Joint Channel Estimation and Hybrid Beamforming via Deep-Unfolding

Kai Kang *, Qiyu Hu *, Yunlong Cai *, Guanding Yu *, Jakob Hoydis†, and Yonina C. Eldar‡
* College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China
† NVIDIA, 06906 Sophia Antipolis, France
‡ Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 7610001, Israel
E-mail: {kangkai, qiyhu, ylcai, yuguanding}@zju.edu.cn, jhoydis@nvidia.com, yonina.eldar@weizmann.ac.il

***Abstract*—In this work, we propose an end-to-end deep-unfolding neural network (NN) based joint channel estimation and hybrid beamforming (JCEHB) algorithm to maximize the sum rate in massive multiple-input multiple-output (MIMO) systems. Specifically, the recursive least-squares (RLS) and stochastic successive convex approximation (SSCA) algorithms are unfolded for channel estimation and hybrid beamforming, respectively. We consider a mixed-timescale scheme, where analog beamforming matrices are designed based on the channel state information (CSI) statistics once in each frame, while the digital beamforming matrices are designed at each time slot based on the equivalent CSI matrices. Simulation results show that the proposed algorithm can significantly outperform conventional algorithms.**

***Index Terms*—Deep-unfolding, hybrid beamforming, channel estimation, mixed-timescale scheme, massive MIMO.**

I. Introduction

Hybrid analog-digital beamforming in massive multiple-input multiple-output (MIMO) systems with a small number of radio frequency (RF) chains has received a lot of attention in recent years [1], [2]. There have been a number of algorithms proposed for hybrid beamforming and channel estimation in massive MIMO systems [3]–[7]. In [3] and [4] a hybrid beamforming framework was suggested for improving the bit error rate and system sum rate performance, respectively. Considering hardware constraints, codebook-based methods for hybrid beamforming were investigated in [5]. Channel estimation plays an important role in hybrid beamforming design [6], [7]. The authors of [6] developed an algorithm that uses a Hidden Markov Model (HMM) for sparse channel estimation. A recursive least-squares (RLS) adaptive estimation algorithm was developed for MIMO interference channels in [7], which can track time-varying channels as the wireless environment changes.

Conventional single-timescale hybrid beamformers are optimized based on the high-dimensional full channel state information (CSI), which leads to large signaling overhead and transmission delay. To address these issues, several hybrid beamforming algorithms under mixed-timescale schemes have been investigated in [8], [9]. In this approach, long-term analog beamformers are optimized based on the channel statistics while the short-term digital beamformers are updated based on reduced-dimensional CSI. However, these approaches typically require high complexity and signaling overhead. Moreover, these two modules are generally designed separately, which may result in performance loss. We consider a joint design for channel estimation and hybrid beamforming with low-complexity and reduced overhead.

In recent years, deep-unfolding neural networks (NNs) have been applied to communications [10]–[12]. These techniques unfold iterative algorithms into layer-wise networks and introduce trainable parameters to improve system performance. The authors of [12] proposed a symbol detector named ViterbiNet, which integrates black-box NNs into the Viterbi algorithm. ViterbiNet does not obtain CSI and can be employed for complex channel models with reduced computational complexity.

In this work, we propose an end-to-end deep-unfolding scheme for joint channel estimation and hybrid beamforming (JCEHB) design in massive MIMO systems to maximize the system sum rate. The proposed approach consists of channel estimation deep-unfolding NN (CEDUN) and hybrid beamforming deep-unfolding NN (HBDUN). For the CEDUN, we design the pilot training module and unfold the RLS algorithm into a layer-wise NN with introduced trainable parameters. For the HBDUN, we propose a stochastic successive convex approximation (SSCA) algorithm induced deep-unfolding NN, where the high computational complexity operations are replaced by trainable parameters. Moreover, to reduce the signaling overhead, we consider a mixed-timescale hybrid beamforming scheme, where we employ the channel statistics to update the analog beamformers once in each frame, which consists of several time slots; while we employ the estimated reduced-dimensional equivalent CSI matrices to design the digital beamformers at each time slot. Simulation results show that our proposed deep-unfolding algorithm can significantly outperform the conventional RLS and SSCA algorithms with reduced complexity.

II. System Model and Problem Formulation

In this section, we first introduce the system model for downlink massive MIMO and then formulate our problem mathematically.

A. System Model

1) Signal Model: Consider a downlink massive MIMO system working in time-division duplex (TDD) mode. The base station (BS) is equipped with N_t transmit antennas and N_t^{RF} RF chains, sending N_s data streams to each user at the receiver with K users, where $KN_s \leq N_t^{RF} \leq N_t$. Each user is equipped with N_r receive antennas and N_r^{RF} RF chains, where $N_s \leq N_r^{RF} \leq N_r$. At the transmitter, the RF chains are connected with a network of phase shifters that expands the N_t^{RF} digital outputs to N_t precoded analog signals feeding the transmit antennas. Similarly, at the receiver, the N_r receive antennas are followed by a network of phase shifters that feeds the N_r^{RF} RF chains. The BS sends N_s data streams to user $k \in \mathcal{K} \triangleq \{1, \ldots, K\}$, denoted as $\mathbf{s}_k \in \mathbb{C}^{N_s \times 1}$. The received signal vector for user k is

