a digital transmit beamforming matrix $V \triangleq [P_1, \ldots, P_K] \in \mathbb{C}^{R_t \times D}$, where $P_k \in \mathbb{C}^{R_t \times D_k}$ 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 $F \in \mathbb{C}^{N_t \times R_t}$ implemented by means of phase shifters, i.e., $[F]_{m,n} = 1$. The transmit power constraint at the BS is given by

$$E\{\|FVs\|^2\} = \|FS\|^2 = P_T,$$

where $P_T$ denotes the total transmit power budget.

The signal received at user $k$ is given by

$$y_k = H_k F V s + n_k = H_k F \sum_{k=1}^K P_k b_k + n_k,$$

where $H_k \in \mathbb{C}^{N_r,k \times N_t}$ 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

$$H_k = \sqrt{\frac{N_t N_r,k}{N_C N_R}} \sum_{i=1}^{N_C} \sum_{j=1}^{N_R} \alpha_{i,j} a_r(\theta_{i,j}) a_t(\theta_{i,j})^H,$$

where $\alpha_{i,j}$ denotes the complex propagation gain of the $j$-th ray in the $i$-th cluster, while $a_r(\theta_{i,j}) \triangleq \frac{1}{\sqrt{N_r}} \left[1, e^{j\pi \sin\theta_{i,j}}, \ldots, e^{j\pi(N_r-1)\sin\theta_{i,j}}\right]^T$ and $a_t(\theta_{i,j}) \triangleq \frac{1}{\sqrt{N_t}} \left[1, e^{j\pi \sin\theta_{i,j}}, \ldots, e^{j\pi(N_t-1)\sin\theta_{i,j}}\right]^T$ denote the corresponding normalized response vectors of the transmit and receive antenna arrays, with $\theta_{i,j}$ and $\theta_{i,j}$ denoting the angles of arrival and departure, respectively. The term $n_k \in \mathbb{C}^{N_r,k \times 1}$ in (2) represents the additive noise at user $k$, modeled as a complex circularly symmetric Gaussian vector with zero-mean and covariance matrix $E[n_k n_k^H] = \sigma_n^2 I$.

Similar to the BS, the hybrid AD receiver at user $k$ is comprised of an analog receive beamforming matrix $U_k \in \mathbb{C}^{R_r,k \times N_r}$, followed by a digital receive beamforming matrix $W_k \triangleq [w_{1,k}, \ldots, w_{D_k,k}] \in \mathbb{C}^{R_r,k \times D_k}$, where $w_{i,k} \in \mathbb{C}^{R_r,k \times 1}$ and $i \in D_k \triangleq \{1, \ldots, D_k\}$. Accordingly, the output signal vector $b_k \triangleq [b_{1,k}, \ldots, b_{D_k,k}]^T \in \mathbb{C}^{D_k \times 1}$ of the hybrid receiver at user $k$ can be expressed as

$$\tilde{b}_k = W_k^H U_k y_k = W_k^H U_k H_k F \sum_{k=1}^K P_k b_k + n_k,$$

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

Finally, the estimate of the transmitted symbol vector $b_k$ at the output of receiver $k$ is

$$\hat{b}_k = Q\{\tilde{b}_k\},$$

where $Q\{\cdot\}$ 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 $\{P_k, W_k, U_k, F\}$ 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 TRANSCIEVER DESIGN

In this section, we focus 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 a MSER-based GD iterative algorithm for finding the stationary points, which will serve as the basis in the elaboration of our proposed deep unfolding NN.

A. MSER Criterion

The decisions for the detection of the real and imaginary parts of element $i \in D_k$ in the $k$-th user’s symbol vector $b_k = [b_{1,k}, \ldots, b_{D_k,k}]^T$ are made as

$$\begin{align*}
\Re\{\hat{b}_{i,k}\} &= \begin{cases} +1, & \text{if } \Re\{\hat{b}_{i,k}\} \geq 0 \\ -1, & \text{if } \Re\{\hat{b}_{i,k}\} < 0 \end{cases}, \\
\Im\{\hat{b}_{i,k}\} &= \begin{cases} +1, & \text{if } \Im\{\hat{b}_{i,k}\} \geq 0 \\ -1, & \text{if } \Im\{\hat{b}_{i,k}\} < 0 \end{cases}
\end{align*} \quad (6)$$

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

$$\mathcal{X} \triangleq \{s^1, s^2, \ldots, s^{N_b}\}, \quad (7)$$

where $s^q = [(b_{1,k})^T, (b_{2,k})^T, \ldots, (b_{K,k})^T]^T$, $q \in N_b = \{1, \ldots, N_b\}$, and we assume an equiprobable model for the $N_b$ possible transmit vectors $s^q$. The noise-free component for the $i$-th element of the hybrid receiver’s output at user $k$ is from the set

$$\mathcal{Y}_{i,k}\triangleq \{\bar{b}_{i,k} = w_{i,k}^H U_k H_k F s^q, q \in N_b\}. \quad (8)$$

