Mixed-Timescale Deep-Unfolding for Joint Channel Estimation and Hybrid Beamforming

Kai Kang⬥, Qiyu Hu, *Student Member, IEEE*, Yunlong Cai⬥, *Senior Member, IEEE*,
Guanding Yu⬥, *Senior Member, IEEE*, Jakob Hoydis⬥, *Senior Member, IEEE*,
and Yonina C. Eldar⬥, *Fellow, IEEE*

Abstract—In massive multiple-input multiple-output (MIMO) systems, hybrid analog-digital beamforming is an essential technique for exploiting the potential array gain without using a dedicated radio frequency chain for each antenna. However, due to the large number of antennas, the conventional channel estimation and hybrid beamforming algorithms generally require high computational complexity and signaling overhead. In this work, we propose an end-to-end deep-unfolding neural network (NN) joint channel estimation and hybrid beamforming (JCEHB) algorithm to maximize the system sum rate in time-division duplex (TDD) massive MIMO. Specifically, the recursive least-squares (RLS) algorithm and stochastic successive convex approximation (SSCA) algorithm are unfolded for channel estimation and hybrid beamforming, respectively. In order to reduce the signaling overhead, we consider a mixed-timescale hybrid beamforming scheme, where the analog beamforming matrices are optimized based on the channel state information (CSI) statistics offline, while the digital beamforming matrices are designed at each time slot based on the estimated low-dimensional equivalent CSI matrices. We jointly train the analog beamformers together with the trainable parameters of the RLS and SSCA induced deep-unfolding NNs based on the CSI statistics offline. During data transmission, we estimate the low-dimensional equivalent CSI by the RLS induced deep-unfolding NN and update the digital beamformers. In addition, we propose a mixed-timescale deep-unfolding NN where the analog beamformers are optimized online, and extend the framework to frequency-division duplex (FDD) systems where channel feedback is considered. Simulation results show that the proposed algorithm can significantly outperform conventional algorithms with reduced computational complexity and signaling overhead.

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

Kai Kang, Qiyu Hu, Yunlong Cai, and Guanding Yu are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: kangkai@zju.edu.cn; qiyhu@zju.edu.cn; ylcai@zju.edu.cn; yuguanding@zju.edu.cn).

Jakob Hoydis is with NVIDIA, 06906 Sophia Antipolis, France (e-mail: jhoydis@nvidia.com).

Yonina C. Eldar is with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 7610001, Israel (e-mail: yonina.eldar@weizmann.ac.il).

I. INTRODUCTION

THANKS to large-scale spatial multiplexing and highly directional beamforming, massive multiple-input multiple-output (MIMO) has been recognized as a pivotal technology for improving system reliability and data rate [1]–[5]. However, due to the exorbitant cost and energy consumption of radio frequency (RF) chains and analog-to-digital converters, the employment of conventional fully-digital beamforming is impractical with current technologies. Thus, hybrid analog-digital beamforming which requires a smaller number of RF chains has received great attention [6], [7]. There have been a number of algorithms proposed for hybrid beamforming and channel estimation in massive MIMO systems [8]–[19]. 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 of channel estimation and hybrid beamforming with low-complexity and reduced overhead. A number of previous algorithms have been proposed for hybrid beamforming in [8]–[14]. In [9], the authors proved that if the RF chain equipped with the hybrid beamforming structure is twice the total number of data streams, the performance approaches that of fully-digital beamforming. In [10] and [11], 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 [12]–[14]. In particular, a hierarchical codebook design for hybrid beamforming was proposed by [12] while a codebook-based RF precoding designed to maximize the spectral efficiency and energy efficiency simultaneously was designed in [13]. Channel estimation plays an important role in hybrid beamforming design [15]–[20]. The authors of [15] developed an effective algorithm that uses an hidden Markov model (HMM) for sparse channel estimation. In [16], the authors proposed a compressive sensing method for channel estimation by exploiting the spatial sparsity. A recursive least-squares (RLS) adaptive estimation algorithm was developed for MIMO interference channels in [19], which provides low computational complexity and can track the time-varying channels as the environment changes.

Conventional single-timescale hybrid beamformers are optimized based on the high-dimensional full channel state infor-

mation (CSI), which leads to large signaling overhead and transmission delay. To address these issues, several hybrid beamforming algorithms under the mixed-timescale scheme have been investigated in [21]–[23]. In this approach, long-term analog beamformers are optimized based on the channel statistics while the short-term digital beamformers are updated based on the reduced-dimensional CSI. However, these algorithms are challenging to implement in practice owing to the large number of iterations for convergence and high computational complexity operations, such as matrix inversion in each iteration.

In recent years, deep learning techniques have been widely applied in wireless communications such as channel estimation [24], signal detection [25]–[27], and CSI feedback in MIMO systems [28]. Compared to traditional algorithms, deep learning-based techniques have much lower computational complexity and often do not require CSI. In [29]–[32], the authors designed hybrid beamforming by employing convolutional neural networks (CNNs) and multi-layer perception (MLP) which are referred to as black-box neural networks (NNs). However, these NNs have poor interpretability and many samples are required for training. Deep-unfolding NNs have been recently receiving growing interest in various areas [33]. This approach unfolds iterative algorithms into layer-wise networks and introduces trainable parameters to improve system performance. Compared with black-box NNs, deep-unfolding NNs are more interpretable and require less training data, and have much lower computational complexity compared to traditional algorithms with comparable performance. Deep-unfolding NNs have been applied in communications [34], for example, resource allocation [35], [36], detection [37]–[40], channel estimation [41], [42], and transceiver design [43]–[46]. In [39], the authors proposed a symbol detector named ViterbiNet, which integrates black-box NNs into the Viterbi algorithm.

In prior works, deep-unfolding NNs are employed for a single module design. In this work, we propose an end-to-end deep-unfolding framework for joint channel estimation and hybrid beamforming (JCEHB) design in time-division duplex (TDD) massive MIMO systems to maximize the system sum rate. To reduce the signaling overhead, we employ the mixed-timescale hybrid beamforming scheme where analog beamformers are optimized offline. In addition, we extend the framework to other application scenarios.

The main contributions of this work are as follows.

- We propose an end-to-end mixed-timescale deep-unfolding framework for maximizing the system sum rate in massive MIMO, which jointly designs channel estimation and hybrid beamforming.
- We develop a RLS algorithm induced channel estimation deep-unfolding NN (CEDUN) and an SSCA algorithm induced 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.

- Under the mixed-timescale scheme, we develop a two-stage joint training method for the deep-unfolding NNs, where the analog beamformers are treated as trainable parameters and optimized offline based on the channel statistics. During the data transmission stage, we fix the analog beamformers and only estimate the low-dimensional equivalent CSI by the CEDUN to update the digital beamformers. When channel statistics change, we employ transfer learning to fine-tune the deep-unfolding NN.
- We extend our framework for the following different application scenarios: (i) An online mixed-timescale scheme where the analog beamformers are optimized in an online manner; (ii) Frequency-division duplex (FDD) system that incorporates channel quantization and feedback. We propose an end-to-end deep-learning based framework where deep-unfolding NNs are designed for channel estimation and hybrid beamforming and black-box NNs are designed for channel quantization and feedback, respectively.
- We provide detailed analysis of the performance and computational complexity of the proposed deep-unfolding algorithm. Simulation results show that our proposed deep-unfolding can significantly outperform conventional RLS and SSCA algorithms with reduced complexity.

The rest of the paper is structured as follows. Section II introduces the system model and problem formulation. Section III proposes the mixed-timescale deep-unfolding framework and presents the joint training method. Section IV develops the RLS induced deep-unfolding NN for channel estimation and Section V proposes the SSCA deep-unfolding NN for hybrid beamforming design. Section VI extends the deep-unfolding framework for different application scenarios. Section VII analyzes the computational complexity and performance of the proposed algorithm. Section VIII presents simulation results and conclusions are drawn in Section IX.

Throughout the paper we use the following notations. Scalars, vectors and matrices are respectively denoted by lower case, boldface lower case and boldface upper case letters; $\mathbf{I}$ represents an identity matrix and $\mathbf{0}$ denotes an all-zero matrix. For a matrix $\mathbf{A}$, $\mathbf{A}^T$, $\mathbf{A}^*$, $\mathbf{A}^H$, and $\|\mathbf{A}\|$ denote its transpose, conjugate, conjugate transpose and Frobenius norm, respectively. For a square matrix $\mathbf{A}$, $\text{Tr}\{\mathbf{A}\}$ is its trace. For a vector $\mathbf{a}$, $\|\mathbf{a}\|$ represents its Euclidean norm, $\mathbb{E}\{\cdot\}$ denotes the statistical expectation, and $\Re\{\cdot\}$ ($\Im\{\cdot\}$) are the real (imaginary) part of a variable. The operator $\text{vec}(\cdot)$ stacks the elements of a matrix in a column vector, $|\cdot|$ denotes the absolute value of a complex scalar, and $\mathbb{C}^{m \times n}$ ($\mathbb{R}^{m \times n}$) are the space of $m \times n$ complex (real) matrices.

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.
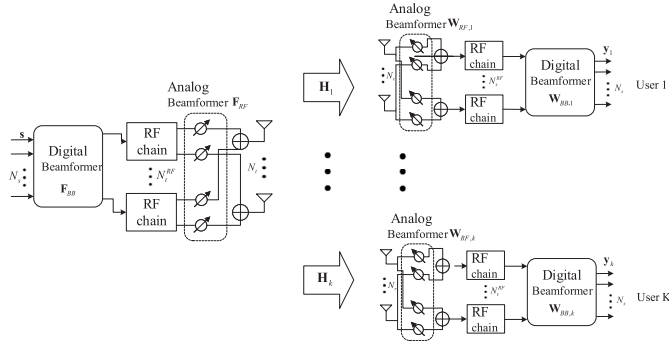
Fig. 1. Downlink massive MIMO system with hybrid beamforming.

A. System Model

1) Signal Model: Consider a downlink massive MIMO system working in TDD mode as shown in Fig. 1. 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 $\mathbf{s}_k \in \mathbb{C}^{N_s \times 1}$. Through the beamformers at the BS, the signal $\mathbf{u}_k \in \mathbb{C}^{N_t \times 1}$ for user k can be written as

$$\mathbf{u}_k = \sqrt{P}\mathbf{F}_{RF,k}\mathbf{F}_{BB,k}\mathbf{s}_k, \tag{1}$$

where P denotes the transmit power of the BS, $\mathbf{F}_{RF,k} \in \mathbb{C}^{N_t \times N_t^{RF}}$ is the analog beamformer which is subject to a unit modulus constraint, i.e., $|[\mathbf{F}_{RF}]_{i,j}| = \frac{1}{\sqrt{N_t}}, \forall i, j$, and $\mathbf{F}_{BB,k} \triangleq [\mathbf{f}_{BB,k,1}, \ldots, \mathbf{f}_{BB,k,N_s}] \in \mathbb{C}^{N_t^{RF} \times N_s}$ is the digital beamformer. 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.