$$\begin{aligned}\mathbf{y}_k = &\mathbf{W}_{BB,k}^H \mathbf{W}_{RF,k}^H \mathbf{H}_k \mathbf{F}_{RF,k} \mathbf{F}_{BB,k} \mathbf{s}_k + \mathbf{W}_{BB,k}^H \\ &\mathbf{W}_{RF,k}^H \mathbf{H}_k \sum_{m=1, m\neq k}^{K} \mathbf{F}_{RF,m} \mathbf{F}_{BB,m} \mathbf{s}_m + \mathbf{W}_{BB,k}^H \mathbf{W}_{RF,k}^H \mathbf{z}_k,\end{aligned} \tag{1}$$

This work was supported in part by the National Natural Science Foundation of China under Grants 61971376 and 61831004, and in part by the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars under Grant LR19F010002.

where $\mathbf{H}_k \in \mathbb{C}^{N_r \times N_t}$ represents the channel matrix, $\mathbf{z}_k \in \mathbb{C}^{N_r \times 1} \sim \mathcal{CN}(\mathbf{0}, \sigma_k^2 \mathbf{I})$ is the additive white Gaussian noise (AWGN) with σ_k denoting the noise power, $\mathbf{F}_{RF,k} \in \mathbb{C}^{N_t \times N_t^{RF}}$ and $\boldsymbol{W}_{RF,k} \in \mathbb{C}^{N_r \times N_r^{RF}}$ are the analog beamformers which are subject to a unit modulus constraint, i.e.,

$$\| [\mathbf{F}_{RF,k}]_{ij} \|^2 = \frac{1}{\sqrt{N_t}}, \forall k, i, j, \tag{2a}$$

$$\| [\mathbf{W}_{RF,k}]_{ij} \|^2 = \frac{1}{\sqrt{N_r}}, \forall k, i, j, \tag{2b}$$

while $\mathbf{F}_{BB,k} \triangleq [\mathbf{f}_{BB,k,1}, \ldots, \mathbf{f}_{BB,k,Ns}] \in \mathbb{C}^{N_t^{RF} \times N_s}$ and $\boldsymbol{W}_{BB,k} \triangleq [\mathbf{w}_{BB,k,1}, \ldots, \mathbf{w}_{BB,k,Ns}] \in \mathbb{C}^{N_r^{RF} \times N_s}$ are the digital beamformers. The digital precoder $\mathbf{F}_{BB,k}$ is normalized as $\|\mathbf{F}_{RF,k}\mathbf{F}_{BB,k}\|_F^2 = N_s$ to ensure that the power constraint is satisfied at the BS. Using (1), the signal-to-interference-plus-noise ratio (SINR) for stream l of user k is

$$\Gamma_{k,l} = \frac{|\mathbf{w}_{BB,k,l}^H \mathbf{H}_{eq,k} \mathbf{f}_{BB,k,l}|^2}{\sum\limits_{\substack{i=1 \\ (i,j \neq k,l)}}^{K} \sum\limits_{j=1}^{N_s} |\mathbf{w}_{BB,k,l}^H \tilde{\mathbf{H}}_{eq,k} \mathbf{f}_{BB,i,j}|^2 + \sigma_k^2 \|\mathbf{w}_{BB,k,l}^H \mathbf{W}_{RF,k}^H\|^2}, \tag{3}$$

where $\mathbf{H}_{eq,k} = \mathbf{W}_{RF,k}^H \mathbf{H}_k \mathbf{F}_{RF,k} \in \mathbb{C}^{N_r^{RF} \times N_t^{RF}}$ denotes the low-dimensional equivalent CSI matrix. The system sum rate can be calculated as

$$\sum_{k=1}^{K} \sum_{l=1}^{N_s} \log(1 + \Gamma_{k,l}). \tag{4}$$

2) Channel Estimation: It is essential for the BS to obtain CSI for hybrid beamforming. Here we consider the estimation of the low-dimensional equivalent CSI. Thanks to the channel reciprocity in TDD systems, we only need to estimate the uplink channels. Thus, we consider an uplink pilot training stage before data transmission. The k-th user first sends training pilots $\tilde{\mathbf{X}}_{eq,k} \in \mathbb{C}^{N_t^{RF} \times L}$ to the BS, where L denotes the length of pilots. Then, the received signal $\tilde{\mathbf{Y}}_{eq,k} \in \mathbb{C}^{N_r^{RF} \times L}$ at the BS is given by

$$\tilde{\mathbf{Y}}_{eq,k} = \mathbf{H}_{eq,k} \tilde{\mathbf{X}}_{eq,k} + \mathbf{W}_{RF,k}^H \mathbf{H}_k \sum_{u=1, u \neq k}^{K} \mathbf{F}_{RF,u} \tilde{\mathbf{X}}_{eq,u} + \tilde{\mathbf{Z}}_{eq,k}, \tag{5}$$

where $\tilde{\mathbf{Z}}_{eq,k}$ denotes AWGN. The transmitted pilot signal in the l-th pilot slot (the l-th column of $\tilde{\mathbf{X}}_{eq,k}$) should meet the power constraint: $\|\tilde{\mathbf{x}}_{eq,k,l}\|^2 \leq P$. The BS estimates the channel $\hat{\mathbf{H}}_{eq,k} \in \mathbb{C}^{N_r \times N_t}$ based on the received signal $\tilde{\mathbf{Y}}_{eq,k}$ and the pilot $\tilde{\mathbf{X}}_{eq,k}$ via

$$\hat{\mathbf{H}}_{eq,k} = \mathcal{F}(\tilde{\mathbf{Y}}_{eq,k}, \tilde{\mathbf{X}}_{eq,k}), \tag{6}$$

where $\mathcal{F}(\cdot)$ denotes a specific channel estimation algorithm.