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

$$f(x|b_{i,k}) = \frac{1}{N_b \sqrt{2\pi w_{i,k} H_k F V F^H H_k U_k^H U_k w_{i,k} \sigma_n}} \sum_{q=1}^{N_b} e^{-\frac{|x - w_{i,k}^H U_k H_k F V F^H H_k U_k^H U_k w_{i,k} \sigma_n|^2}{2 w_{i,k}^H U_k H_k F V F^H H_k U_k^H U_k w_{i,k} \sigma_n}}, \quad (9)$$

where $\bar{b}_{i,k} \in \mathcal{Y}$. Due to the large value of $N_b$, it will not be possible to consider the sum over all 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 $j \in \{1, \ldots, J\}$, and employ a constant kernel width $\varrho$ to replace the term $\sqrt{w_{i,k}^H U_k H_k F V F^H H_k U_k^H U_k w_{i,k} \sigma_n}$ appearing in (9) for complexity reduction. The parameter $\varrho$ 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

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

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

$$P_e(b_{i,k}) = P_e^R(b_{i,k}) + P_e^I(b_{i,k}) - P_e^R(b_{i,k})P_e^I(b_{i,k}), \quad (11)$$

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

$$P_e^R(b_{i,k}) = \frac{1}{J \sqrt{2\varrho \theta}} \sum_{j=1}^{J} e^{-\frac{|(\bar{b}_{i,k})_j|^2}{2 \varrho \theta}} e^{-s^2} ds, \quad (12)$$

$$P_e^I(b_{i,k}) = \frac{1}{J \sqrt{2\varrho \theta}} \sum_{j=1}^{J} e^{-\frac{|(\bar{b}_{i,k})_j|^2}{2 \varrho \theta}} e^{-s^2} ds. \quad (13)$$

According to the discussion in [11], we can drop the product term $P_e^R(b_{i,k})P_e^I(b_{i,k})$ and focus on the upper bound $P_e(b_{i,k}) \triangleq P_e^R(b_{i,k}) + P_e^I(b_{i,k})$ for reducing the computational complexity. Indeed, for small values of SER, $P_e(b_{i,k})$ is very close to the true SER $P_e(b_{i,k})$, 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 $\{P_k, W_k, U_k, F\}$, where index $k$ runs through $K$. Accordingly, the problem is formulated as

$$\begin{align*}
\min_{\{P_k, W_k, U_k, F\}} \quad & \sum_{k=1}^{K} \sum_{i=1}^{D_k} P_e(b_{i,k}) \leq \sum_{k=1}^{K} \sum_{i=1}^{D_k} \left( P_e^R(b_{i,k}) + P_e^I(b_{i,k}) \right) \\
\text{s.t.} & \quad \|FV\|^2 = P_T, \quad (14a) \\
& \quad |F|_{m,n} = 1, \quad |U_k|_{m,n} = 1, \quad \forall k, m, n, \quad (14b)
\end{align*}$$

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 unimodulus constraints in (14c), we define the analog beamforming phase matrices $\theta_U$ and $\theta_F$. Consequently, $U_k$ and $F$ can be obtained by $U_k = \exp(j\theta_U)$ and $F = \exp(j\theta_F)$, 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 $V = [P_1, \ldots, P_K]$ at the end of each iteration of the proposed MSER-based GD algorithm. Therefore, let us consider the following unconstrained problem for simplicity:

$$\begin{align*}
\min_{\{P_k, W_k, U_k, \theta_U, \theta_F\}} \quad & \sum_{k=1}^{K} \sum_{i=1}^{D_k} \tilde{P}_e(b_{i,k}) \leq \sum_{k=1}^{K} \sum_{i=1}^{D_k} (P_e^R(b_{i,k}) + P_e^I(b_{i,k})). \\
\end{align*} \quad (15)$$

Based on the objective function of MSER, the gradient of $\tilde{P}_e(b_{i,k})$ with respect to (w.r.t.) the hybrid beamforming matrices is given by $\nabla \tilde{P}_e(b_{i,k}) \triangleq \nabla P_e^R(b_{i,k}) + \nabla P_e^I(b_{i,k})$, where $\nabla P_e^R(b_{i,k})$ and $\nabla P_e^I(b_{i,k})$ 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 $\{P_k, W_k, U_k, F\}$. Specifically, computing the gradient of (12), as in (12), we obtain

$$\begin{align*}
\nabla P_e^R &= -\frac{1}{J \sqrt{2\varrho \theta}} \sum_{j=1}^{J} e^{-\frac{|(\bar{b}_{i,k})_j|^2}{2 \varrho \theta}} \Re\{b_{i,k}\}, \quad (16) \\
\nabla w_{i,k}^H U_k H_k F V s^q &= \frac{1}{J \sqrt{2\varrho \theta}} \sum_{j=1}^{J} e^{-\frac{|(\bar{b}_{i,k})_j|^2}{2 \varrho \theta}} \Re\{b_{i,k}\}, \quad (17)
\end{align*}$$
of each GD iteration, the overall digital transmit beamforming based on the MSER criterion. The GD update equations, which are obtained by substituting the gradients (16)–(21) in the following expressions

The gradients w.r.t. the analog beamforming phase matrices \( \theta_{uk} \) and \( \theta_{F} \) can be obtained as

\[
\nabla \theta_{uk} P_{e}^{R} = R \{ \nabla_{u_{k}} P_{e}^{R} \circ j u_{k} - \nabla_{U_{k}} P_{e}^{R} \circ j U_{k} \} \\
= -\nabla_{u_{k}} P_{e}^{R} \circ U_{k}^{l} - \nabla_{U_{k}} P_{e}^{R} \circ U_{k}^{R} \\
+ \nabla_{u_{k}} P_{e}^{R} \circ (U_{k}^{l})^{*} + \nabla_{U_{k}} P_{e}^{R} \circ (U_{k}^{R})^{*},
\]

\[
\nabla \theta_{F} P_{e}^{R} = R \{ \nabla_{F} P_{e}^{R} \circ j F - \nabla_{F} P_{e}^{R} \circ j F^{*} \} \\
= -\nabla_{F} P_{e}^{R} \circ F^{l} - \nabla_{F} P_{e}^{R} \circ F^{R} \\
+ \nabla_{F} P_{e}^{R} \circ (F^{l})^{*} + \nabla_{F} P_{e}^{R} \circ (F^{R})^{*}.
\]