After passing through the channel and the beamformers at the user, the received signal vector for user k is

$$\begin{aligned}\mathbf{y}_k &= \sqrt{P}\mathbf{W}_{BB,k}^H\mathbf{W}_{RF,k}^H\mathbf{H}_k\mathbf{F}_{RF,k}\mathbf{F}_{BB,k}\mathbf{s}_k + \sqrt{P}\mathbf{W}_{BB,k}^H \\ &\quad \times \mathbf{W}_{RF,k}^H\mathbf{H}_k \sum_{m=1,m\neq k}^{K} \mathbf{F}_{RF,m}\mathbf{F}_{BB,m}\mathbf{s}_m \\ &\quad + \mathbf{W}_{BB,k}^H\mathbf{W}_{RF,k}^H\mathbf{z}_k, \end{aligned} \tag{2}$$

where $\mathbf{H}_k \in \mathbb{C}^{N_r \times N_t}$ represents the channel matrix, $\mathbf{W}_{BB,k} \triangleq [\mathbf{w}_{BB,k,1}, \ldots, \mathbf{w}_{BB,k,N_s}] \in \mathbb{C}^{N_r^{RF} \times N_s}$ and $\mathbf{W}_{RF,k} \in \mathbb{C}^{N_r \times N_r^{RF}}$ denote the digital and analog beamformers for user k, respectively, and $\mathbf{z}_k \in \mathbb{C}^{N_r \times 1} \sim \mathcal{CN}(\mathbf{0}, \sigma_k^2\mathbf{I})$ is the additive white Gaussian noise (AWGN) with σ_k denoting the noise power. Similar to $\mathbf{F}_{RF}$, the analog combiner satisfies a unit modulus constraint $|[\mathbf{W}_{RF}]_{i,j}| = \frac{1}{\sqrt{N_r}}, \forall i, j$. Using (2), given perfect knowledge of the equivalent channel, the signal-to-interference-plus-noise

ratio (SINR) for stream l of user k is given as

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

$$\tag{3}$$

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

2) Channel Estimation: It is essential for the BS to obtain the CSI for hybrid beamforming. Here we consider estimation of the low-dimensional equivalent CSI. Thanks to 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 at the BS $\tilde{\mathbf{Y}}_{eq,k} \in \mathbb{C}^{N_r^{RF} \times L}$ is given by

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

$$\tag{4}$$

where $\tilde{\mathbf{Z}}_{eq,k}$ denotes the 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: $\|\tilde{\mathbf{x}}_{eq,k,l}\|^2 \leq P$. Then, 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 $\tilde{\mathbf{X}}_{eq,k}$, which can be expressed as

$$\hat{\mathbf{H}}_{eq,k} = \mathcal{F}(\tilde{\mathbf{Y}}_{eq,k}, \tilde{\mathbf{X}}_{eq,k}), \tag{5}$$

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

3) Hybrid Beamforming: After acquiring the channel information, the BS designs the hybrid beamformers based on the channel $\mathbf{H}_k$. The hybrid beamforming design scheme $\mathcal{Q}(\cdot)$ at the transmitter can be denoted as

$$\{\mathbf{F}_{RF,k}, \mathbf{W}_{RF,k}, \mathbf{F}_{BB,k}, \mathbf{W}_{BB,k}\} = \mathcal{Q}(\mathbf{H}_k), \quad \forall k. \tag{6}$$

B. Mixed-Timescale Frame Structure

Generally, the dimension of the channel matrix is high in massive MIMO systems due to the large number of antennas. It is therefore impractical to estimate each instantaneous CSI due to the unacceptable signaling overhead and high computational complexity. To address this problem, we consider a practical mixed-timescale frame structure as shown in Fig. 2, which takes into consideration both the instantaneous CSI and the channel statistics. We consider a superframe during which the channel statistics are constant. It consists of T_f frames, each of which is made of T_s time slots. The instantaneous CSI remains unchanged during each time slot. We introduce two different timescales as follows:

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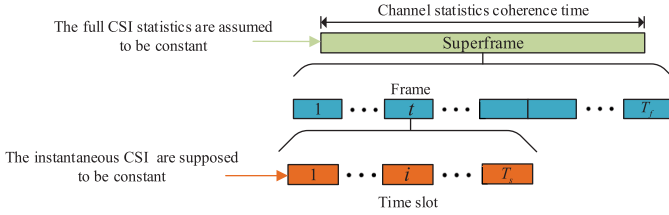- Short-timescale: The instantaneous CSI is constant in each time slot.

Fig. 2.   Mixed-timescale frame structure.

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 the dimension 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 can optimize the analog and digital beamformers at different timescales. We update the long-term analog beamformers $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ based on the long-term channel statistics when channel statistics [1] change, and 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 within each frame can be formulated as

$$\max_{\mathcal{X}, \mathcal{Y}} \quad \sum_{k=1}^{K} \sum_{l=1}^{N_s} \log(1 + \Gamma_{k,l}), \tag{7a}$$

$$\text{s.t.} \quad \| \, [\mathbf{F}_{RF,k}]_{ij} \, \|^2 = \frac{1}{\sqrt{N_t}}, \quad \forall k, i, j, \tag{7b}$$

$$\| \, [\mathbf{W}_{RF,k}]_{ij} \, \|^2 = \frac{1}{\sqrt{N_r}}, \quad \forall k, i, j, \tag{7c}$$

$$\| \, \mathbf{F}_{RF,k} \mathbf{F}_{BB,k}^i \, \|_F^2 = N_s, \quad \forall k, \tag{7d}$$

$$\| \tilde{\mathbf{x}}_{eq,k,l} \|^2 \leq P, \quad \forall k, l, \tag{7e}$$

$$\hat{\mathbf{H}}_{eq,k}^i = \mathcal{F}_{eq}(\tilde{\mathbf{Y}}_{eq,k}, \tilde{\mathbf{X}}_{eq,k}), \quad \forall k, \tag{7f}$$

$$\{\mathbf{F}_{RF,k}, \mathbf{W}_{RF,k}\} = \mathcal{Q}_{fu}(\mathbf{H}_k), \quad \forall k, \tag{7g}$$

$$\{\mathbf{F}_{BB,k}^i, \mathbf{W}_{BB,k}^i\} = \mathcal{Q}_{eq}(\hat{\mathbf{H}}_{eq,k}^i), \forall k, \tag{7h}$$

where $\mathcal{X} \triangleq \{\mathbf{F}_{RF,k}, \mathbf{W}_{RF,k}, \mathbf{F}_{BB,k}^i, \mathbf{W}_{BB,k}^i\}$ and $\mathcal{Y} \triangleq \{\tilde{\mathbf{X}}_{eq,k}\}$. In particular, $\mathbf{H}_k$ denotes long-term channel statistics, $\hat{\mathbf{H}}_{eq,k}^i$ denotes the estimated equivalent CSI, and $\{\mathbf{F}_{BB,k}^i, \mathbf{W}_{BB,k}^i\}$ represent the digital beamformers at the i-th time slot. In addition, $\mathcal{F}_{eq}(\cdot)$ denotes the estimation scheme for low-dimensional equivalent CSI, and $\mathcal{Q}_{fu}(\cdot)$ and $\mathcal{Q}_{eq}(\cdot)$ represent the analog and digital beamforming schemes, respectively. Constraints (7b) and (7c) reflect the unit modulus constraints for analog beamformers, and (7d) and (7e) represent the transmit power constraints. As we can see, the mixed-timescale problem is challenging to solve. In the following, a deep-unfolding framework is proposed for tackling this problem.

III. PROPOSED MIXED-TIMESCALE DEEP-UNFOLDING FRAMEWORK

In this section, we propose a mixed-timescale deep-unfolding framework, the structure of which is shown in Fig. 3. We first present how to jointly train the proposed deep-unfolding NNs offline and then show the whole process of the training and data transmission under the mixed-timescale scheme.

A. The Two-Stage Joint Training

During data transmission, the analog beamformers are fixed and only digital beamformers are updated based on the estimated equivalent channel at each time slot. To model the transmission process, we divide the training process for the deep-unfolding NN into two stages where the analog beamformers are optimized in the first stage and then fixed in the second stage.

1) The First Training Stage: Fig. 3(a) denotes the deep-unfolding NN for the first training stage, which presents the architecture of the HBDUN that consists of the analog NN and digital NN for designing the analog and digital beamforming, respectively. The input of the analog NN is the full CSI sample matrix $\mathbf{H}$ (possibly outdated) and the outputs are analog beamformers $\{\mathbf{W}_{RF}, \mathbf{F}_{RF}\}$. The input of the digital NN is the real-time low-dimensional equivalent CSI matrix $\mathbf{H}_{eq}$ and the outputs are digital beamformers $\{\mathbf{W}_{BB}, \mathbf{F}_{BB}\}$.

a) Forward propagation: We obtain channel samples offline based on the long-term channel statistics as the input of the analog NN and the outputs are analog beamformers $\{\mathbf{W}_{RF}, \mathbf{F}_{RF}\}$. Then we obtain the low-dimensional equivalent CSI matrices $\mathbf{H}_{eq}$ which pass through the digital NN that outputs the digital beamformers $\{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$. We introduce the detailed structure of the HBDUN in Section V-B. The forward propagation for the first stage $\mathcal{P}_1(\cdot)$ is expressed as

$$\{\mathbf{W}_{RF}, \mathbf{F}_{RF}, \mathbf{W}_{BB}, \mathbf{F}_{BB}\} = \mathcal{P}_1(\mathbf{\Upsilon}_B; \mathbf{H}), \tag{8}$$

where $\mathbf{\Upsilon}_B$ represents the trainable parameters of the HBDUN. Note that $\mathbf{\Upsilon}_B$ consists of $\mathbf{\Psi}$ and $\mathbf{\Omega}$, which represent the trainable parameters of the analog NN and digital NN, respectively.

b) Loss function: The loss function of the first stage is denoted as $\mathcal{L}_1(\mathbf{\Upsilon}_B; \mathbf{H})$, which is the system sum rate:

$$\mathcal{L}_1(\mathbf{\Upsilon}_B; \mathbf{H}) = -\sum_{k=1}^{K} \sum_{l=1}^{N_s} \log(1 + \Gamma_{k,l}). \tag{9}$$

c) Back propagation: All the trainable parameters of the HBDUN are updated based on the stochastic gradient descent (SGD) algorithm. Specifically, in the i-th round of the training process, we update the trainable parameters as follows:

$$\mathbf{\Upsilon}_B^{i+1} = \mathbf{\Upsilon}_B^i - \eta \frac{\partial \mathcal{L}_1(\mathbf{\Upsilon}_B; \mathbf{H})}{\partial \mathbf{\Upsilon}_B}, \tag{10}$$
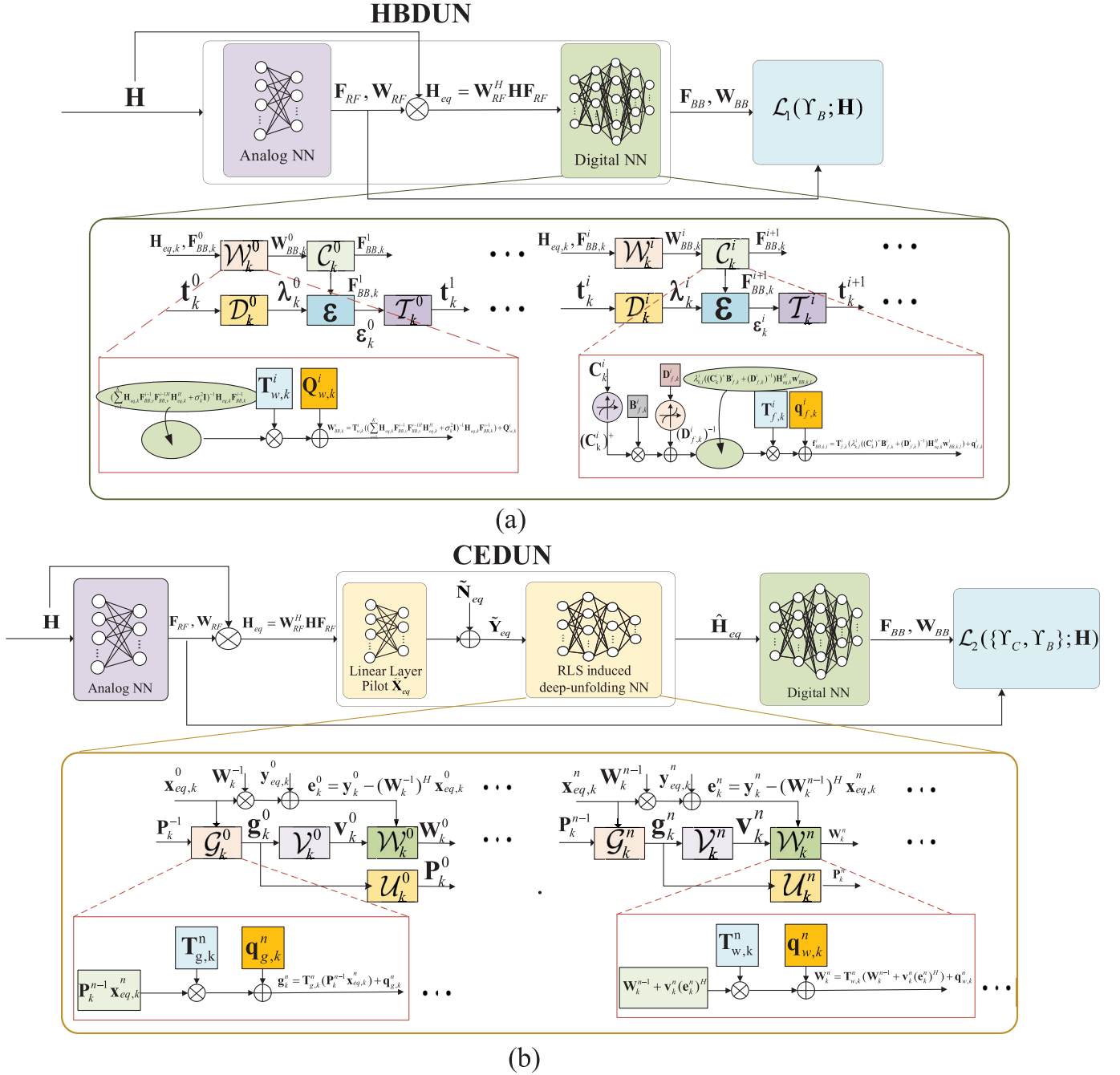
where η is the learning rate.

Fig. 3. Structure of the proposed deep-unfolding framework: (a) Deep-unfolding NN for the first training stage; (b) deep-unfolding NN for the second training stage.