3) Hybrid Beamforming: After acquiring channel information, the BS designs the hybrid beamformers based on the channel $\mathbf{H}_k$. For each user k, the hybrid beamforming design scheme at the transmitter is denoted as

$$\{\mathbf{F}_{RF,k}, \mathbf{W}_{RF,k}, \mathbf{F}_{BB,k}, \mathbf{W}_{BB,k}\} = \mathcal{Q}(\mathbf{H}_k). \tag{7}$$

B. Mixed-Timescale Frame Structure

We consider a practical mixed-timescale frame structure as shown in Fig. 1, which takes into consideration both the instantaneous CSI and the channel statistics. We consider a frame during which the channel statistics are constant. It consists of T_s time slots and the instantaneous CSI remains unchanged during each time slot. We introduce two different timescales as follows:

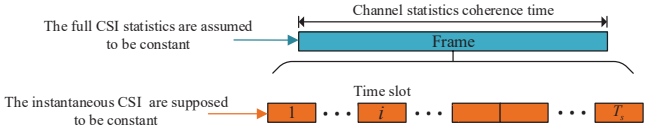

Fig. 1: Mixed-timescale frame structure.

- Long-timescale: The channel statistics are unchanged during each frame which consists of several time slots.
- Short-timescale: The instantaneous CSI is constant in each time slot.

In general, the dimension of the equivalent CSI matrix $\mathbf{H}_{eq} \in \mathbb{C}^{N_r^{RF} \times N_t^{RF}}$ is much smaller than that of the full CSI matrix $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$. Thus, we consider acquiring the low-dimensional equivalent CSI at each time slot. In this way, we optimize the analog and digital beamformers at different timescales. We update the long-term analog beamformers $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ based on full CSI, while we optimize the short-term digital beamformers $\{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$ based on the low-dimensional equivalent CSI at each time slot.

C. Problem Formulation

We aim at jointly designing the mixed-timescale hybrid beamforming and channel estimation to maximize the system sum rate. The optimization problem can be formulated as

$$\max_{\mathcal{X}, \mathcal{Y}} \quad \sum_{k=1}^{K} \sum_{l=1}^{N_s} \log(1 + \Gamma_{k,l}), \tag{8a}$$

$$\text{s.t.} \quad \| \mathbf{F}_{RF,k} \mathbf{F}_{BB,k}^i \|_F^2 = N_s, \forall k, \tag{8b}$$

$$\|\tilde{\mathbf{x}}_{eq,k,l}\|^2 \leq P, \forall k, l, \tag{8c}$$

$$\hat{\mathbf{H}}_{eq,k} = \mathcal{F}(\tilde{\mathbf{Y}}_{eq,k}, \tilde{\mathbf{X}}_{eq,k}), \forall k, \tag{8d}$$

$$\{\mathbf{F}_{RF,k}, \mathbf{W}_{RF,k}\} = \mathcal{Q}_{fu}(\mathbf{H}_k), \forall k, \tag{8e}$$

$$\{\mathbf{F}_{BB,k}, \mathbf{W}_{BB,k}\} = \mathcal{Q}_{eq}(\hat{\mathbf{H}}_{eq,k}), \forall k, \tag{8f}$$

$$\text{(2a), (2b)}, \tag{8g}$$

where $\mathcal{X} \triangleq \{\mathbf{F}_{RF,k}, \mathbf{W}_{RF,k}, \mathbf{F}_{BB,k}, \mathbf{W}_{BB,k}\}$ and $\mathcal{Y} \triangleq \{\tilde{\mathbf{X}}_{eq,k}\}$, $\mathcal{Q}_{eq}(\cdot)$ and $\mathcal{Q}_{fu}(\cdot)$ represent the analog and digital beamforming design, respectively. In the following, a deep-unfolding framework is proposed for tackling this problem.

III. Mixed-Timescale Deep-Unfolding

A. The Structure of the Deep-Unfolding Framework

We propose a deep-unfolding framework, the structure of which is shown in Fig. 2. The analog and digital NNs are designed for hybrid beamforming while the CEDUN is designed for channel estimation for the low-dimensional CSI. Based on this framework, we introduce the training stage and the data transmission stage in the following.