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

\[
P_{k+1} = P_{k} - \mu_{P} \nabla P_{e}^{R},
\]

\[
W_{k+1} = W_{k} - \mu_{W} \nabla W_{k} P_{e}^{R},
\]

\[
\theta_{uk+1} = \theta_{uk} - \mu_{\theta_{uk}} \nabla \theta_{uk} P_{e}^{R},
\]

\[
\theta_{F+1} = \theta_{F} - \mu_{\theta_{F}} \nabla \theta_{F} P_{e}^{R},
\]

where \( \{ \mu_{P}, \mu_{W}, \mu_{\theta_{uk}}, \mu_{\theta_{F}} \} \) 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 \( V \) needs to be scaled as follows:

\[
V \leftarrow \frac{\sqrt{P} F}{\| F V \|} V.
\]

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 \( \{ \mu_{P}, \mu_{W}, \mu_{\theta_{uk}}, \mu_{\theta_{F}} \} \). Set the iteration index to \( t = 0 \).
2: Initialize \( P_{k} \) to satisfy the power constraint. Initialize \( \{ W_{k}, \theta_{uk}, \theta_{F} \} \).
3: repeat
4: Update \( P_{k+1}^{t+1} \) with fixed \( \{ W_{k}, \theta_{uk}, \theta_{F} \} \), \( \forall k \in K \), according to (22).
5: Update \( W_{k+1}^{t+1} \) with fixed \( \{ P_{k}, \theta_{uk}, \theta_{F} \} \), \( \forall k \in K \), according to (23).
6: Update \( \theta_{uk+1}^{t+1} \) with fixed \( \{ P_{k}, W_{k}, \theta_{F}^{t+1} \} \), \( \forall k \in K \), according to (24).
7: Update \( \theta_{F+1}^{t+1} \) with fixed \( \{ P_{k}, W_{k}, \theta_{uk}^{t+1} \} \), according to (25).
8: Scale \( P_{k+1}^{t+1} \) 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 \( t > T \).

IV. PROPOSED DEEP-UNFOLDING NN FOR HYBRID AD TRANSCIEVER 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 proposed NN structure is inspired by the MSER-based GD algorithm, where the iterative algorithm is unfolded into a multi-layer structure and a number of adjustable step size and bias 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. In the testing stage, we perform the FP process based on the trained parameters for computing the AD beamforming matrices. 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 iteration of the latter algorithm, the step sizes used for updating the hybrid AD beamforming matrices, i.e., \( \{ \mu_{P}, \mu_{W}, \mu_{\theta_{uk}}, \mu_{\theta_{F}} \} \) greatly affect the SER performance; furthermore, are usually determined based on experience and experiments. Therefore, in layer \( t \in \{0, \ldots, L - 1 \} \) of the proposed NN, where \( L \) denotes the number of layers, we introduce the trainable matrix parameters \( \{ \alpha_{P_{t}}, \alpha_{W_{t}}, \alpha_{\theta_{uk}}, \alpha_{\theta_{F}} \} \) as the learning rates to replace the step sizes of the MSER-based GD algorithm.

Fig. 2: Structure of the proposed MSER-based iterative GD algorithm.
Recall that the term \( \sqrt{w_k^H U_k H_k F V V^H F^H H_k^H U_k^H w_k} \) in (25) is set as a constant kernel width \( \varrho \) in the MSER-based GD algorithm, which may cause performance loss. Hence, we replace \( \varrho \) in (16)–(19) by the set of trainable parameters \( \{\rho_{p_k}, \rho_{W_k}, \rho_{U_k}, \rho_{F}\} \). Moreover, to increase the degrees of freedom for the parameters, we introduce the trainable offset matrix parameters \( \{\theta_{O_{p_k}}, \theta_{O_{W_k}}, \theta_{O_{U_k}}, \theta_{O_{F}}\} \) for the computation of the beamforming matrices. The update expressions for the proposed deep-unfolding NN are given by

\[
P_{k+1}^l = P_k - \alpha_{P_k}^l \circ \nabla P_{k}^{l+1} + \Omega_{P_k}, \quad (27)
\]

\[
W_{k+1}^l = W_k - \alpha_{W_k}^l \circ \nabla W_{k}^{l+1} + \Omega_{W_k}, \quad (28)
\]

\[
\theta_{U_k}^{l+1} = \theta_{U_k}^l - \alpha_{\theta_{U_k}}^l \circ \nabla \theta_{U_k}^{l+1} + \Omega_{\theta_{U_k}}, \quad (29)
\]

\[
\theta_{F}^{l+1} = \theta_{F}^l - \alpha_{\theta_{F}}^l \circ \nabla \theta_{F}^{l+1} + \Omega_{\theta_{F}}, \quad (30)
\]