2) The Second Training Stage: Fig. 3(b) shows the deep-unfolding NN for the second training stage which consists of the modules of channel estimation for low-dimensional equivalent CSI and digital beamforming. Note that the CEDUN represents the NN for equivalent CSI estimation, which consists of the pilot training NN and the RLS induced deep-unfolding NN. The analog and digital NN are employed to obtain analog and digital beamformers, respectively. The input of the CEDUN is the low-dimensional equivalent CSI matrix $\mathbf{H}_{eq}$ and the output is the estimated channel matrix $\hat{\mathbf{H}}_{eq}$.

a) Forward propagation: First, we fix the trained 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 pilot training NN and the RLS deep-unfolding NN which outputs the estimated equivalent CSI matrix $\hat{\mathbf{H}}_{eq}$. Finally, $\hat{\mathbf{H}}_{eq}$ passes through the digital NN that outputs the digital beamformers $\{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$. We introduce the detailed structure of the CEDUN in Section IV-B. The forward propagation for the second stage $\mathcal{P}_2(\cdot)$ is expressed as

$$\{\mathbf{W}_{RF}, \mathbf{F}_{RF}, \mathbf{W}_{BB}, \mathbf{F}_{BB}\} = \mathcal{P}_2(\{\mathbf{\Upsilon}_C, \mathbf{\Upsilon}_B\}; \mathbf{H}), \quad (11)$$

where $\mathbf{\Upsilon}_C$ consists of $\tilde{\mathbf{X}}_{eq}$ and $\mathbf{\Xi}$, which represent the trainable parameters of the pilot training NN and the RLS induced deep-unfolding NN, respectively.
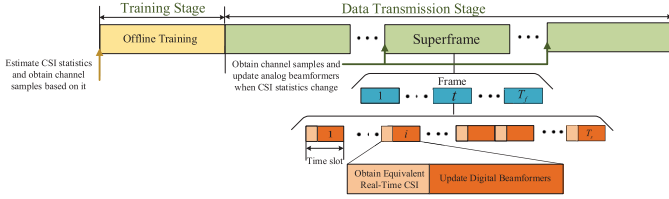
Fig. 4. Training stage and data transmission of the offline mixed-timescale scheme.

b) Loss function: The loss function of the second stage is denoted as $\mathcal{L}_2(\{\boldsymbol{\Upsilon}_C, \boldsymbol{\Upsilon}_B\}; \mathbf{H})$, which is the system sum rate.

$$\mathcal{L}_2(\{\boldsymbol{\Upsilon}_C, \boldsymbol{\Upsilon}_B\}; \mathbf{H}) = -\sum_{k=1}^{K}\sum_{l=1}^{N_s}\log(1 + \Gamma_{k,l}). \quad (12)$$

c) Back propagation: The trainable parameters of the CEDUN and digital NN are updated based on the SGD algorithm. Note that the parameters of the analog NN are not updated in the second training stage. Thus, in the i-th round of the training process, we update the trainable parameters as follows:

$$\boldsymbol{\Upsilon}_C^{i+1} = \boldsymbol{\Upsilon}_C^{i} - \eta\frac{\partial\mathcal{L}_2(\{\boldsymbol{\Upsilon}_C, \boldsymbol{\Upsilon}_B\}; \mathbf{H})}{\partial\boldsymbol{\Upsilon}_C}, \quad (13)$$

$$\boldsymbol{\Omega}^{i+1} = \boldsymbol{\Omega}^{i} - \eta\frac{\partial\mathcal{L}_2(\{\boldsymbol{\Upsilon}_C, \boldsymbol{\Upsilon}_B\}; \mathbf{H})}{\partial\boldsymbol{\Omega}}. \quad (14)$$

3) The Advantages of Joint Training: Compared with conventional separate designs, the proposed joint design framework has potential performance gains. Conventional algorithms optimize the channel estimation and hybrid beamforming modules separately under different objective functions, i.e., the minimization of MSE for channel estimation and maximization of sum rate for hybrid beamforming, respectively. In comparison, the proposed joint design NN ties the two modules tightly and jointly trains them. It provides an end-to-end deep-unfolding framework without explicitly estimating the CSI and both of the channel estimation and hybrid beamforming are designed to maximize the sum rate, which leads to better sum rate performance and is more efficient in terms of the signaling overhead.

B. The Training and the Data Transmission Stage

Training and data transmission for the mixed-timescale is shown in Fig. 4. We first obtain channel samples based on the long-term channel statistics offline and jointly train the deep-unfolding NN as mentioned above. Then in the data transmission stage, we fix the analog beamformers during the channel statistics coherence time and estimate the low-dimensional equivalent real-time CSI by CEDUN to update the digital beamformers. The analog beamformers obtained by offline training can well adapt to CSI statistics when it does not change [21]. When it changes, we obtain channel samples and employ transfer learning [47] to fine tune the parameters of the deep-unfolding NN, where the analog beamformers are updated to better fit the change of CSI statistics [48].

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**

IV. DEEP-UNFOLDING NETWORK FOR CHANNEL ESTIMATION

In this section, we first introduce a RLS based algorithm for estimating the low-dimensional equivalent CSI matrices and then describe the deep-unfolding NN by introducing trainable parameters.

A. The RLS Channel Estimation Algorithm

We focus on the expression (4), which represents the received pilot signal for user k. We aim to obtain the estimated low-dimensional equivalent channel matrix $\hat{\mathbf{H}}_{eq,k}$ by minimizing the mean square estimation error which is defined as $\mathbb{E}\{\|\hat{\mathbf{H}}_{eq,k} - \mathbf{H}_{eq,k}\|^2\}$. The solution of the least squares (LS) approach for solving this problem is given by

$$\hat{\mathbf{H}}_{eq,k} = \tilde{\mathbf{Y}}_{eq,k}\tilde{\mathbf{X}}_{eq,k}^H(\tilde{\mathbf{X}}_{eq,k}\tilde{\mathbf{X}}_{eq,k}^H)^{-1}. \quad (15)$$

In the RLS algorithm, the matrix inversion in (15) is replaced by an iterative process. 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 $\hat{\mathbf{X}}_{eq,k}$ 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.

B. 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. In the proposed offline mixed-timescale framework, we only need to estimate the low-dimensional equivalent CSI at each time slot. To estimate the equivalent channel $\mathbf{H}_{eq,k}$, the k-th user sends the training pilot matrix $\hat{\mathbf{X}}_{eq,k}$, and at the BS the received pilot signal matrix $\tilde{\mathbf{Y}}_{eq,k}$ is denoted as (4). Then $\tilde{\mathbf{X}}_{eq,k}$ and $\tilde{\mathbf{Y}}_{eq,k}$ are input to the RLS induced deep-unfolding NN to obtain the estimated equivalent channel $\hat{\mathbf{H}}_{eq,k}$. The structure of the CEDUN for user k is shown in Fig. 3(b).

1) Pilot Training NN: Different from the conventional Gaussian pilots and discrete fourier transform (DFT) pilots, in the proposed deep-unfolding NN, we set the pilots as trainable parameters that can adapt to the CSI statistics to further improve the performance. As shown in Fig. 3(b), 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 we set $\tilde{\mathbf{X}}_{eq}$ as the trainable parameter. Note that $\tilde{\mathbf{X}}_{eq}$ needs to be scaled to satisfy the power constraint (7e).

2) RLS Induced Deep-Unfolding NN: We unfold the RLS 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, we introduce the structure

$$\mathbf{Y}_{out}^n = \mathbf{T}_y^n \mathbf{X}_{in}^n + \mathbf{q}_y^n, \tag{16}$$

where $\mathbf{X}_{in}^n$ and $\mathbf{Y}_{out}^n$ represent the input and output of the n-th layer, respectively, $\mathbf{T}_y^n$ and $\mathbf{q}_y^n$ are the introduced multiplier and offset trainable parameter of the n-th layer, respectively.

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. 3(b), based on Algorithm 1, $\mathcal{G}_k^n$, $\mathcal{V}_k^n$, $\mathcal{U}_k^n$, $\mathcal{W}_k^n$ represent the sub-layers of the n-th layer of the deep-unfolding NN, i.e., (17a)-(17d).

$$\mathbf{g}_k^n = \mathbf{T}_{g,k}^n (\mathbf{P}_k^{n-1} \tilde{\mathbf{x}}_{eq,k}^n) + \mathbf{q}_{g,k}^n, \tag{17a}$$

$$\mathbf{v}_k^n = \mathbf{T}_{v,k}^n \left(\frac{\mathbf{g}_k^n}{\gamma_k^n + (\mathbf{g}_k^n)^H \tilde{\mathbf{x}}_{eq,k}^n} \right) + \mathbf{q}_{v,k}^n, \tag{17b}$$

$$\mathbf{P}_k^n = \mathbf{T}_{p,k}^n (\gamma_k^n)^{-1} (\mathbf{P}_k^{n-1} - \mathbf{v}_k^n (\mathbf{g}_k^n)^H) + \mathbf{q}_{p,k}^n, \tag{17c}$$

$$\mathbf{W}_k^n = \mathbf{T}_{w,k}^n (\mathbf{W}_k^{n-1} + \mathbf{v}_k^n (\mathbf{e}_k^n)^H) + \mathbf{q}_{w,k}^n. \tag{17d}$$

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

V. DEEP-UNFOLDING NETWORK FOR HYBRID BEAMFORMING

We next turn to design an SSCA algorithm for hybrid beamforming and then introduce the proposed HBDUN which unfolds the SSCA algorithm.

A. The SSCA-Based Hybrid Beamforming Algorithm

For the hybrid beamforming design, we first introduce the mixed-timescale SSCA framework [21]. Let us express the element of analog beamformers in terms of $e^{j\phi}$ as

$$\mathbf{F}_{RF} = e^{j\boldsymbol{\phi}_F}, \boldsymbol{\phi}_F \in \mathbb{C}^{N_t \times N_t^{RF}}, \tag{18a}$$

$$\mathbf{W}_{RF} = e^{j\boldsymbol{\phi}_W}, \boldsymbol{\phi}_W \in \mathbb{C}^{N_r \times N_r^{RF}}, \tag{18b}$$

where $\boldsymbol{\phi} \triangleq \{\boldsymbol{\phi}_F, \boldsymbol{\phi}_W\}$ denotes the angle of phase shifter. Here, the objective function can be expressed as $r_0(\boldsymbol{\phi}, \mathbf{M}; \mathbf{H})$, where $\mathbf{M} \triangleq \{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$.

Algorithm 2 The SCA Algorithm for Digital Beamforming

1: Input: The equivalent channel $\mathbf{H}_{eq}$.
2: Initialize $\mathbf{t}_{k,l} = 0$ and digital precoder $\mathbf{F}_{BB}$ satisfying the power constraint condition and set the maximum iteration number I_{max};
3: **for** $i = 1, 2, 3, \ldots$ **do**
4:    Update digital combiner $\mathbf{W}_{BB,k}^i$ based on (24);
5:    Update dual variable $\lambda_{k,l}^i = \frac{\alpha^{\mathbf{t}_{k,l}^i}}{\log \alpha}$;
6:    Update digital precoder

$$\mathbf{f}_{BB,k,l}^i = \lambda_{k,l}^i \left(\sum_{(m,n)}^{(K,N_s)} \lambda_{m,n}^i \mathbf{H}_{eq,m}^H \mathbf{w}_{BB,m,n}^i \right.$$

$$\left. \times (\mathbf{w}_{BB,m,n}^i)^H \mathbf{H}_{eq,m} + \tau_k \mathbf{I} \right)^{-1} \mathbf{H}_{eq,k}^H \mathbf{w}_{BB,k,l}^i;$$

7:    Calculate MSE $\epsilon_{k,l}^i$ based on (23);
8:    Update $\mathbf{t}_{k,l}^{i+1} = \mathbf{t}_{k,l}^i + \frac{1}{\log \alpha}(1 - \epsilon_{k,l}^i \alpha^{\mathbf{t}_{k,l}^i})$;
9: **Until** desired level of convergence or $i > I_{max}$.
10: **end for**

---

1) Long-Term Analog Beamformers: We optimize the analog beamformers based on the full CSI samples $\mathbf{H}$. First, we fix the digital beamformers $\mathbf{M}^J$, which means that the digital beamformers are obtained by running the algorithm that optimizes the short-term variables for J iterations. Then we employ a convex surrogate function to replace the objective function to optimize the analog beamformers. The surrogate function in the t-th iteration is [21]

$$\bar{f}^t(\boldsymbol{\phi}) = f^t + (\boldsymbol{f}_\phi^t)^T (\boldsymbol{\phi} - \boldsymbol{\phi}^t) + \tau \parallel \boldsymbol{\phi} - \boldsymbol{\phi}^t \parallel^2, \tag{19}$$

where

$$\boldsymbol{f}_\phi^t = (1 - \rho^t) \boldsymbol{f}_\phi^{t-1} + \rho^t \nabla_\phi r_0(\boldsymbol{\phi}^t, \mathbf{M}^J; \mathbf{H}) \tag{20}$$

with $\boldsymbol{f}_\phi^{-1} = \mathbf{0}$. Here $\nabla_\phi r_0(\boldsymbol{\phi}^t, \mathbf{M}^J; \mathbf{H})$ is the partial derivative of $r_0(\boldsymbol{\phi}^t, \mathbf{M}^J; \mathbf{H})$, and the constant f^t is calculated as

$$f^t = (1 - \rho^t) f^{t-1} + \rho^t r_0(\boldsymbol{\phi}^t, \mathbf{M}^J; \mathbf{H}) \tag{21}$$

with $f^{-1} = 0$. We take the derivative with formula (19) to update the long-term variable $\boldsymbol{\phi}$ as

$$\bar{\boldsymbol{\phi}} = \boldsymbol{\phi} - \eta \boldsymbol{f}_\phi^t, \tag{22}$$

where η denotes the step size.