1) The Training Stage: First, we obtain the channel samples offline[1] and input them into the analog NN to obtain analog beamformers $\{\mathbf{W}_{RF}, \mathbf{F}_{RF}\}$ and the low-dimensional equivalent CSI matrix $\mathbf{H}_{eq}$. Then $\mathbf{H}_{eq}$ passes through the CEDUN that outputs the estimated equivalent CSI matrix $\hat{\mathbf{H}}_{eq}$. Finally, $\hat{\mathbf{H}}_{eq}$ passes through the digital NN that outputs

[1]If the channel statistics is known, we can obtain channel samples based on it. If the channel statistics is unknown, we can send pilots to estimate the full CSI at regular intervals to estimate CSI.

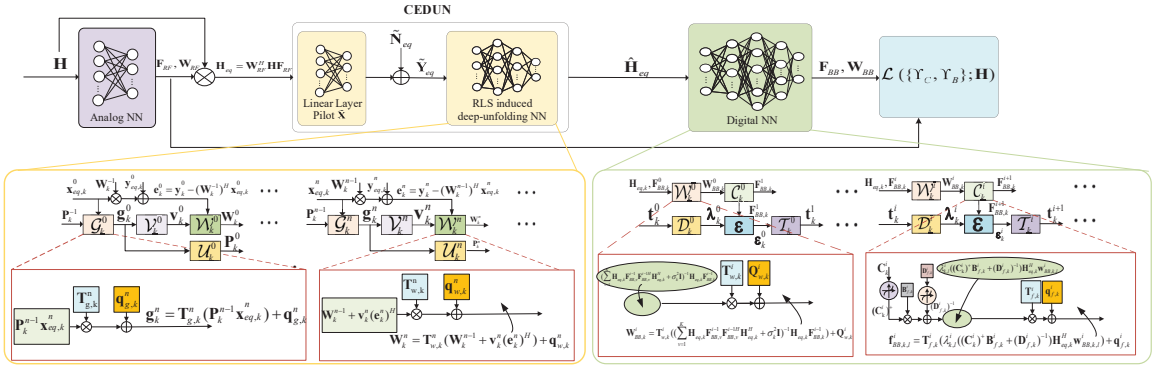

Fig. 2: Architecture of the proposed deep-unfolding framework.

Algorithm 1 The RLS algorithm for channel estimation

1: Input: Pilot $\tilde{\mathbf{X}}_{eq,k}$ and received signal $\tilde{\mathbf{Y}}_{eq,k}$;
2: Initialize the estimated matrix $\mathbf{W}_k^{-1} = \mathbf{0}$ and intermediate variable $\mathbf{P}_k^{-1} = \delta^{-1}\mathbf{I}$;
3: **for** $n = 1, 2, \ldots, L$ **do**
4: Update $\{\mathbf{g}_k^n\}$ based on $\mathbf{g}_k^n = \mathbf{P}_k^{n-1}\tilde{\mathbf{x}}_{eq,k}^{n-1}$;
5: Update $\{\mathbf{v}_k^n\}$ based on $\mathbf{v}_k^n = \frac{\mathbf{g}_k^n}{\beta_k + (\mathbf{g}_k^n)^H \tilde{\mathbf{x}}_{eq,k}^n}$;
6: Update $\{\mathbf{P}_k^n\}$ based on $\mathbf{P}_k^n = \beta_k^{-1}(\mathbf{P}_k^{n-1} - \mathbf{v}_k^n(\mathbf{g}_k^n)^H)$;
7: Calculate residual $\mathbf{e}_k^n = \tilde{\mathbf{y}}_k^n - (\mathbf{W}_k^{n-1})^H \tilde{\mathbf{x}}_{eq,k}^n$;
8: Update the estimated matrix $\mathbf{W}_k^n = \mathbf{W}_k^{n-1} + \mathbf{v}_k^n(\mathbf{e}_k^n)^H$;
9: $n = n + 1$;
10: **end for**

the digital beamformers $\{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$. The loss function is the system sum rate, which is denoted as $\mathcal{L}(\{\boldsymbol{\Upsilon}_C, \boldsymbol{\Upsilon}_B\}; \mathbf{H})$, where $\boldsymbol{\Upsilon}_B$ consists of $\boldsymbol{\Psi}$ and $\boldsymbol{\Omega}$, which represent the trainable parameters of the analog NN and digital NN, respectively, and $\boldsymbol{\Upsilon}_C$ represents the trainable parameteres of the CEDUN.

2) The Data Transmission Stage: We first train the deep-unfolding NN as mentioned above. Then in data transmission, we fix the analog beamfomers and the pilots during the channel statistics coherence time and estimate the low-dimensional equivalent real-time CSI by the CEDUN to update the digital beamformers. The analog beamformers obtained by the offline training can well adapt to CSI statistics when it does not change [8]. When it changes, we need to obtain several full CSI samples and employ transfer learning [13] to fine tune the parameters of the deep-unfolding NN, where the analog beamformers are updated for tracking the changes of CSI statistics.