where \( \{\alpha_{P_k}^l, \alpha_{W_k}^l\} \subseteq \mathbb{R}_{+}^{R \times D_k}, \{\alpha_{U_k}^l, \alpha_{F}^l\} \subseteq \mathbb{R}_{+}^{R \times R_k \times D_k}, \{\theta_{O_{p_k}}, \theta_{O_{W_k}}, \theta_{O_{U_k}}, \theta_{O_{F}}\} \subseteq \mathbb{R}_{+}^{R \times R_k \times R_k \times D_k}, \)

and \( \{\nabla P_{k}^{l+1}, \nabla W_{k}^{l+1}, \nabla \theta_{U_k}^{l+1}, \nabla \theta_{F}^{l+1}\} \) 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 \( P(\varrho), W(\varrho), U(\varrho), \text{ and } F(\varrho) \) represent (27)–(30), respectively. Specifically, in layer \( l \in \{0, \ldots, L - 1\} \), we first update the digital beamforming matrices \( P_k \) and \( W_k \) successively, followed by the analog beamforming phase matrices \( \theta_{U_k} \) and \( \theta_{F} \). For each of these one of the updates, the enlarged diagram in the red dotted rectangle (bottom) illustrates how the GD updates make use of the additional parameters \( \alpha_{\varrho_{k}}, \rho_X, \text{ and } O_X \), 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 \( \theta_{F} \) is the last updated parameter matrix, the iteration rule of \( \theta_{F} \) adopts the original formula (25) without introducing additional parameters.

Since the channel matrices \( H_k \) are random by nature, the final loss function incorporates an expectation operation \( \mathbb{E}_H \) over the ensemble of channel matrices. Hence, the loss function \( L \) in the top-right corner of Fig. 3 is modified as

\[
L = \sum_{k=1}^{K} \mathbb{E}_H \left\{ \frac{1}{J \sqrt{2\pi}} \sum_{i=1}^{J} \int_{-\infty}^{\infty} e^{-\frac{(s \lambda_{i,k}^l)^2}{2\sigma_{k}^2}} \, ds \right\}.
\]

As mentioned in Section III, \( V = [P_1, \ldots, P_K] \) 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 \( \{G_{p_k}^l, G_{W_k}^l, G_{U_k}^l, G_{F}^l, G_{F}^l\} \) 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. \( F^l \) and \( (F^*)^l \) in the last layer as

\[
G_{F}^L = \langle \nabla F \hat{P}_F^L \rangle^H, \quad G_{F}^L = \langle \nabla F \hat{P}_F^L \rangle^H.
\]

By differentiating on both sides of (20) and (21), it is readily seen that \( G_{U_k}^l \) and \( G_{\theta}^l \) can be computed based on \( \{G_{U_k}^l, G_{U_k}^l\} \) and \( \{G_{F}^l, G_{F}^l\} \), respectively, as

\[
G_{U_k}^l = G_{U_k}^l \circ jU_k^T - G_{U_k}^l \circ jU_k^T, \quad (32)
\]

\[
G_{\theta}^l = G_{\theta}^l \circ jF^T - G_{\theta}^l \circ jF^T, \quad (33)
\]

\[
G_{\theta}^l = G_{\theta}^l \circ jF^T - G_{\theta}^l \circ jF^T, \quad (34)
\]

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 \( G_{P}^l \) and \( G_{P}^l \), 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 \( j \), and the index of iteration \( l \). Based on (27) and the GCR shown in Appendix A we have

\[
\text{Tr} \left\{ G_{P}^{l+1} P_{k}^{l+1} \right\} = \sum_{j=1}^{J} \text{Tr} \left\{ G_{P}^{l+1} P_{k}^{l+1} - B_{k} G_{P}^{l+1} \right\}
\]

\[
(\alpha_{p}^{l+1} T) (F^l) H_{k} H_{k} (U_{k})^H W_{k}^l \circ (b_k (W_{k})^H U_{k} H_{k} F^l)
\]

\[
dP_{k} + D_{H_{k}} (U_{k})^H dW_{k} + C (W_{k})^H U_{k} H_{k} dF + D (W_{k})^H dU_{k} + H_{k}^H U_{k} C^H dH + W_{k}^H dD
\]

\[
d(U_{k})^H \right)^2 + b_k^2 A G_{p}^{l+1} \circ (\alpha_{p}^{l+1} T) (F^l) H_{k} H_{k} (U_{k})^H
\]

\[
dW_{k} + b_k A G_{p}^{l+1} \circ (\alpha_{p}^{l+1} T) (F^l) H_{k} H_{k} (U_{k})^H
\]

\[
+ \sum_{i=1}^{D_k} \frac{b_i k^2}{\sqrt{2\pi} \rho_{k,i}} e^{-\frac{||b_i k||^2}{2\sigma_{k,i}^2}} C \triangleq A \frac{b_i^2}{\rho_{k,i}^2} C \triangleq D \triangleq H_k F^* C. \]
with (28)-(30), we obtain all the necessary relations linking the various gradients in adjacent layers.