2) Short-Term Digital Beamformers: 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 [49]. For the data stream l of user k, the MSE expression is given as

$$\epsilon_{k,l} = |1 - \mathbf{w}_{BB,k,l}^H \mathbf{H}_{eq,k} \mathbf{f}_{BB,k,l}|^2$$

$$+ \sum_{m,n(m \neq k, n \neq l)}^{K,N_s} |\mathbf{w}_{BB,k,l}^H \mathbf{H}_{eq,k} \mathbf{f}_{BB,m,n}|^2$$

$$+ \sigma_k^2 \|\mathbf{w}_{BB,k,l}\|^2. \tag{23}$$

The optimal digital beamformer is given by

$$\mathbf{W}_{BB,k} = (\sum_{v=1}^{K} \mathbf{H}_{eq,k}\mathbf{F}_{BB,v}\mathbf{F}_{BB,v}^{H}\mathbf{H}_{eq,k}^{H} + \sigma_k^2\mathbf{I})^{-1}$$
$$\times \mathbf{H}_{eq,k}\mathbf{F}_{BB,k}. \quad (24)$$

We equivalently transform the optimization objective function into the problem that minimizes the MSE [49]:

$$\min_{\mathbf{F}_{BB}} \sum_{k=1}^{K} \sum_{l=1}^{N_s} \log(\epsilon_{k,l}). \quad (25)$$

Then we design the digital precoder $\mathbf{F}_{BB}$. By introducing a monotonic log-concave function $g(\mathbf{t}_{k,l})$, the target optimization function becomes [49]

$$\min_{\mathbf{F}_{BB},\mathbf{t}_{k,l}} \sum_{k=1}^{K} \sum_{l=1}^{N_s} \log(g(\mathbf{t}_{k,l})^{-1}), \quad (26a)$$

$$\text{s.t.} \quad \epsilon_{k,l} \leq g(\mathbf{t}_{k,l})^{-1}. \quad (26b)$$

We then use the first-order Taylor approximation

$$\bar{g}(\mathbf{t}_{k,l}, \mathbf{t}_{k,l}^0) = g(\mathbf{t}_{k,l}^0) + (\mathbf{t}_{k,l} - \mathbf{t}_{k,l}^0)\frac{\partial}{\partial \mathbf{t}_{k,l}}g(\mathbf{t}_{k,l}^0). \quad (27)$$

Problem (26) can be formulated as

$$\min_{\mathbf{F}_{BB},\mathbf{t}_{k,l}} \sum_{k=1}^{K} \sum_{l=1}^{N_s} \log(\bar{g}(\mathbf{t}_{k,l}, \mathbf{t}_{k,l}^{(i)})), \quad (28a)$$

$$\text{s.t.} \quad \epsilon_{k,l} \leq [\bar{g}(\mathbf{t}_{k,l}, \mathbf{t}_{k,l}^{(i)})]. \quad (28b)$$

To simplify the problem, we consider a log-linear function $g(x) = \alpha^x$, where $\alpha > 1$. We present the SCA algorithm in Algorithm 2, where τ_k denotes the Lagrange multiplier.

B. The SSCA Algorithm Induced Deep-Unfolding NN

We next unfold the SSCA algorithm leading to HBDUN. The structure of HBDUN is shown in Fig. 3(a), where the first network and the second network are referred to as the analog and digital NNs, respectively.

The input of the analog NN is the full channel samples $\mathbf{H}$ and the outputs are the analog beamforming matrix $\{\mathbf{F}_{RF,k}, \mathbf{W}_{RF,k}\}$. 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: (i) We take the inverse of the diagonal entries of matrix $\mathbf{A}$ and set the other elements in $\mathbf{A}$ as zero, and denote the result as $\mathbf{A}^+$. For example,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad \mathbf{A}^+ = \begin{bmatrix} a_{11}^{-1} & 0 & 0 \\ 0 & a_{22}^{-1} & 0 \\ 0 & 0 & a_{33}^{-1} \end{bmatrix}; \quad (29)$$

(ii) we set the imaginary part of the diagonal elements of $\mathbf{D}$ to zero, expressed as $\mathbf{D}^-$:

$$\mathbf{D} = \begin{bmatrix} d_{r,11} + jd_{i,11} & d_{r,12} + jd_{i,12} & d_{r,13} + jd_{i,13} \\ d_{r,21} + jd_{i,21} & d_{r,22} + jd_{i,22} & d_{r,23} + jd_{i,23} \\ d_{r,31} + jd_{i,31} & d_{r,32} + jd_{i,32} & d_{r,33} + jd_{i,33} \end{bmatrix},$$

$$\mathbf{D}^- = \begin{bmatrix} d_{r,11} & d_{r,12} + jd_{i,12} & d_{r,13} + jd_{i,13} \\ d_{r,21} + jd_{i,21} & d_{r,22} & d_{r,23} + jd_{i,23} \\ d_{r,31} + jd_{i,31} & d_{r,32} + jd_{i,32} & d_{r,33} \end{bmatrix}. \quad (30)$$

Based on this, we employ the following structure to approximate the matrix inversion.

- First, we use $\mathbf{A}^+\mathbf{B}$ with non-linear operation $\mathbf{A}^+$ and trainable parameter $\mathbf{B}$, where $\mathbf{B}$ is introduced to improve performance. It can be seen that $\mathbf{A}^{-1} = \mathbf{A}^+$ if matrix $\mathbf{A}$ is diagonal. We observe that the diagonal elements of the matrix are much larger than the other elements in the SCA algorithm.
- Secondly, we introduce the offset trainable matrix $\mathbf{D}$ to better approximate the inverse matrix. We find that the imaginary part of the diagonal elements of the inverse matrix are close to zero. Thus, we employ $\mathbf{D}^-$ as the offset.

Thus, the matrix inversion $\mathbf{A}^{-1}$ is approximated by $\mathbf{A}^+\mathbf{B} + \mathbf{D}^-$. The computational complexity of matrix inversion is $\mathcal{O}(n^3)$ while that of the approximation is $\mathcal{O}(n^{2.37})$. Note that we introduce trainable parameters $\{\mathbf{B}_{f,k}^i, \mathbf{D}_{f,k}^i\}$ to approximate the inversion of variable $\mathbf{f}_{BB,k,l}^i$ 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 $\varpi_B^i \triangleq \{\mathbf{T}_{w,k}^i, \mathbf{Q}_{w,k}^i\} \cup \{\mathbf{T}_{\lambda,k}^i, \mathbf{q}_{\lambda,k}^i\} \cup \{\mathbf{T}_{f,k}^i, \mathbf{q}_{f,k}^i\} \cup \{\mathbf{T}_{t,k}^i, \mathbf{q}_{t,k}^i\}$ are introduced in updating the variables $\mathbf{W}_{BB,k}^i$, $\lambda_{k,l}^i$, $\mathbf{f}_{BB,k,l}^i$, and $\mathbf{t}_{k,l}^i$ in the i-th layer, respectively. As shown in Fig. 3(a), based on Algorithm 2, $\mathcal{W}_k^i$, $\mathcal{D}_k^i$, $\mathcal{C}_k^i$, and $\mathcal{T}_k^i$ represent the sub-layers of the i-th layer of the deep-unfolding NN, i.e., (31a)-(31d), where

$$\mathbf{C}_k^i \triangleq \sum_{(m,n)}^{(K,N_s)} \lambda_{m,n}^i \mathbf{H}_{eq,m}^H \mathbf{w}_{BB,m,n}^i (\mathbf{w}_{BB,m,n}^i)^H \mathbf{H}_{eq,m} + v_k^i\mathbf{I}.$$

In addition, the constant $\frac{1}{\log \alpha}$ is set as trainable parameter μ_k^i to speed up convergence.

$$\mathbf{W}_{BB,k}^i = \mathbf{T}_{w,k}^i\Big((\sum_{v=1}^{K} \mathbf{H}_{eq,k}\mathbf{F}_{BB,v}^{i-1}\mathbf{F}_{BB,v}^{i-1H}\mathbf{H}_{eq,k}^H + \sigma_k^2\mathbf{I})^{-1}$$
$$\times \mathbf{H}_{eq,k}\mathbf{F}_{BB,k}^{i-1}\Big) + \mathbf{Q}_{w,k}^i, \quad (31a)$$

$$\lambda_{k,l}^i = \mathbf{T}_{\lambda,k}^i(\mu_k^i\alpha^{t_{k,l}^i}) + \mathbf{q}_{\lambda,k}^i, \quad (31b)$$

$$\mathbf{f}_{BB,k,l}^i = \mathbf{T}_{f,k}^i\Big(\lambda_{k,l}^i((\mathbf{C}_k^i)^+\mathbf{B}_{f,k}^i + (\mathbf{D}_{f,k}^i)^-)\mathbf{H}_{eq,k}^H\mathbf{w}_{BB,k,l}^i\Big)$$
$$+ \mathbf{q}_{f,k}^i, \quad (31c)$$

$$\mathbf{t}_{k,l}^i = \mathbf{T}_{t,k}^i(\mathbf{t}_{k,l}^{i-1} + \mu_k^i(1 - \epsilon_{k,l}\alpha^{t_{k,l}^{i-1}})) + \mathbf{q}_{t,k}^i. \quad (31d)$$

Thus, the trainable parameters of the digital NN are $\mathbf{\Omega} \triangleq \bigcup_{i=1}^{L_h}\{\mathbf{B}_{f,k}^i, \mathbf{D}_{f,k}^i\} \cup \varpi_B^i \cup \mu_k^i$, where L_h is the number of layers of the digital NN. All the trainable parameters of the HBDUN are denoted as $\mathbf{\Upsilon}_B \triangleq \mathbf{\Psi} \cup \mathbf{\Omega}$. In addition, to avoid
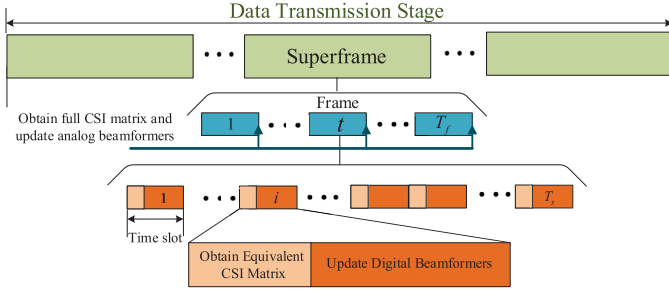
Fig. 5. Frame structure of the online mixed-timescale scheme.

gradient explosion and satisfy the power constraint, we normalize $\mathbf{F}^i_{BB,k}$ by N_s at each layer to $\frac{\sqrt{N_s}}{\|\mathbf{F}_{RF,k}\mathbf{F}^i_{BB,k}\|_F}\mathbf{F}^i_{BB,k}$.

It is interesting to investigate the relationship of the analog beamformers in the proposed deep-unfolding algorithm and the conventional SSCA-based algorithm. In SSCA, ϕ is updated based on the gradient $\frac{\partial r_0((\phi, \boldsymbol{M}); \mathbf{H})}{\partial \phi}|_{\phi=\phi^i}$ while $\boldsymbol{\Psi}$ is updated in the HBDUN based on the gradient

$$
\begin{aligned}
\frac{\partial \mathcal{L}_1(\boldsymbol{\Upsilon}_B; \mathbf{H})}{\partial \boldsymbol{\Psi}}|_{\boldsymbol{\Psi}=\boldsymbol{\Psi}^i} &= \frac{\partial r_0((\boldsymbol{\Psi}, \mathcal{P}_1(\{\boldsymbol{\Psi}, \boldsymbol{\Omega}\}; \mathbf{H})); \mathbf{H})}{\partial \boldsymbol{\Psi}}|_{\boldsymbol{\Psi}=\boldsymbol{\Psi}^i} \\
&= \frac{\partial r_0((\boldsymbol{\Psi}, \mathcal{P}_1(\{\boldsymbol{\Psi}^i, \boldsymbol{\Omega}\}; \mathbf{H})); \mathbf{H})}{\partial \boldsymbol{\Psi}}|_{\boldsymbol{\Psi}=\boldsymbol{\Psi}^i} \\
&+ (\frac{\partial r_0}{\partial \mathcal{P}_1})^T \frac{\partial \mathcal{P}_1(\{\boldsymbol{\Psi}^i, \boldsymbol{\Omega}\}; \mathbf{H})}{\partial \boldsymbol{\Psi}}|_{\boldsymbol{\Psi}=\boldsymbol{\Psi}^i}.
\end{aligned}
\tag{32}
$$

The gradient of the SSCA algorithm is the same as the first term in (32) except that the hybrid beamformers are acquired by the HBDUN. The second term is the gradient of the NN which only exists in the deep-unfolding NN but not in the SSCA algorithm. This term further ties the analog and digital beamformers.