B. Deep-Unfolding NN for Channel Estimation

1) The RLS Channel Estimation Algorithm: The procedure of the RLS channel estimation algorithm is presented in Algorithm 1, where $\tilde{\mathbf{x}}_{eq,k}^n$ and $\tilde{\mathbf{y}}_{eq,k}^n$ are the n-th column of $\tilde{\mathbf{X}}_{eq,k}$ and $\tilde{\mathbf{Y}}_{eq,k}$, respectively. Note that $\mathbf{g}_k^n$, $\mathbf{v}_k^n$, and $\mathbf{P}_k^n$ are intermediate variables, $\beta_k \in (0,1)$ is the forgetting factor and δ denotes a small positive number, $\mathbf{W}_k^n$ denotes the weight matrix and the estimated channel $\hat{\mathbf{H}}_{eq,k} = (\mathbf{W}_k^n)^H$. Moreover, the estimation error $\mathbf{e}_k^n$ descends with the update of $\mathbf{W}_k^n$. The number of iterations is the length of pilots L. In addition, the inputs of the algorithm are the pilots and received signal, and the output is the estimated channel matrix.

2) Deep-Unfolding NN for Channel Estimation: Based on the RLS algorithm, we propose the CEDUN which contains the pilot training NN and RLS induced deep-unfolding NN. The structure of the CEDUN for user k is shown in Fig. 2.

In the pilot training NN, we set the pilots as trainable parameters. As shown in Fig. 2, to model the process of pilot training for estimating the low-dimensional equivalent CSI matrix $\mathbf{H}_{eq,k}$, the input and output of the NN are $\mathbf{H}_{eq,k}$ and $\tilde{\mathbf{Y}}_{eq,k}$, respectively, and $\tilde{\mathbf{X}}_{eq,k}$ is set as the trainable parameter. Note that $\tilde{\mathbf{X}}_{eq,k}$ needs to be scaled to satisfy the power constraint (8c).

For the RLS induced deep-unfolding NN, we unfold the RLS algorithm into a network with significantly less layers. The inputs of the n-th layer of the NN are $\{\tilde{\mathbf{x}}_{eq,k}^n, \tilde{\mathbf{y}}_{eq,k}^n, \mathbf{P}_k^{n-1}, \mathbf{W}_k^{n-1}\}$ and the outputs are $\{\mathbf{P}_k^n, \mathbf{W}_k^n\}$. To increase the degrees of freedom for the parameters and speed up the convergence, we employ the structure $\mathbf{Y}_{out}^n = \mathbf{T}_y^n \mathbf{X}_{in}^n + \mathbf{q}_y^n$ in the n-th layer of the NN, where $\mathbf{X}_{in}^n$ and $\mathbf{Y}_{out}^n$ represent the input and output, $\mathbf{T}_y^n$ and $\mathbf{q}_y^n$ are the defined introduced multiplier and offset trainable parameters of the n-th layer, respectively. The structure is inspired by the neurons of conventional DNNs, e.g., $\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$, where the multiplier trainable parameter $\mathbf{T}_y^n$ corresponds to the weight matrix $\mathbf{W}$ and offset trainable parameter $\mathbf{q}_y^n$ corresponds to the bias vector $\mathbf{b}$. Note that $\varpi_C^n \triangleq \{\mathbf{T}_{g,k}^n, \mathbf{q}_{g,k}^n\} \cup \{\mathbf{T}_{v,k}^n, \mathbf{q}_{v,k}^n\} \cup \{\mathbf{T}_{p,k}^n, \mathbf{q}_{p,k}^n\} \cup \{\mathbf{T}_{w,k}^n, \mathbf{q}_{w,k}^n\}$ are the multiplier and offset trainable parameters to update the variables $\mathbf{g}_k^n$, $\mathbf{v}_k^n$, $\mathbf{P}_k^n$, and $\mathbf{W}_k^n$ in the n-th layer, respectively. As shown in Fig. 2, $\mathcal{G}_k^n$, $\mathcal{V}_k^n$, $\mathcal{U}_k^n$, $\mathcal{W}_k^n$ represent the sub-layers of the n-th layer of the deep-unfolding NN, i.e., (9a)-(9d). The forgetting factor β_k^n is also set as trainable parameter γ_k^n:

$$\mathbf{g}_k^n = \mathbf{T}_{g,k}^n(\mathbf{P}_k^{n-1}\tilde{\mathbf{x}}_{eq,k}^n) + \mathbf{q}_{g,k}^n, \tag{9a}$$

$$\mathbf{v}_k^n = \mathbf{T}_{v,k}^n\left(\frac{\mathbf{g}_k^n}{\gamma_k^n + (\mathbf{g}_k^n)^H \tilde{\mathbf{x}}_{eq,k}^n}\right) + \mathbf{q}_{v,k}^n, \tag{9b}$$

$$\mathbf{P}_k^n = \mathbf{T}_{p,k}^n(\gamma_k^n)^{-1}(\mathbf{P}_k^{n-1} - \mathbf{v}_k^n(\mathbf{g}_k^n)^H) + \mathbf{q}_{p,k}^n, \tag{9c}$$

$$\mathbf{W}_k^n = \mathbf{T}_{w,k}^n(\mathbf{W}_k^{n-1} + \mathbf{v}_k^n(\mathbf{e}_k^n)^H) + \mathbf{q}_{w,k}^n. \tag{9d}$$

Thus, the trainable parameters of the CEDUN can be denoted as $\boldsymbol{\Upsilon}_C \triangleq \tilde{\mathbf{X}}_{eq,k} \cup \{\bigcup_{n=1}^{L_c} \varpi_C^n \cup \gamma_k^n\}$, where L_c is the number of layers of the RLS induced deep-unfolding NN.