By isolating and rearranging corresponding terms in $dP^l_k$, we obtain the desired relationship for $G^l_{P_k}$ as

$$G^l_{P_k} = - \frac{1}{J} \sum_{j=1}^{J} \left( b_k H G^{l+1}_P + (\alpha^l_{P_k})^T F^l H^H U^l_k H^H \right) \left( T \right)$$

where $E \triangleq b_k (W^l_k)^H U^l_k H^H F^l$. 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 parameter updating rules in each layer 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., $\alpha^{l+1}_{P_k} = \alpha^l_{P_k} - \mu^l_{\alpha_{P_k}} \alpha^l_{P_k} \nabla^l_{P_k} P^l$, where $\mu^l_{\alpha_{P_k}}$ denotes the step size of the update in SGD of $\alpha^l_{P_k}$ in the $t$-th iteration, which incoporates an attenuation factor dependent on iteration $t$. In order to avoid vanishing gradient problem in the BP, normalization is employed after the update of $\{G^l_{P_k}, G^l_{W_k}, G^l_{U_k}, G^l_{F_k}, G^l_{P_k}\}$. The specific rule of normalizing $G^l_{P_k}$ is given as

$$G^l_{P_k} \leftarrow \frac{KG^l_{P_k}}{\sum ||G^l_{P_k}||}.$$  \hspace{1cm} (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 $\{P^0_k, W^0_k, U^0_k, \theta^0_F\}$ are initialized based on the conventional channel alignment method [6]. The training process of the deep-unfolding NN is shown in Algorithm [2], where $\mathcal{H} \triangleq \{H^0_k, H^1_k, \ldots, H^N_k\}$ 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 insight 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.

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 stationary point with much reduced number of layers.

The proofs of these claims 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 $P_t$ to $P_{t+2}$, where we omit indices $k$ and $l$ for clarity:

$$P_{t+2} = P_t - \mu P A \sum_{j=1}^{J} e^{-j(\theta^j_k)^2} \mathbb{R}\{b^j F^H_t H^H U^H_{t+1} W_{t+1}(b^j_k)^H \} \hspace{1cm} (38)$$

Similarly, in the deep-unfolding NN, we have the following mapping from $P_t$ to $P_{t+1}$ in the $l$-th layer:

$$P_{t+1} = P_t - \alpha^l_{P_k} \sum_{j=1}^{J} e^{-j(\theta^j_k)^2} \mathbb{R}\{b^j F^H_t H^H U^H_{t+1} W_{t+1}(b^j_k)^H \} \hspace{1cm} (39)$$

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., $P_t = P$, we need to prove that $P_{t+1}$ approaches $P_{t+2}$, i.e., $\|P_{t+2} - P_{t+1}\|^2 < \varepsilon$, for any $\varepsilon > 0$. In the following, we provide the analytical justification.

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

1: Generate training data set $\{X, H\}$. 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 from $\{X, H\}$. Calculate $\{\theta_F, F^l, l = 1, \ldots, L-1\}$ and $\{P_k, W_k, \theta_{P_k}, U_k, l = 1, \ldots, L\}$ based on (23)-(30).

4: Calculate $\theta_F$ based on (25). Calculate $F^l$ based on $\theta_F^l$. Substitute $\{P_k, W_k, \theta_{P_k}, F^l\}$ into (31).

5: Backward propagation: Firstly, calculate the gradients of $\{F^l, \theta_F^l\}$ for $\{P_k, W_k, \theta_{P_k}\}$. Secondly, calculate the gradients of $\{P_k, W_k, \theta_{P_k}\}$ for $\{P_k, W_k, \theta_{P_k}, F^l\}$ 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 $t > I_{max}$.
for two different cases.

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

$$\alpha_P = \mu_P 1_{R_t \times D_k}, \quad (40a)$$

$$O_P = -\mu_P A \sum_{j=1}^J e^{-\frac{(\hat{\gamma} P_j)^2}{2\sigma^2}} \mathbb{R}\{b\} F_{t+1}^H U_{t+1}^H W_{t+1}(b_{\hat{r}_j})^H, \quad (40b)$$

where $1_{R_t \times D_k}$ 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 $H$ conforms to a certain distribution, we have to illustrate the existence of parameters $\alpha_X$, $\rho_X$, and $O_X$ such that the following inequality is satisfied:

$$E[H\{\|P_{t+2} - P_{t+1}\|^2\} \leq \varepsilon. \quad (41)$$

Setting $\alpha_P = \mu_P 1_{R_t \times D_k}$ and $\rho_P = \varrho$, we need to prove

$$E[H\{O_P - \mu_P A \sum_{j=1}^J e^{-\frac{(\hat{\gamma} P_j)^2}{2\sigma^2}} \mathbb{R}\{b\} F_{t+1}^H U_{t+1}^H W_{t+1}(b_{\hat{r}_j})^H\} \leq \varepsilon. \quad (42)$$

The variables $F_{t+1}$, $U_{t+1}$, and $W_{t+1}$ are all related to $P_t$. To simplify the presentation, we only expand $W_{t+1}$ here but the other variables can be handled similarly. The left side of (42) can be expressed as

$$E[H\{O_P - \mu_P A \sum_{j=1}^J e^{-\frac{(\hat{\gamma} P_j)^2}{2\sigma^2}} \mathbb{R}\{b\} F_{t+1}^H U_{t+1}^H W_{t+1}(b_{\hat{r}_j})^H\} = \mu_P W A^2 \sum_{j=1}^J \sum_{k=1}^K e^{-\frac{(\hat{\gamma} P_j)^2}{2\sigma^2}} \mathbb{R}\{b\} F_{t+1}^H U_{t+1}^H W_{t+1}(b_{\hat{r}_j})^H) \leq \varepsilon.$$

where the first equality is obtained based on (23) and (17), and the second inequality is derived based on The Absolute Value Inequality. The term

$$\mu_P A \sum_{j=1}^J e^{-\frac{(\hat{\gamma} P_j)^2}{2\sigma^2}} \mathbb{R}\{b\} F_{t+1}^H U_{t+1}^H W_{t+1}(b_{\hat{r}_j})^H$$