VI. EXTENSIONS OF THE MIXED-TIMESCALE DEEP-UNFOLDING FRAMEWORK

In this section, we extend the framework proposed in Section III for other scenarios. In particular, we propose a mixed-timescale deep-unfolding framework where the analog beamformers are optimized online. Besides, we propose an end-to-end deep-learning based framework and employ it in the FDD system, where channel quantization and feedback are considered in the design.

A. Online Mixed-Timescale Deep-Unfolding Framework

1) Scenarios for Offline and Online Optimization: It is appropriate for optimizing analog beamformers offline in scenarios where CSI characteristics are known and a number of channel samples can be collected before data transmission stage. Besides, when channel statistics change, we need to collect channel samples to fine-tune the deep-unfolding NN. During the time it takes to collect channel samples, the analog beamformers are not optimized. Thus, if it is difficult to collect channel samples, the offline optimization is not appropriate. If we do not know the CSI statistics or it is difficult to collect a large number of channel samples, the analog beamformers

need to be updated online. In this way, we collect channel samples and optimize analog beamformers at the same time during data transmission. The performance gradually improves as the number of collected samples increases and eventually converges when there are enough channel samples.

2) Online Mixed-Timescale Scheme: In the online mixed-timescale scheme, we optimize long-term analog beamformers and short-term digital beamformers at different timescales. The frame structure is shown in Fig. 5. In particular, at the end of each frame, we obtain full CSI samples and optimize analog beamformers using the long-term deep-unfolding NN while at each time slot, we obtain equivalent CSI matrices to optimize the digital beamformers using the short-term deep-unfolding NN.

3) Structure of the Online Deep-Unfolding Framework: We propose a mixed-timescale deep-unfolding framework for online training, the structure of which is shown in Fig. 6. At the end of each frame, we train the long-term deep-unfolding NN, as shown in Fig. 6(a), to optimize the analog beamformers. The long-term deep-unfolding NN consists of long-term CEDUN for full CSI estimaiton and HBDUN for hybrid beamforming. At each time slot, we train the short-term deep-unfolding NN shown in Fig. 6(b) to optimize the digital beamformers. The training process is similar to that in Section III-A.

B. End-to-End Deep-Learning Framework for FDD Systems

1) Channel Estimation and Feedback for FDD Systems: In FDD systems, there is no channel reciprocity. To acquire downlink CSI, the BS needs to send pilots to the users. Then the users estimate the CSI matrix $\mathbf{H}$ which is quantized as bits and fed back to the BS. The BS recovers the CSI matrix for hybrid beamforming based on the feedback bits. During the data transmission in the mixed-timescale scheme, we fix the analog beamformers and only estimate the equivalent CSI at each time slot. Thus, we only need to feed the equivalent CSI back to the BS.

2) Structure of the Deep-Learning Based Framework for FDD Systems: In an end-to-end FDD system, the modules of channel estimation and hybrid beamforming are required to have strong generalization and interpretability, which can be designed by deep-unfolding approaches. However, the module of channel feedback, which is used to extract and compress channel features and recover the channel, does not need to have strong interpretability and generalization for signal-to-noise ratio (SNR). Black-box NNs can effectively extract channel features and reduce feedback overhead [28], which are suitable for channel feedback design. Besides, it is difficult to unfold the conventional algorithm [16], [50], [51] for channel quantization and feedback due to the non-differentiable procedures. Thus, we propose an autoencoder based on the CRNet in [52] which significantly compresses the channel matrix and reduces the transmission overhead. The structure of the feedback autoencoder is shown in Fig. 7, and consists of several convolution layers and fully connected (FC) layers. The dimension of the equivalent CSI matrix is much lower than that of full CSI matrix and less information needs to be fed back.
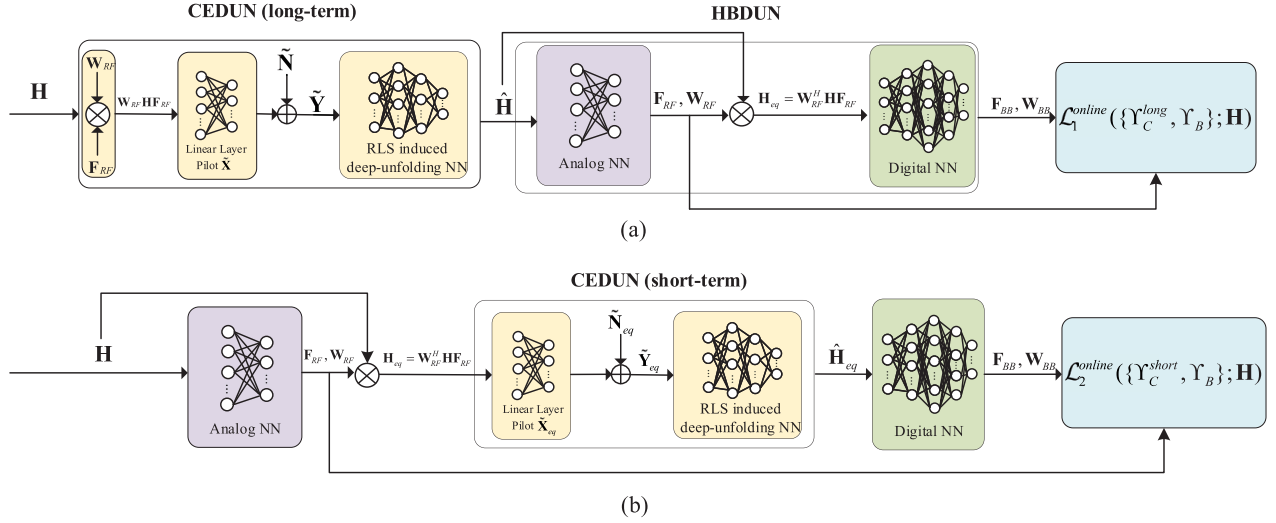
Fig. 6. Structure of the proposed online deep-unfolding framework: (a) Long-term deep-unfolding NN; (b) short-term deep-unfolding NN.
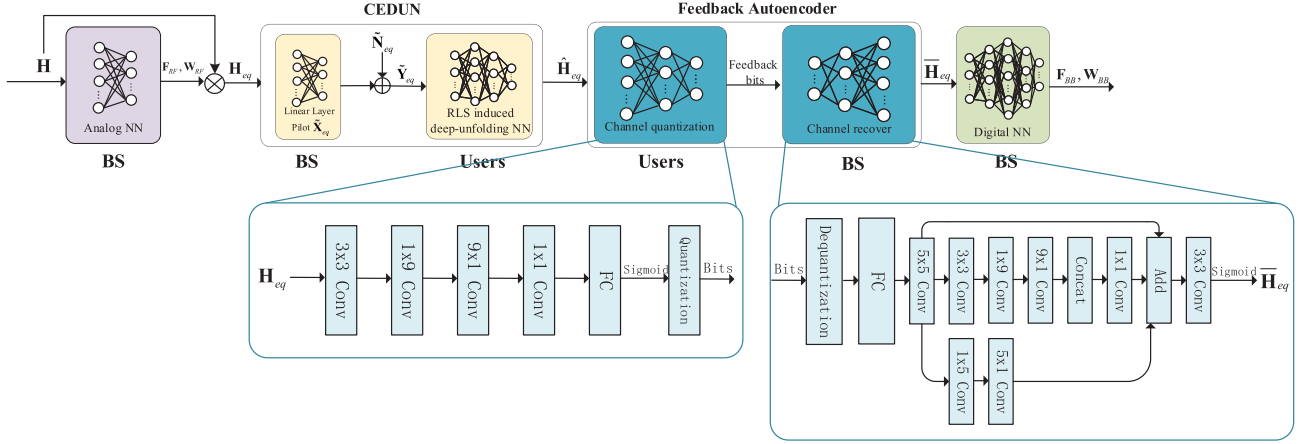

Fig. 7. Architecture of the deep-unfolding framework for FDD systems.

We jointly design the feedback autoencoder and deep-unfolding NNs in the proposed deep-learning framework. Here the RLS induced deep-unfolding NN and the channel quantization module are deployed at the users while the pilot training NN, channel recovery module, and HBDUN are deployed at the BS.

The performance of the autoencoder only depends on the equivalent CSI statistics but not on the transmit power or the channel noise. Thus, the autoencoder is not required to have a strong generalization ability for SNR and can be trained separately. The training process is thus as follows.

- **Training Stage I:** First we jointly train the deep-unfolding NN except for the autoencoder according to Section III.A.
- **Training Stage II:** Then we separately train the autoencoder for channel quantization and recovery. We fix the analog beamformers and obtain the equivalent CSI matrix $\mathbf{H}_{eq}$, which is the input of the autoencoder. Its output is the recovered CSI matrix $\bar{\mathbf{H}}_{eq}$. The loss function is normalized mean square error (NMSE), which is defined as follows:

$$\text{NMSE} = \frac{\parallel \bar{\mathbf{H}}_{eq} - \mathbf{H}_{eq} \parallel_2^2}{\parallel \mathbf{H}_{eq} \parallel_2^2}. \tag{33}$$

- **Training Stage III:** Finally, we load the well-trained models into the deep-unfolding NN and the autoencoder and cascade them as shown in Fig. 7. Then we jointly train them with the sum rate as the loss function.

VII. COMPUTATIONAL COMPLEXITY AND PERFORMANCE ANALYSIS

In this section, we first develop a black-box NN for the JCEHB design for comparison. Then we analyze the computational complexity of the conventional algorithms, the proposed deep-unfolding NN and the black-box NN. We further analyze the convergence of the proposed deep-unfolding NN based algorithm.

A. The Black-Box NNs for JCEHB Design

The structure of the black-box NN is shown in Fig. 8. For the analog beamforming design, we propose the analog black-box NN which sets the phase of analog beamformers as trainable parameters similar to the proposed deep-unfolding NN. For channel estimation and digital beamforming design, we propose the channel estimation and digital black-box NN, which adopt the DNN that consists of conventional FC layers and batch normalization (BN) layers. ReLU is used
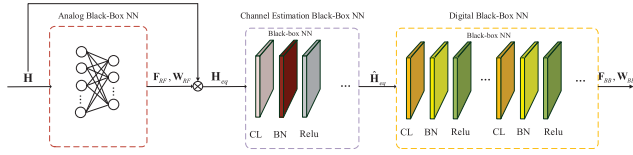
Fig. 8.   Architecture of the black-box NN.

as the activation function. Specifically, the full CSI sample $\mathbf{H}_k$ passes through the analog black-box NN and analog beamformers $\{\mathbf{F}_{RF,k}, \mathbf{W}_{RF,k}\}$ are obtained. Then we obtain the equivalent CSI matrix $\mathbf{H}_{eq,k}$ which passes through the channel estimation black-box NN that outputs the estimated equivalent channel matrix $\hat{\mathbf{H}}_{eq,k}$. Finally, $\hat{\mathbf{H}}_{eq,k}$ passes through the digital black-box NN which outputs the digital beamformers $\{\mathbf{F}_{BB,k}, \mathbf{W}_{BB,k}\}$. The input of the black-box NN is the full CSI matrix and the outputs are hybrid beamformers. The loss function of the black-box NN is the system sum rate.

B. Computational Complexity

We analyze the computational complexity of the conventional RLS and SSCA algorithm, the proposed deep-unfolding NN, and the black-box NN.

1) Conventional Algorithms: The computational complexity of the RLS algorithm is

$$\mathcal{O}(L_r K (N_t^{RF})^2), \tag{34}$$

where L_r is the number of iterations. The computational complexity of the SSCA algorithm is

$$\mathcal{O}(L_s(KN_s(N_t^{RF})^3 + K^2 N_s^2 (N_t^{RF})^2 N_r^{RF} \\ + K N_s (N_t^{RF})^2 N_r^{RF})), \tag{35}$$

where L_s is the number of iterations.