C. Deep-Unfolding NN for Hybrid Beamforming

1) The SSCA-based Hybrid Beamforming Algorithm: For the hybrid beamforming design, we first introduce the mixed-

timescale SSCA framework [8]. We optimize the analog beamformers based on the full CSI samples $\mathbf{H}$. The digital beamformers are fixed and we update the analog beamformers by employing a convex surrogate function to replace the objective function and taking its derivative [8]. By fixing the analog beamformers, we optimize the digital beamformers based on the low-dimensional equivalent CSI matrix $\mathbf{H}_{eq}$. We adopt the successive convex approximation (SCA) algorithm to optimize the short-term digital beamformers [14].

$$\mathbf{W}^i_{BB,k} = \mathbf{T}^i_{w,k}\Big((\sum_{v=1}^{K}\mathbf{H}_{eq,k}\mathbf{F}^{i-1}_{BB,v}\mathbf{F}^{i-1H}_{BB,v}\mathbf{H}^H_{eq,k}+\sigma_k^2\mathbf{I})^{-1}\mathbf{H}_{eq,k}\mathbf{F}^{i-1}_{BB,k}\Big)+\mathbf{Q}^i_{w,k}, \tag{10a}$$

$$\lambda^i_{k,l} = \mathbf{T}^i_{\lambda,k}(\mu^i_k\alpha^{\mathbf{t}^i_{k,l}})+\mathbf{q}^i_{\lambda,k}, \tag{10b}$$

$$\mathbf{f}^i_{BB,k,l}=\mathbf{T}^i_{f,k}\Big(\lambda^i_{k,l}((\mathbf{C}^i_k)^{+}\mathbf{B}^i_{f,k}+(\mathbf{D}^i_{f,k})^{-})\mathbf{H}^H_{eq,k}\mathbf{w}^i_{BB,k,l}\Big)+\mathbf{q}^i_{f,k}, \tag{10c}$$

$$\mathbf{t}^i_{k,l} = \mathbf{T}^i_{t,k}(\mathbf{t}^{i-1}_{k,l}+\mu^i_k(1-\epsilon_{k,l}\alpha^{\mathbf{t}^{i-1}_{k,l}}))+\mathbf{q}^i_{t,k}. \tag{10d}$$

2) The SSCA Algorithm Induced Deep-Unfolding NN: We unfold the SSCA algorithm leading to HBDUN. The structure of HBDUN is shown in Fig. 2, which consists of the analog NN and the digital NN.

The input of the analog NN is the full channel samples $\mathbf{H}$ and the outputs are the analog beamforming matrices $\{\mathbf{F}_{RF,k},\mathbf{W}_{RF,k}\}$. We set the angle of the phase shifters for analog beamformers as trainable parameters $\mathbf{\Psi}\triangleq\{\mathbf{\Psi}_F,\mathbf{\Psi}_W\}$ of the analog NN and employ the operation $e^{j(\cdot)}$ to satisfy the unit modulus constraint.

The input of the digital NN is the low-dimensional real-time equivalent CSI matrix $\mathbf{H}_{eq,k}$ and the outputs are the digital beamforming matrix $\{\mathbf{F}_{BB,k},\mathbf{W}_{BB,k}\}$. We next introduce the detailed structure of the digital NN, which unfolds the SCA algorithm into a layer-wise structure. Two non-linear operations are defined for approximating matrix inversion. First, we take the inverse of the diagonal entries of matrix $\mathbf{A}$ and set the other elements as zero and denote the result as $\mathbf{A}^{+}$. Next, we set the imaginary part of the diagonal elements of $\mathbf{D}$ to zero, expressed as $\mathbf{D}^{-}$. Based on this, we employ the following structure to approximate the matrix inversion.

- Firstly, we use $\mathbf{A}^{+}\mathbf{B}$ with the non-linear operation $\mathbf{A}^{+}$ and trainable parameter $\mathbf{B}$, where $\mathbf{B}$ is introduced to improve performance.
- Next, we introduce the offset trainable matrix $\mathbf{D}$ to better approximate the inverse matrix. In addition, we find that the imaginary part of the diagonal elements of the inverse matrix is close to zero so we apply $\mathbf{D}^{-}$.