is a function of $H$ and its mean value, denoted as $\gamma$ can be calculated based on $E[H]$. According to The Law of Large Numbers, when we sample enough $H$ in the calculation, as long as $O_P$ is set to $\gamma$, (43) can be simplified as

$$E[H\{\|P_{t+2} - P_{t+1}\|^2\} \leq E[H\left\{\sum_{j=1}^J \sum_{k=1}^K \|\mu P W A^2 e^{-\frac{(\hat{\gamma} P_j)^2}{2\sigma^2}} \mathbb{R}\{b\} F_{t+1}^H U_{t+1}^H W_{t+1}(b_{\hat{r}_j})^H\right\} < \varepsilon. \quad (44)$$

Through the expression, we can see that the performance terms in the right side of (44) are all less than 1, so $\varepsilon$ is a term which is far less than 1. Indeed, $\varepsilon$ 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 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 “$\Theta_P$-NN” in the top dotted rectangle at the transmitter, implement the digital transmit beamforming matrices $P_k$ and the analog transmit beamforming phase matrix $\theta_P$, respectively, where the original symbol vectors $b_k$ are converted into the precoded signals. The CNNs labeled “W_NN” and “$\Theta_W$-NN” in the bottom dotted rectangle at the receiver implement the digital receive beamforming matrices $W_k$ and the analog receive beamforming phase matrices $\theta_W$, respectively, which use the received signals $y_k$ to generate $b_k$.

In the training stage, the SER is obtained by comparing the original symbol vector $b_k$ with the detected symbol vector $\hat{b}_k$, 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 $H_k \in \mathbb{C}^{N_t \times N_t}$ is transformed into a $2 \times N_t \times N_t$ dimensional real-number tensor $\bar{H}_k$. The inputs of “$\Theta_P$-NN” and “$\Theta_W$-NN” are $H_k$ and $b_k$, and the outputs are used to generate a low-dimensional equivalent channel $\bar{H}_{eq}$, which are served as the inputs of “P_NN” and “W_NN”.
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 \(2(3K + KN_ND + KN_r,kD + KN_r,kN_r,k + R_tN_t)\). Since the parameters \(\{\alpha_{b_p}^i, \beta_{F}^i, \rho_{F}^i, \Omega_{b_p}^i\}\) for the analog transmit beamforming matrix \(F\) are not used in the last layer, i.e., \(L\)-th layer, the total number of parameter dimension in the network is \(2KL(3 + N_tD + N_rD + D_{r,k}N_{r,k}) + 2(L - 1)R_tN_t\).

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 \(\sum_{l=1}^{L_c} C_l C_{l-1} C_l\), where \(L_c\) denotes the number of convolutional layers, \(C_l\) denotes the size of convolution kernel, and \(C_1\) denotes the number of paths in the \(l\)-th convolutional layer. The parameter dimension of the fully-connected layers is given by \(K^2N_tN_{r,k}F_{in}F_{out}\), 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 \(O(TJS(KN_tD + KN_{r,k}D + KN_r,kN_r,k + R_tN_t))\), 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 \(O(\sum_{l=2}^{L_c} L_c^2 C_l C_{l-1} C_l + K^2N_tN_r,kF_{in}F_{out} + KN_tD + KN_r,kD + KN_r,kN_r,k + R_tN_t)\), where \(L_c \triangleq (C_{in} - K_t + 2P)/S_t + 1\). \(C_{in}\) 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 \(O(LJS(KN_tD + KN_{r,k}D + KN_r,kN_r,k + R_tN_t))\), 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 greatly 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 \((N_t, N_{r,k}, K)\) and the smaller system is characterized by \((N_t, N_{r,k}, K_1)\), where \(N_{r,k} \leq N_r, N_{r,k} \leq N_{r,k}\), and \(K_0 \leq K_1\). In the testing stage, we only need to input \(\{H_k, k \leq K_1\}\) and set \(\{H_k = 0, K_1 < k \leq K_0\}\). Meanwhile, the corresponding column and row vectors in \(H_k\) should be set to \(0\).

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

\[
\begin{align*}
\Re\{b_{i,k}\} = & \begin{cases} 
F_1, & \text{if } \Re(b_{i,k}) \leq c_{i,k}(F_1 + 1) \\
F_m, & \text{if } c_{i,k}(F_m - 1) < \Re(b_{i,k}) \leq c_{i,k}(F_m + 1), \\
F_{\sqrt{M}}, & \text{if } \Re(b_{i,k}) > c_{i,k}(F_{\sqrt{M} - 1}),
\end{cases} \\
\Im\{b_{i,k}\} = & \begin{cases} 
F_1, & \text{if } \Im(b_{i,k}) \leq c_{i,k}(F_1 + 1) \\
F_n, & \text{if } c_{i,k}(F_n - 1) < \Im(b_{i,k}) \leq c_{i,k}(F_n + 1), \\
F_{\sqrt{M}}, & \text{if } \Im(b_{i,k}) > c_{i,k}(F_{\sqrt{M} - 1}),
\end{cases}
\end{align*}
\]

where \(2 \leq m, n \leq \sqrt{M} - 1\) and \(c_{i,k} \triangleq w_{i,k}^H U_{k} H_{k} F_{i,k}\). 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:

\[
\begin{align*}
D_{i,k} & \leftarrow \frac{c_{i,k}}{|c_{i,k}|} D_{i,k}, & W_{i,k} & \leftarrow \frac{c_{i,k}}{|c_{i,k}|} W_{i,k}, \\
U_{k} & \leftarrow \frac{c_{i,k}}{|c_{i,k}|} U_{k}, & F & \leftarrow \frac{c_{i,k}}{|c_{i,k}|} F,
\end{align*}
\]

which make the detection rule (45) effective.