2) Proposed Deep-Unfolding NN: The computational complexity of the CEDUN is

$$\mathcal{O}(L_c K (N_t^{RF})^2), \tag{36}$$

where L_c ($L_c \ll L_r$) represents the number of layers of the CEDUN. The computational complexity of the HBDUN is

$$\mathcal{O}(L_h(KN_s(N_t^{RF})^{2.37} + K^2 N_s^2 (N_t^{RF})^2 N_r^{RF} \\ + K N_s (N_t^{RF})^2 N_r^{RF})), \tag{37}$$

where L_h ($L_h \ll L_a$) represents the number of layers of the HBDUN. Compared to the traditional algorithms, the proposed deep-unfolding NN has a lower computational complexity:

- The number of layers of the proposed deep-unfolding NN is reduced, i.e., $L_c \ll L_r, L_h \ll L_s$.
- The HBDUN approximates the matrix inversion operation with computational complexity $\mathcal{O}(n^3)$ by matrix multiplication operation with lower computational complexity $\mathcal{O}(n^{2.37})$.

3) Black-Box NN: The computational complexity of the proposed black-box NN is

$$\mathcal{O}\left(\sum_{l=1}^{L_b-1} KF_{c,l}F_{c,l+1} + \sum_{l=1}^{L_d-1} KF_{h,l}F_{h,l+1}\right), \tag{38}$$

where L_b and L_d denote the numbers for fully connected layers of channel estimation and digital black-box NN, respectively, and $F_{c,l}$ and $F_{h,l}$ denote the output sizes of the l-th layer for channel estimation and hybrid beamforming black-box NN, respectively.

C. Performance Analysis for the Deep-Unfolding NN

We show that the performance of one layer of the deep-unfolding NN can approach that of several iterations of the conventional iterative algorithm. We take the HBDUN as an example; the CEDUN can be analyzed in the same way.

We consider the update of $\mathbf{t}$ as an example, where $\mathbf{W}_{BB}$ and $\mathbf{F}_{BB}$ are treated as constants. As shown in Algorithm 2, in the i-th iteration of the SCA algorithm, we have the following mapping from $\mathbf{t}_{k,l}^i$ to $\mathbf{t}_{k,l}^{i+2}$:

$$\mathbf{t}_{k,l}^{i+2} = \mathbf{t}_{k,l}^i + \frac{2}{\log\alpha} - \frac{1}{\log\alpha}(1 + \alpha^{\frac{1}{\log\alpha}(1-\varepsilon_{k,l}^{i+1}\alpha^{\mathbf{t}_{k,l}^i})})\varepsilon_{k,l}^i \alpha^{\mathbf{t}_{k,l}^i}. \tag{39}$$

Similarly, we have the following mapping from $\mathbf{t}_{k,l}^j$ to $\mathbf{t}_{k,l}^{j+1}$ in the HBDUN:

$$\mathbf{t}_{k,l}^{j+1} = \mathbf{T}_{t,k}^j \mathbf{t}_{k,l}^j - \mathbf{T}_{t,k}^j \mu_k^j \varepsilon_k^j \alpha^{\mathbf{t}_{k,l}^j} + \mathbf{q}_{t,k}^j. \tag{40}$$

We need to prove that $\mathbf{t}_{k,l}^{j+1}$ approaches $\mathbf{t}_{k,l}^{i+2}$, i.e., $\|\mathbf{t}_{k,l}^{j+1} - \mathbf{t}_{k,l}^{i+2}\|^2 < \zeta$, for any $\zeta > 0$.

For deterministic channels, we can demonstrate that there exist trainable parameters $\mathbf{T}_{t,k}^j, \mu_k^j$ and $\mathbf{q}_{t,k}^j$ that satisfy $\|\mathbf{t}_{k,l}^{j+1} - \mathbf{t}_{k,l}^{i+2}\|^2 < \zeta$. Based on (39) and (40), we obtain the following formulation:

$$\mathbf{T}_{t,k}^j = 1, \quad \forall k, \tag{41a}$$

$$\mathbf{q}_{t,k}^j = \frac{2}{\log\alpha}, \forall k, \tag{41b}$$

$$\mu_k^j = \frac{1}{\log\alpha}(1 + \alpha^{\frac{1}{\log\alpha}(1-\varepsilon_{k,l}^{i+1}\alpha^{\mathbf{t}_{k,l}^i})}), \quad \forall k, \tag{41c}$$

where $\varepsilon_{k,l}^{i+1}$ can be obtained by $\mathbf{W}_{BB}$ and $\mathbf{F}_{BB}$ in the i-th iteration according to (23). For channels that follow a certain distribution, we need to show that $\mathbb{E}_{\mathbf{H}}\{\|\mathbf{t}_{k,l}^{j+1} - \mathbf{t}_{k,l}^{i+2}\|^2\} < \delta$.

By taking $\mathbf{T}_{t,k}^j = 1$ and $\mathbf{q}_{t,k}^j = \frac{2}{\log\alpha}, \forall k$, we need to prove that

$$\mathbb{E}_{\mathbf{H}}\{\|(\mu_k^i - \frac{1}{\log\alpha}(1 + \alpha^{\frac{1}{\log\alpha}(1-\varepsilon_{k,l}^{i+1}\alpha^{\mathbf{t}_{k,l}^i})}))\varepsilon_{k,l}^i \alpha^{\mathbf{t}_{k,l}^i}\|\} < \zeta. \tag{42}$$

Based on *CauchyC-Schwarz Inequality*, we have

$$\mathbb{E}_{\mathbf{H}}\{(\mu_k^i - \frac{1}{\log\alpha}(1 + \alpha^{\frac{1}{\log\alpha}(1-\varepsilon_{k,l}^{i+1}\alpha^{\mathbf{t}_{k,l}^i})}))\varepsilon_{k,l}^i \alpha^{\mathbf{t}_{k,l}^i}\} \\ \le \mathbb{E}_{\mathbf{H}}\{\|(\mu_k^i - \frac{1}{\log\alpha}(1 + \alpha^{\frac{1}{\log\alpha}(1-\varepsilon_{k,l}^{i+1}\alpha^{\mathbf{t}_{k,l}^i})}))\|\}. \tag{43}$$

Then we set

$$\mu_k^i = \mathbb{E}_{\mathbf{H}}\{\frac{1}{\log\alpha}(1 + \alpha^{\frac{1}{\log\alpha}(1-\varepsilon_{k,l}^{i+1}\alpha^{\mathbf{t}_{k,l}^i})})\}. \tag{44}$$

According to *The Law of Large Numbers*, (43) converges to 0 with a sufficiently large number of channel samples. Thus

there exists the trainable parameter μ_k^j which satisfies (42) for any $\zeta > 0$, i.e., the performance of one layer of the proposed deep-unfolding NN can approach that of two iterations of the corresponding conventional iterative algorithm. Indeed, we can extend that the performance generated by one layer of the deep-unfolding NN with a more complex structure can approach that of multiple iterations of the conventional iterative algorithm.

VIII. 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 $N_t = 64$, $N_t^{RF} = 16$, $N_r = 32$, $N_r^{RF} = 4$, $K = 4$, and $N_s = 4$. We set the SNR as 10 dB and the number of the layers for CEDUN and HBDUN are 16 and 5, respectively. The CSI matrix is given by

$$\mathbf{H} = \sqrt{\frac{N_t N_r}{N_c N_{ray}}} \sum_{i=1}^{N_C} \sum_{l=1}^{N_{Ray}} \alpha_{il} \mathbf{a}_r(\phi_{il}^r) \mathbf{a}_t^H(\phi_{il}^t), \quad (45)$$

where N_c and N_{ray} are the number of clusters and propagation rays, respectively, $\alpha_{il} \sim \mathcal{CN}(0, \sigma_\alpha^2)$ is the complex gain, and ϕ_{il}^r and ϕ_{il}^t denote the angle of arrival (AoA) and angle of departure (AoD) for the l-th ray in the i-th cluster, respectively. The $\mathbf{a}_r(\phi_{il}^r)$ and $\mathbf{a}_t(\phi_{il}^t)$ denote the receive and transmit array response vectors, respectively, and for a uniform linear array with N antenna elements and angle ϕ, the response vector can be expressed as

$$\mathbf{a}(\phi) = \frac{1}{\sqrt{N}} \left[1, e^{-j2\pi \frac{d}{\lambda} \sin(\phi)}, \cdots, e^{-j2\pi \frac{d}{\lambda}(N-1)\sin(\phi)} \right]^T, \quad (46)$$

where d and λ denote the distances between the adjacent antennas and carrier wavelength, respectively. We set $N_c = 4, N_{ray} = 2, \sigma_\alpha^2 = 0.1, \phi_{il}^r \sim \mathcal{U}(-\pi/3, \pi/3)$ and $\phi_{il}^t \sim \mathcal{U}(-\pi/3, \pi/3)$. We use Python (version 3.6) as the programming language and use the library of Pytorch (version 1.8.1) to build the deep learning framework. Besides, the simulations are carried out on a computer with Intel i5 CPU running at 2.8GHz and with 16GB RAM.

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.
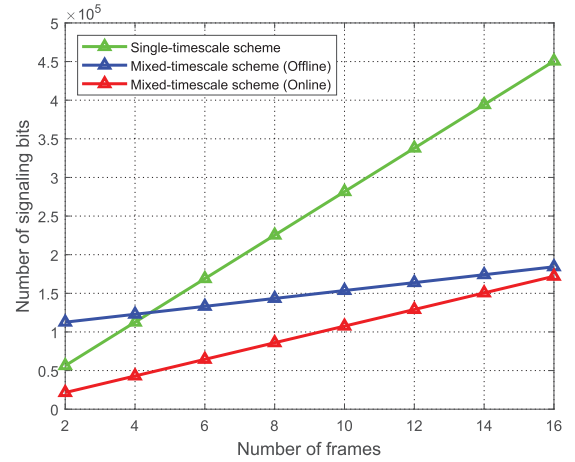

Fig. 9. Signaling bits versus the number of frames.

- RLS-ZF: The cascaded RLS algorithm for the design of channel estimation and the zero-forcing (ZF) algorithm for the hybrid beamforming design.
- Black-box NN: The proposed black-box NN for the JCEHB design.

B. Pilot Overhead

We compare the signaling bits for the feedback of pilots versus the number of frames for the single-timescale and mixed-timescale schemes in Fig. 9. For the full CSI estimation, the users need to transmit the pilots $\tilde{\mathbf{W}}_{RF,k} \in \mathbb{C}^{N_r \times N_r^{RF}}$, $\tilde{\mathbf{F}}_{RF,k} \in \mathbb{C}^{N_t \times N_t^{RF}}$, and $\tilde{\mathbf{X}}_k \in \mathbb{C}^{N_t^{RF} \times L}$. The number of feedback signaling bits for pilots is given by

$$Q_f = q(N_r N_r^{RF} + N_t N_t^{RF} + N_t^{RF} L), \quad (47)$$

where q denotes the number of bits for quantifying each element of pilot matrix. For the equivalent CSI estimation, the users only need to feed back the pilot $\tilde{\mathbf{X}}_{eq,k} \in \mathbb{C}^{N_t^{RF} \times L}$ and the number of feedback signaling bits is given by

$$Q_{eq} = q N_t^{RF} L. \quad (48)$$

For the single-timescale scheme, we need to estimate full CSI at each time slot. Thus, the number of feedback signaling bits for the single-timescale scheme over each superframe is given by

$$Q_{single} = T_f T_s Q_f, \quad (49)$$

where T_f is the number of frames in a superframe and T_s is the number of time slots in each frame.

In the offline mixed-timescale scheme, we need to estimate CSI statistics for offline training. We can estimate full CSI and employ the expectation-maximum (EM) algorithm [53] or maximum likelihood (ML) algorithm to estimate the distribution function of channel parameters, e.g., AoAs, AoDs, and complex gains. During the data transmission stage, we only need to estimate the equivalent CSI at each time slot. Thus, the pilot overhead of the offline mixed-timescale scheme over each superframe is given by

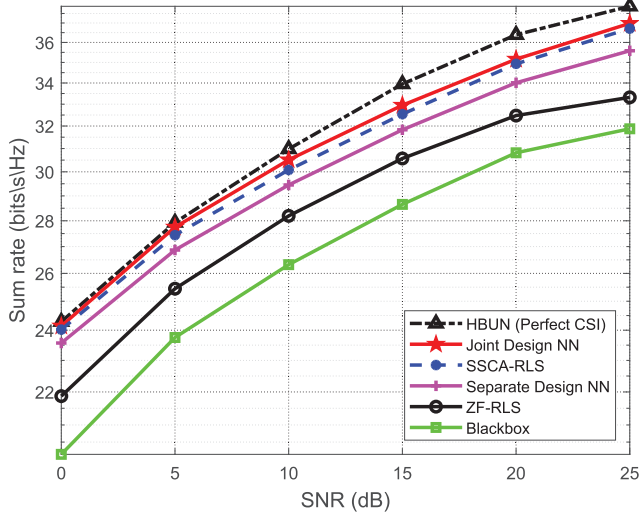$$Q_{offline} = T_f T_s Q_{eq} + N_{sample} Q_f, \quad (50)$$

Fig. 10. The sum rate versus the SNR.