Thus, the matrix inversion $\mathbf{A}^{-1}$ is approximated by employing $\mathbf{A}^{+}\mathbf{B}+\mathbf{D}^{-}$. Note that we introduce trainable parameters $\{\mathbf{B}^i_{f,k},\mathbf{D}^i_{f,k}\}$ to approximate the inversion of variable $\mathbf{f}^i_{BB,k,l}$ in the i-th layer, which reduces the computational complexity. To increase the degrees of freedom for the parameters, the multiplier and offset trainable parameters $\mathbf{\varpi}^i_B\triangleq\{\mathbf{T}^i_{w,k},\mathbf{Q}^i_{w,k}\}\cup\{\mathbf{T}^i_{\lambda,k},\mathbf{q}^i_{\lambda,k}\}\cup\{\mathbf{T}^i_{f,k},\mathbf{q}^i_{f,k}\}\cup\{\mathbf{T}^i_{t,k},\mathbf{q}^i_{t,k}\}$ are introduced in updating the variables $\mathbf{W}^i_{BB,k}$, $\lambda^i_{k,l}$, $\mathbf{f}^i_{BB,k,l}$, and $\mathbf{t}^i_{k,l}$ in the i-th layer, respectively. As shown in Fig. 2, $\mathcal{W}^i_k$, $\mathcal{D}^i_k$, $\mathcal{C}^i_k$, and $\mathcal{T}^i_k$ represent the sub-layers of the i-th layer of the deep-unfolding NN, i.e., (10a)-(10d), where

$$\mathbf{C}^i_k\triangleq\sum_{(m,n)}^{(K,N_s)}\lambda^i_{m,n}\mathbf{H}^H_{eq,m}\mathbf{w}^i_{BB,m,n}(\mathbf{w}^i_{BB,m,n})^H\mathbf{H}_{eq,m}+v^i_k\mathbf{I}. \tag{11}$$

The constant $\frac{1}{\log\alpha}$ is set as trainable parameter μ^i_k to speed up convergence. Thus, the trainable parameters of the digital NN are $\mathbf{\Omega}\triangleq\bigcup_{i=1}^{L_h}\{\mathbf{B}^i_{f,k},\mathbf{D}^i_{f,k}\}\cup\mathbf{\varpi}^i_B\cup\mu^i_k$, where L_h is the number of layers. All the trainable parameters of the HBDUN are denoted as $\mathbf{\Upsilon}_B\triangleq\mathbf{\Psi}\cup\mathbf{\Omega}$. To avoid gradient explosion and satisfy the power constraint, we normalize $\mathbf{F}^i_{BB,k}$ by N_s at each layer to $\frac{\sqrt{N_s}}{\|\mathbf{F}_{RF,k}\mathbf{F}^i_{BB,k}\|_F}\mathbf{F}^i_{BB,k}$.

IV. Simulation Results

In this section, we verify the effectiveness of the proposed deep-unfolding algorithm based on simulation results.

A. Simulation Setup

The simulation setting is given as follows. We set $N_t=64$, $N^{RF}_t=16$, $N_r=32$, $N^{RF}_r=4$, $K=4$, and $N_s=4$. We set the signal-to-noise ratio (SNR) as 10 dB and the number of the layers for CEDUN and HBDUN are 16 and 5, respectively. We employ the 3GPP TR 38.901 channel model [15] to generate the channel data.

We provide the following benchmarks as comparison:

- Joint design NN: The proposed mixed-timescale deep-unfolding NN which jointly trains the CEDUN and HBDUN.
- Separate Design NN: The proposed mixed-timescale deep-unfolding NN which separately trains the CEDUN and HBDUN with the minimization of MSE and the maximization of sum rate as the loss function, respectively.
- HBDUN: The proposed deep-unfolding NN for the design of hybrid beamforming with perfect CSI.
- RLS-SSCA: The cascaded RLS algorithm for the design of channel estimation and the SSCA algorithm for the hybrid beamforming design.
- RLS-ZF: The cascaded RLS algorithm for the design of channel estimation and the zero-forcing (ZF) algorithm for the hybrid beamforming design.

B. System Sum Rate

Fig. 3 illustrates the sum rate of the proposed network and the benchmarks for different values of SNR. It can be seen that the proposed separate deep-unfolding NN achieves comparable performance to the conventional SSCA and RLS algorithms, which indicates the effectiveness of our deep-unfolding NN. The results also illustrate the superiority of the joint design compared to separate design NN.

Table I manifests the sum rate versus the number of users K. We normalize the sum rate of NNs by the corresponding value of the RLS-SSCA algorithm. We can see that when K is small, e.g., $K=2$, the proposed separate design NN can achieve 97.23% performance of the conventional iterative algorithms and the joint design NN outperforms the conventional iterative algorithms. It can be seen that the sum rate of deep-unfolding NN degrades with the increase of K. It is mainly because as K increases, the problem turns more complex and it is difficult for NNs to find a satisfactory solution.

Table II and Table III indicate the sum rate versus the number of layers/iterations for deep-unfolding NN and conventional iterative algorithms. We normalize the results by the sum rate which is achieved by the conventional algorithm with

TABLE I: The sum rate versus different numbers of K.

K	1	2	3	4	5	6
RLS-SSCA (bits/s/Hz)	10.18	19.55	24.78	29.64	35.84	42.63
Separate Design NN	98.33%	97.23%	97.06%	96.77%	95.60%	95.33%
Joint Design NN	103.53%	102.32%	100.31%	100.07%	99.49%	98.78%

TABLE II: The sum rate versus the number of HBDUN/SSCA layers/iterations.

The number of layers of HBDUN	3	4	5	6	7	8	9
sum rate	85.14%	92.65%	101.25%	101.79%	101.79%	100.86%	101.02%
The number of layers of SSCA	30	35	40	45	50	55	60
sum rate	84.84%	89.44%	92.43%	96.99%	98.86%	100%	100%

TABLE III: The sum rate versus the number of CEDUN/RLS layers/iterations.