The number of the legitimate sequences of the transmitted signal \(s\) 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 \(b_{i,k}\) as \(\tilde{b}_{i,k}\). We then obtain the following expressions for the SER,

\[
\begin{align*}
P_e^R &= \frac{\varphi}{J} \sqrt{\pi} \sum_{j=1}^{J} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-s^2} ds, & (47) \\
P_e^I &= \frac{\varphi}{J} \sqrt{\pi} \sum_{j=1}^{J} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-s^2} ds, & (48)
\end{align*}
\]

where \(\varphi \triangleq 2\sqrt{M - 2}/\sqrt{M}\). [41]

The derivation of the MSER-based GD iterative 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., $\forall k \in K$, 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 [2].

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 (13). 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 a better SER performance, in both terms of convergence speed and residual SER after convergence, while a larger batch size has a smaller fluctuation of SER performance after convergence. Indeed, a reduction of the batch size introduces additional randomness in the gradient vector which prevents the NN from being trapped in local optima. To provide a good 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., $\{\mu_X, \mu_{DX}, \mu_{OPX}\}$, 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 choose the step size as $0.02 \times 0.5^{\lceil \frac{L}{2} \rceil}$ which offers a good tradeoff between system performance and convergence speed. 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 $L$ is further increased, 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 following schemes: MMSE-based fully digital GD beamforming algorithm, MMSE-based hybrid GD beamforming algorithm, MMSE-based hybrid OMP beamforming algorithm, MMSE-based hybrid MO beamforming algorithm, MSER-based fully digital GD beamforming algorithm, proposed MSER-based hybrid GD beamforming algorithm, MMSE-based benchmark CNN, MSER-based benchmark CNN and proposed MSER-based deep-unfolding NN. The results show that the proposed MSER-based algorithms significantly outperform the existing MMSE-based algorithms.
in terms of SER. Moreover, the performance of the proposed deep-unfolding NN approaches that of the MSER-based hybrid GD algorithm, and the gap between their performance decreases with the increase of SNR. The proposed deep-unfolding NN clearly outperforms the benchmark approach; while both algorithms provide much better performance compared to the MMSE-based fully digital algorithm at high SNR.

Table I presents the 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 than the black-box CNN to achieve a similar (or better) performance, while a much larger number of iterations is required by the GD algorithm to reach that 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 shows the SER for different MIMO system configurations at SNR = 20 dB. The percentages 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 benchmark CNNs under the MSER and MMSE criteria. It shows that the proposed deep-unfolding NN has a better

| TABLE I: SER versus SNR. |
|--------------------------|
| Hybrid GD | Benchmark CNN | Deep-unfolding NN |
| 500 iterations | 17 layers | 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} |

| TABLE II: SER for different MIMO system configuration at SNR = 20 dB. |
|--------------------------|
| (K, R_{r,k}) | (1, 4) | (2, 4) | (3, 2) | (4, 2) |
| Hybrid GD | 1.15 × 10^{-2} | 3.64 × 10^{-3} | 6.83 × 10^{-3} | 1.56 × 10^{-2} |
| Deep-unfolding NN | 98.43% | 97.85% | 97.07% | 95.86% |
| Benchmark CNN | 73.14% | 70.48% | 68.52% | 65.02% |
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. 9: SER performance of different schemes versus SNR in the presence of imperfect CSI (SNR = 20 dB).

Fig. 10: SER performance of different schemes versus SNR under transfer conditions.

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.

Clarity, the proposed deep-unfolding NN requires less training time compared to the CNN, which proves that the former is much smaller compared to that of the benchmark CNNs. 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.

D. SER Performance for 16-QAM Signals

Fig. 11 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 V aims to show the generalization ability of the proposed deep-unfolding NN. To this end, we first train a large-scale network with $N_t = 128$, $R_t = 32$, $N_c = 16$, $R_c = 4$, $K = 8$, and $D_h = 3$ 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 aims to show the generalization ability of the proposed deep-unfolding NN. To this end, we first train a large-scale network with $N_t = 128$, $R_t = 32$, $N_c = 16$, $R_c = 4$, $K = 8$, and $D_h = 3$ 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.

D. SER Performance for 16-QAM Signals

Fig. 11 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 V aims to show the generalization ability of the proposed deep-unfolding NN. To this end, we first train a large-scale network with $N_t = 128$, $R_t = 32$, $N_c = 16$, $R_c = 4$, $K = 8$, and $D_h = 3$ 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 IV: CPU running time of the proposed algorithm.

| (N_t, K, R_c) | 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 V: Generalization ability of the proposed deep-unfolding NN.

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

TABLE VI: SER versus the iterations/layers for SNR = 25 dB.