Fig. 11. The sum rate versus the number of RF chains N_t^{RF}.

where N_{sample} represents the number of estimated channel samples for CSI statistics estimation.

In the online mixed-timescale scheme, we need to estimate full CSI at the end of each frame and estimate equivalent CSI at each time slot. Thus, the pilot overhead of the online mixed-timescale scheme over each superframe is given by

$$Q_{online} = T_f(T_s Q_{eq} + Q_f). \quad (51)$$

In Fig. 9, we consider a superframe where channel statistics do not change. The number of frames in a superframe depends on the coherence time of the channel. When the CSI statistics change fast, the number of frames in a superframe is small and the CSI statistics need to be estimated frequently in the offline scheme, which leads to high pilot overhead. In contrast, when the CSI statistics change slow, the number of frames is large and the pilot overhead can be reduced in the offline scheme. Here, we assume that the superframe consists of 16 frames, i.e., $T_f = 16$.

It can be observed that when the number of frames is samll, e.g., the number of frames is 2, the pilot overhead of offline mixed-timescale scheme is larger than that of the other two schemes due to the estimation for CSI statistics. As the number of frames increases, the pilot overhead of three schemes increase. We can observe that when the number of frames is large, the pilot overhead of the single-timescale scheme is much larger than that of the mixed-timescale scheme since we need to estimate the full CSI at each time slot in the single-timescale scheme. In addition, the pilot overhead of online and offline mixed-timescale is almost the same. Thus, taking both of the offline training stage and the data transmission stage into consideration, the pilot overhead of the mixed-timescale schemes is much smaller than that of the conventional single-timescale schemes.

C. System Sum Rate

Fig. 10 illustrates the sum rate of the proposed network and the benchmarks for different values of SNR. Firstly it can be seen that for all algorithms, the sum rate increases
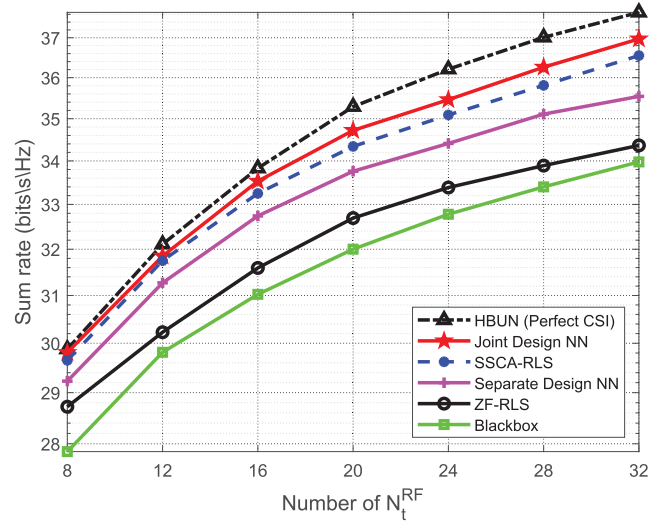
gradually with SNR. The proposed separate deep-unfolding NN can achieve comparable performance compared to the conventional SSCA and RLS algorithms, which indicates the effectiveness of our proposed deep-unfolding NN. The results also illustrate the superiority of the joint design compared to the separate NN. Besides, both of the proposed deep-unfolding NNs significantly outperform the black-box NN. It is mainly because the proposed deep-unfolding NNs are designed based on the iterative optimization algorithms.

Fig. 11 indicates the sum rate of the proposed algorithm and the benchmarks for different numbers of the RF chains N_t^{RF}. We can see that the sum rate achieved by different algorithms increases with the number of RF chains. Moreover, it is observed that the sum rate of the proposed joint design NN exceeds that of the conventional iterative algorithms. Besides, we can see that the performance gap between the joint design NN and separate design NN becomes larger as N_t^{RF} increases, which illustrates the performance gain of the joint 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 rates of deep-unfolding NNs and black-box NNs both degrade 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 presents the sum rate versus the number of training samples. We normalize the results by the sum rate of the RLS-SSCA algorithm. It is obvious that the samples required by the deep-unfolding NN for training are much fewer than that of the black-box NN. In reality, it is challenging to obtain a large quantity of training samples. Thus, the proposed deep-unfolding NN is more practical.

Table III and Table IV indicate the sum rate versus the number of layers/iterations for deep-unfolding NN and con-

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%
Black-box NN	86.95%	85.65%	85.01%	83.99%	83.42%	81.94%

TABLE II
THE SUM RATE VERSUS THE NUMBER OF TRAINING SAMPLES

Training samples	200	300	400	500	600	700	800	900	1000
Deep-unfolding NN	90.14%	94.41%	96.25%	98.09%	99.44%	100.11%	101.30%	101.31%	101.31%
Training samples	2000	3000	4000	5000	6000	7000	8000	9000	10000
Black-box NN	76.77%	77.59%	78.90%	80.52%	82.27%	83.22%	84.62%	85.44%	85.57%

TABLE III
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 IV
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%
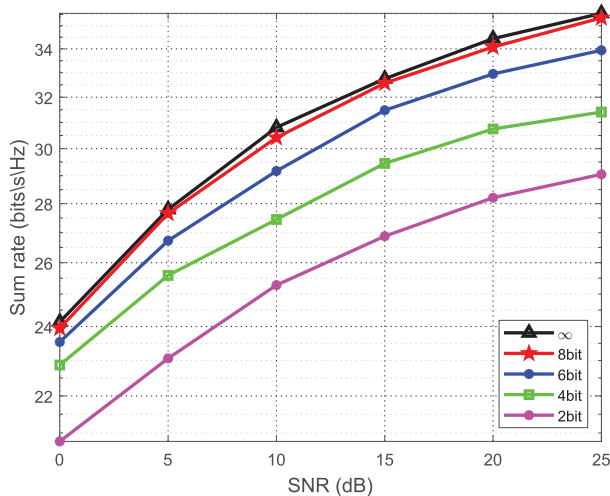
Fig. 12. The sum rate versus the SNR for different numbers of phase shifter quantization bits Q_{RF}.

Fig. 13. The sum rate versus the SNR for one/two-stage training and benchmarks.

ventional iterative algorithms. We normalize the results by the sum rate which is achieved by the conventional algorithm with 90 layers of the RLS algorithm and 70 layers of the SSCA algorithm. We can see that the deep-unfolding NNs achieve better performance than the conventional algorithms with much less number of layers. It is observed that with more layers, the sum rate achieved by the joint deep-unfolding NN improves firstly and then fluctuates. It is mainly because when there are few layers, e.g., 3 layers of the HBDUN, the degree of freedom limits its learning capability. Thus, the sum rate improves with the number of layers. When the number of layers is large, e.g., 8 layers of the HBDUN,
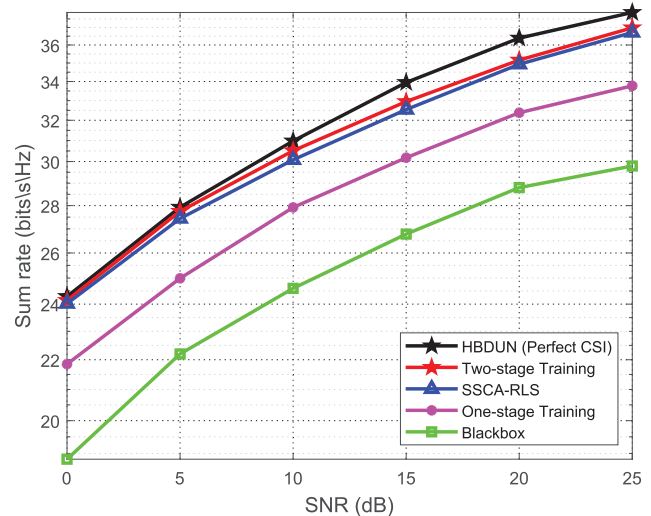
the numerical error caused by the matrix multiplication and inversion becomes large. Then the learning capability of the NN is limited and the sum rate fluctuates. Note that the layers of the CEDUN is the length of the pilots and we can observe that our proposed deep-unfolding NN can save the pilot resources compared to the RLS algorithm. Considering both the sum rate and training time, the optimal choice is 5 layers for the HBDUN and 16 layers for the CEDUN.

Table V shows the training and testing time of different algorithms with different numbers of RF chains N_t^{RF} and users K. It is observed that the training time of the deep-unfolding NN is much less than the black-box NNs,

TABLE V
THE CPU RUNNING TIME OF THE ANALYZED ALGORITHMS

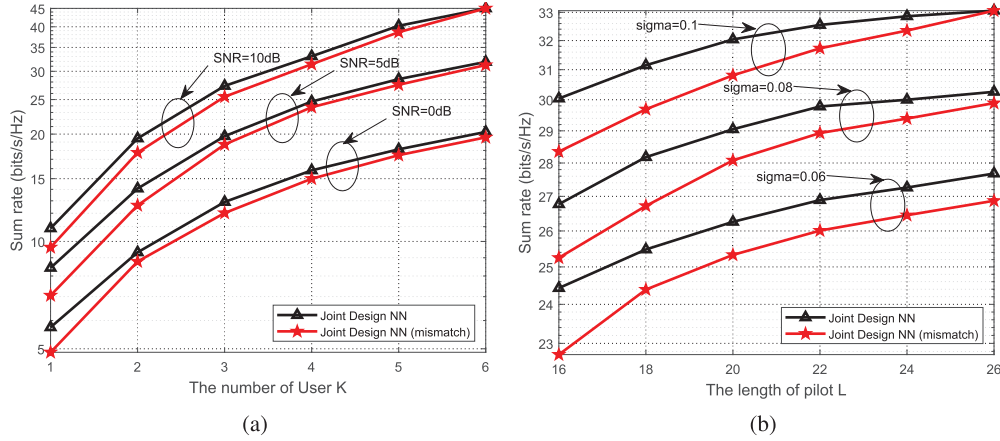| (N_t^{RF},K)	CPU training time (min)		CPU testing time (s)		
	Joint Design NN	Black-box NN	RLS-SSCA	Joint Design NN	Black-box NN
(8,2)	46.82	80.57	2.94	0.26	0.13
(16,2)	59.13	96.81	3.11	0.28	0.14
(32,2)	76.34	110.45	3.42	0.30	0.14
(16,4)	189.23	350.41	11.20	0.96	0.36
(32,4)	274.23	435.54	12.81	1.021	0.40
(32,6)	394.25	740.61	26.45	1.58	0.76

Fig. 14. The generalization ability: (a) The number of user K and SNR. (b) The length of pilot L and the complex gain of rays.

which indicates that the deep-unfolding NN converges faster. It is because that the CEDUN and HBDUN employ the structure of the RLS and SSCA algorithms, respectively. Moreover, the gap of CPU training time between the deep-unfolding NN and black-box NN increases with N_t^{RF} and K. Furthermore, the testing time of the black-box NN and the joint design NN is much less than that of the RLS-SSCA algorithm, which becomes more obvious as N_t^{RF} and K increase.

We investigate the impact of finite resolution phase shifters on the deep-unfolding NN. Specifically, we considered a uniform quantization scheme for the analog phase shifters. Fig. 12 presents the sum rate versus the SNR for different numbers of phase shifter quantization bits Q_{RF}. It can be seen that the sum rate of the proposed deep-unfolding NN improves with Q_{RF} as expected. In particular, the performance with $Q_{RF} = 8$ bits can approach the performance with infinite resolution phase shifters.

Fig. 13 indicates the sum rate versus the SNR of the one/two-stage training and benchmarks. The curve "Two-stage Training" shows the sum rate of the proposed two-stage training while the curve "One-stage Training" shows that of one-stage training where we jointly train the whole network shown in Fig.3 (b) in one stage. We see that the performance of two-stage training is better than that of one-stage training as expected, which illustrates the advantage of the proposed two-stage training method.

Fig. 14(a) illustrates the generalization ability for different numbers of K and SNR. We train the joint deep-unfolding NN in the case of $K = 6$ and SNR $= 10$ dB and test it under different values of K and SNR, such as $K = 4$ and SNR $= 5$ dB. The performance of "Joint Design NN" is obtained when the testing scenario is the same as the

training scenario while the performance of "Joint Design NN (mismatch)" is obtained when there is a mismatch between testing scenario and training scenario. It can be observed that there is only a tiny performance gap between the two curves, which demonstrates that the deep-unfolding NN has satisfactory generalization ability for different numbers of K and SNR. It is also observed that the performance gap becomes smaller when K increases.