The number of layers of CEDUN	8	10	12	14	16	18	20
sum rate	92.18%	95.44%	98.15%	99.09%	101.25%	101.46%	101.5%
The number of layers of RLS	40	50	60	70	80	90	100
sum rate	79.64%	86.69%	91.85%	95.47%	100%	100%	100%

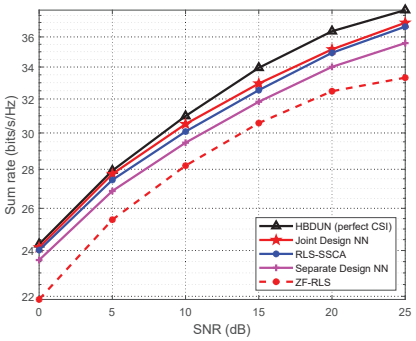

Fig. 3: The sum rate versus the SNR.

90 layers of the RLS algorithm and 70 layers of the SSCA algorithm. We see that the deep-unfolding NNs achieve better performance than the conventional algorithms with fewer layers, which illustrates that the computational complexity of the deep-unfolding NN is lower than that of conventional algorithms. With more layers, the sum rate achieved by the joint deep-unfolding NN improves and then fluctuates. This is mainly because when there are few layers, the degrees of freedom limit the learning capability. Thus, the sum rate improves with the number of layers. When the number of layers is large, the numerical error caused by matrix multiplication and inversion becomes large. Note that the number of layers of the CEDUN is the length of the pilots. Thus, our proposed deep-unfolding NN can save pilot resources compared to the RLS algorithm.

V. Conclusion

In this work, a mixed-timescale deep-unfolding based JCEHB framework has been proposed for hybrid massive MIMO systems. We developed a RLS algorithm induced deep-unfolding NN and a SSCA algorithm induced deep-unfolding NN for channel estimation and hybrid beamforming, respectively. Specifically, we introduced trainable parameters and non-linear operations to replace the high complexity operations and increase convergence speed. Simulation results showed that the proposed deep-unfolding algorithm can outperform conventional iterative algorithms. For future study, it is worth generating our deep-unfolding framework to solve more complex wireless systems around the research hotspots, such as multicell MIMO, drones and intelligent reflecting surface systems.

References

[1] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. Sel. Topics in Signal Process.*, vol. 8, no. 5, pp. 831–846, Jul. 2014.

[2] A. F. Molisch, V. V. Ratnam, S. Han, Z. Li, S. L. H. Nguyen, L. Li, and K. Haneda, "Hybrid beamforming for massive MIMO: A survey," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 134–141, Sep. 2017.

[3] S. S. Ioushua and Y. C. Eldar, "A family of hybrid analog-digital beamforming methods for massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 67, no. 12, pp. 3243–3257, Apr. 2019.

[4] X. Zhai, Y. Cai, Q. Shi, M. Zhao, G. Y. Li, and B. Champagne, "Joint transceiver design with antenna selection for large-scale MU-MIMO mmwave systems," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2085–2096, Jun. 2017.

[5] Z. Xiao, T. He, P. Xia, and X.-G. Xia, "Hierarchical codebook design for beamforming training in millimeter-wave communication," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3380–3392, May 2016.

[6] A. Liu, L. Lian, V. K. N. Lau, and X. Yuan, "Downlink channel estimation in multiuser massive MIMO with Hidden Markovian sparsity," *IEEE Trans. Signal Process.*, vol. 66, no. 18, pp. 4796–4810, Aug. 2018.

[7] Y. Cai, B. Champagne, and R. C. de Lamare, "Low-complexity adaptive transceiver techniques for k-pair MIMO interference channels," in *2013 IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2013, pp. 4071–4076.

[8] A. Liu, V. K. N. Lau, and M.-J. Zhao, "Online successive convex approximation for two-stage stochastic nonconvex optimization," *IEEE Trans. Signal Process.*, vol. 66, no. 22, pp. 5941–5955, Nov. 2018.

[9] Y. Cai, K. Xu, A. Liu, M. Zhao, B. Champagne, and L. Hanzo, "Two-timescale hybrid analog-digital beamforming for mmWave full-duplex MIMO multiple-relay aided systems," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 9, pp. 2086–2103, Sep. 2020.

[10] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Feb. 2021.

[11] Q. Hu, Y. Cai, Q. Shi, K. Xu, G. Yu, and Z. Ding, "Iterative algorithm induced deep-unfolding neural networks: Precoding design for multiuser MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1394–1410, Feb. 2021.

[12] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "Viterbinet: A deep learning based viterbi algorithm for symbol detection," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3319–3331, Feb. 2020.

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[14] J. Kaleva, A. Tlli, and M. Juntti, "Decentralized sum rate maximization with QoS constraints for interfering broadcast channel via successive convex approximation," *IEEE Trans. Signal Process.*, vol. 64, no. 11, pp. 2788–2802, Jun. 2016.

[15] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), TR 38.901, June 2018, version 14.0.0.