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

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 a MSER-based GD iterative algorithm to find stationary points. We then unfolded the GD iterative 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 GD iterative 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

\[
\min_{X} \ g(X; Z) \quad \text{s.t. } X \in \mathcal{S},
\]

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

\[
X^{l+1} = G_l(X^l; Z),
\]

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

\[
\min_{X} \mathbb{E}_Z \{g(X; \lambda, Z)\} \quad \text{s.t. } X \in \mathcal{S}.
\]

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

\[
X^{l+1} = G_l(X^l; \lambda^l, Z),
\]

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

\[
\mathbb{E}_Z \{G_l(X^l; \lambda^l, Z)\} = \mathbb{E}_Z \{G_l(X^l; \lambda^l, Z)\} - \nabla_{X^l} \mathbb{E}_Z \{G_l(X^l; \lambda^l, Z)\},
\]

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

\[
G_l^T(\cdot) = G_l^T(\cdot) \circ \mathcal{J}(X^l; \lambda^l, Z),
\]

In this work, when \( X \) denotes \( P_k \), we find that \( \mathcal{J}(\cdot) = \mathcal{J}_1(\cdot) \circ \mathcal{J}_2(\cdot) \), where \( \mathcal{J}_1(\cdot) = \mathbb{E} - b_k^T \mathcal{B}(\alpha_{P_k})^T (F^l)^H H_k^H (U_k)^H W_k^T \) and \( \mathcal{J}_2(\cdot) = \mathbb{E} \), where \( \mathbb{E} \) 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.

$$\nabla_{\alpha_{pk}} \hat{P}_e = - \gamma (\mathbf{G}^H_{pk} \circ \nabla \mathbf{P}_e) \hat{P}_e^H, \quad \nabla_{\gamma} \hat{P}_e = (\mathbf{G}^H_{pk})^H, \quad \nabla_{\theta_F} \hat{P}_e = (\mathbf{G}^H_{pk})^H \circ \mathbf{F}^H \circ \nabla \mathbf{P}_e, \quad \nabla_{\theta_J} \hat{P}_e = (\mathbf{G}^H_{pk})^H \circ \mathbf{J} \circ \nabla \mathbf{P}_e, \quad \nabla_{\theta_D} \hat{P}_e = (\mathbf{G}^H_{pk})^H \circ \mathbf{D} \circ \nabla \mathbf{P}_e, \quad \nabla_{\theta_B} \hat{P}_e = (\mathbf{G}^H_{pk})^H \circ \mathbf{B} \circ \nabla \mathbf{P}_e, \quad \nabla_{\theta_G} \hat{P}_e = (\mathbf{G}^H_{pk})^H \circ \mathbf{G} \circ \nabla \mathbf{P}_e.
$$

$$\nabla_{\mathbf{P}_e} \mathbf{P}_e = \frac{1}{2} \mathbf{J} \gamma (\mathbf{P}_e^H)^2 \circ \left[ \left( \frac{\mathbf{G}^H_{pk}}{2} \right) \mathbf{P}_e \right]^{\mathbf{F}^H \mathbf{H}^H_{pk} \mathbf{U}_k} \left( \mathbf{w}_{i,k} \right).
$$

(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 mobile communications for 5G cellular,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Apr. 2018, pp. 3579-3583.

[4] K. L. Law and C. Maseuros, “Symbol error rate minimization precoding for interference exploitation,” IEEE Trans. Commun., vol. 66, no. 11, pp. 5718-5731, Nov. 2018.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidirooulos, “Learning to optimize: Training deep neural networks for interference management,” IEEE Trans. Signal Process., vol. 66, no. 20, pp. 5438-5453, Oct. 2018.

[11] 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.

[12] F. Liang, C. Shen, W. Yu, and F. Wu, “Towards optimal power control via ensemble deep neural networks,” IEEE Trans. Commun., vol. 68, no. 8, pp. 1760-1770, Mar. 2020.

[13] 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.

[14] 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.

[15] C. Sidharth, S. M. Hirenath, and S. K. Patra, “Deep learning based hybrid precoding for mmWave massive MIMO system using conceptNet,” in International Conference on Communication and Signal Processing (ICCSSP), Jul. 2020, pp. 1317-1321.

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

[17] 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.

[18] N. Samuel, T. Diskin, and A. Wiesel, “Learning to detect,” IEEE Trans. Signal Process., vol. 67, no. 10, pp. 2519-2554, May 2019.

[19] 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.

[20] W. Xu, X. You, C. Zhang, and Y. Be’ery, “Polar decoding on sparse graphs with deep learning,” in 52nd Asilomar Conference on Signals, Systems, and Computers, Oct. 2018, pp. 599-603.

[21] 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.

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

[23] Y. Han, K. He, L. Bou, 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.

[24] N. Sun, Z. Zhang, S. Jin, C. -K. Wen, and G. Y. Li, “Model-driven deep learning for massive MIMO-OFDM with finite-alphabet precoding,” IEEE Wireless Commun. Lett., vol. 24, no. 10, pp. 2216-2220, Oct. 2020.

[25] Y. Qiu, H. Cai, Q. Shi, K. Xu, G. Yu, and Z. Ding, “Iterative algorithm based deep-unfolding neural networks: Precoding design for massive MIMO systems,” IEEE Trans. Wireless Commun., vol. 20, no. 2, pp. 1394-1410, Feb. 2021.

[26] 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. Wireless Commun., vol. 12, no. 1, pp. 132-143, Feb. 2013.

[27] M. Yao, Z. Wang, C. Fu, and X. Chen, “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.

[28] Y. Cai, R. C. de Lamare, A. 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.

[29] Y. Cai, R. C. de Lamare, A. 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.

[30] Y. Cai, R. C. de Lamare, A. 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.

[31] M. Zhao, “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.