Fig. 14(b) illustrates the generalization ability for the length of pilot L and the complex channel gain. We train the joint deep-unfolding NN in the case of $L = 26$ and $\sigma_\alpha = 0.1$ and test it under different values of L and σ_α, such as $L = 22$ and $\sigma_\alpha = 0.08$. Again, there is only a tiny performance gap between the two curves, which demonstrates that the deep-unfolding NN has satisfactory generalization ability for different values of L and σ_α. The performance gap becomes larger with decreasing L since the number of layers of the CEDUN decreases with L and the channel estimation error becomes larger.

We show the generalization ability of the deep-unfolding NN for the channel with different AoAs/AoDs and clusters/rays in Fig. 15. We train the deep-unfolding NN in the scenario where $N_c = 4, N_{ray} = 2, \phi_{il}^r \sim \mathcal{U}(-\pi/3, \pi/3)$ and $\phi_{il}^t \sim \mathcal{U}(-\pi/3, \pi/3)$. The performance of "Joint Design NN (no mismatch)" is obtained when the testing scenario is the same as the training scenario. The performance of "Joint Design NN (mismatch)" is obtained when $N_c = 3, N_{ray} = 3, \phi_{il}^r \sim \mathcal{U}(-\pi/2, \pi/2)$ and $\phi_{il}^t \sim \mathcal{U}(-\pi/2, \pi/2)$ in the testing scenario, which is different from that of training scenario. It can be seen that there is a small gap between the performance of "Joint Design NN (no mismatch)" and "Joint Design NN (mismatch)", which demonstrates that the deep-unfolding NN achieves satisfactory generalization ability for different
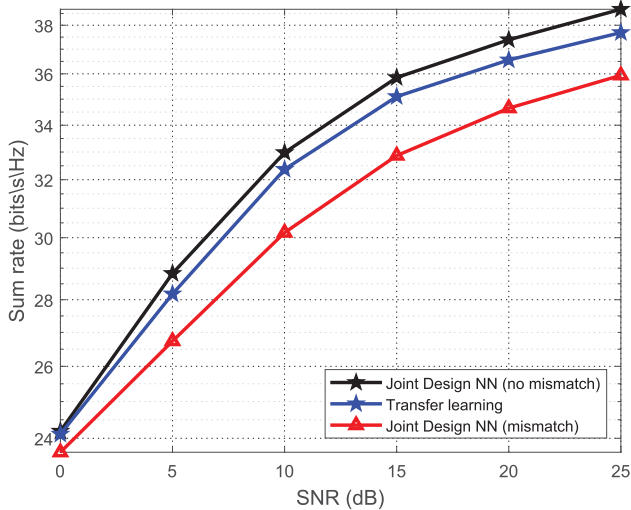
Fig. 15. The generalization ability for the channel with different AoAs/AoDs and clusters/rays.
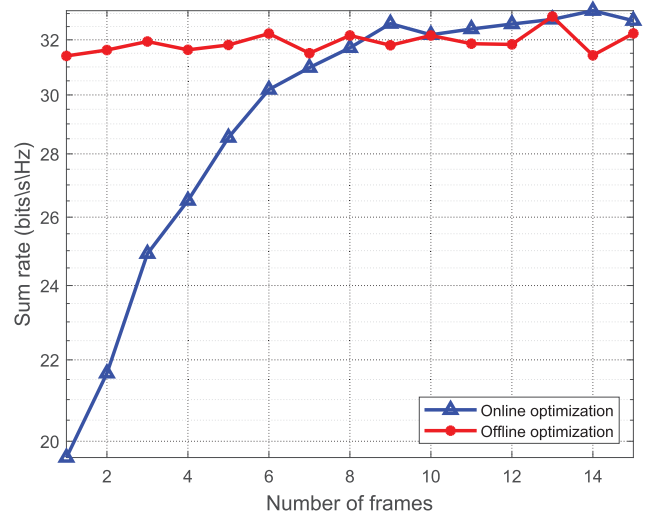

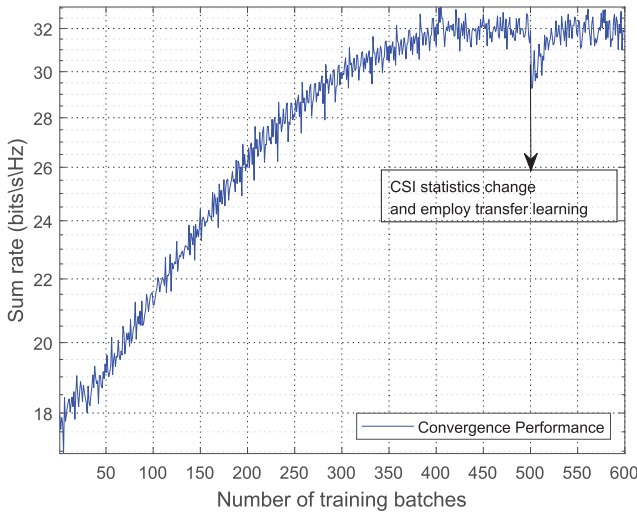Fig. 17. The sum rate of online and offline optimization versus the number of frames.

Fig. 16. The convergence performance when the CSI statistics change.
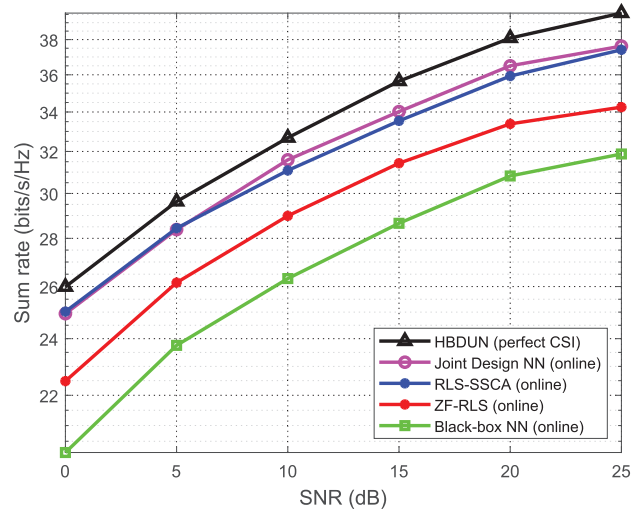

Fig. 18. The sum rate of online optimization versus the SNR.

AoAs/AoDs and numbers of clusters/rays. Furthermore, we employ transfer learning to fine-tune the deep-unfolding NN when the channel statistics change. The "Transfer learning" shows the performance of the deep-unfolding NN after transfer learning and is very close to that of "Joint Design NN (no mismatch)", which indicates the deep-unfolding NN can adapt to the change of CSI statistics after transfer learning.

Fig. 16 presents the convergence performance of the proposed deep-unfolding NN. It can be seen that when the channel statistics change, the sum rate decreases first, and then increases within several numbers of training batches, which shows that the proposed deep-unfolding NN combined with the method of transfer learning can track the channel variation rapidly. If the channel statistics change fast, we can slightly update the analog beamformers to track the variation.

Fig. 17 shows the sum rate of online and offline training versus the number of frames. We collect 50 channel samples each frame to train the deep-unfolding NN in the online training. It can be seen that the sum rate of offline training is stable as the number of frames increases because the deep-unfolding

NN is well-trained before data transmission. The sum rate of online optimization gradually improves and eventually converges as the number of frames increases because the deep-unfolding NN is optimized based on the collected channel samples. The converge performance of online training is close to that of offline training.

Fig. 18 illustrates the sum rate of the proposed deep-unfolding NN and the benchmarks for different values of SNR. The analog beamformers are optimized online in these algorithms. It is observed that the sum rate increases with SNR for all the algorithms. The performance of the deep-unfolding NN is prominently better than that of the black-box NN. Besides, the proposed joint design deep-unfolding NN achieves comparable performance of the RLS and SSCA algorithm, which indicates the effectiveness of the proposed framework of online optimization.

Fig. 19 illustrates the sum rate versus different numbers of feedback bits in the FDD system. For conventional algorithms, the optimal Lloyd-Max algorithm is employed to quantize the channel parameters $\{\phi_{il}^r, \phi_{il}^t, \Re\{\alpha_{il}\}, \Im\{\alpha_{il}\}\}$ and the phase
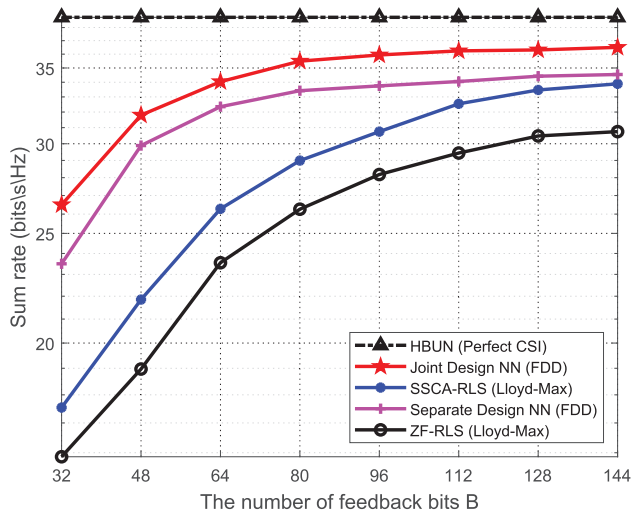
Fig. 19. The sum rate versus different numbers of feedback bits.

of analog beamformers $\{\phi_W, \phi_F\}$. The performance of "Separate Design NN (FDD)" is obtained where the deep-unfolding NNs and the feedback autoencoder are trained separately. The performance of "Joint Design NN (FDD)" is obtained where the deep-unfolding NNs and the feedback autoencoder are jointly trained according to the aforementioned training process. It can be observed that the proposed deep-learning NN outperforms the conventional algorithm with the same number of feedback bits. The joint design NN with 48 feedback bits achieves the same sum rate with the SSCA-RLS algorithm with 112 feedback bits, which verifies that the proposed deep-learning NN can significantly reduce the number of feedback bits. The joint design NN achieves better performance than the separate design NN, which indicates the effectiveness of our end-to-end joint design deep-learning framework.

IX. 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 an SSCA algorithm induced deep-unfolding NN for channel estimation and hybrid beamforming, respectively. Specifically, we introduced some trainable parameters and non-linear operations to replace the high complexity operations and increase the convergence speed. In addition, we propose a mixed-timescale deep-unfolding NN where analog beamformers are optimized online, and extend the framework to the FDD systems where channel feedback is considered. Furthermore, we analyzed the computational complexity and performance of the proposed deep-unfolding algorithm. The simulation results showed that the proposed deep-unfolding algorithm can outperform the conventional iterative algorithms. For the future study, it is worth generating our deep-unfolding framework to solve more complex wireless systems around the research hot spots, such as multi-cell MIMO, drones and intelligent reflecting surface systems.

References

[1] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 101–107, Jun. 2011.

[2] T. S. Rappaport *et al.*, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.

[3] M. R. Akdeniz *et al.*, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1164–1179, Jun. 2014.

[4] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[5] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct. 2014.

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

[7] A. F. Molisch *et al.*, "Hybrid beamforming for massive MIMO: A survey," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 134–141, Sep. 2017.

[8] O. E. Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi, and R. W. Heath, Jr., "Spatially sparse precoding in millimeter wave MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1499–1513, Jan. 2014.

[9] F. Sohrabi and W. Yu, "Hybrid digital and analog beamforming design for large-scale antenna arrays," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 501–513, Apr. 2016.

[10] 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, Jun. 2019.

[11] 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, Sep. 2017.

[12] 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.

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

[14] C. Lin, G. Y. Li, and L. Wang, "Subarray-based coordinated beamforming training for mmWave and sub-THz communications," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2115–2126, Sep. 2017.

[15] 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, Sep. 2018.

[16] Z. Gao, L. Dai, Z. Wang, and S. Chen, "Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6169–6183, Dec. 2015.

[17] J. Dai, A. Liu, and V. K. N. Lau, "FDD massive MIMO channel estimation with arbitrary 2D-array geometry," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2584–2599, May 2018.

[18] Z. Xiao, P. Xia, and X.-G. Xia, "Codebook design for millimeter-wave channel estimation with hybrid precoding structure," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 141–153, Jan. 2017.

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

[20] P. Dong, H. Zhang, G. Y. Li, I. Gaspar, and N. NaderiAlizadeh, "Deep CNN-based channel estimation for mmWave massive MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 989–1000, Sep. 2019.

[21] 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.

[22] 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.

[23] X. Chen, A. Liu, Y. Cai, V. K. N. Lau, and M. Zhao, "Randomized two-timescale hybrid precoding for downlink multicell massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 67, no. 16, pp. 4152–4167, Jul. 2019.

[24] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[25] N. Shlezinger, R. Fu, and Y. C. Eldar, "DeepSIC: Deep soft interference cancellation for multiuser MIMO detection," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1349–1362, Feb. 2021.

[26] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.

[27] N. Farsad, N. Shlezinger, A. J. Goldsmith, and Y. C. Eldar, "Data-driven symbol detection via model-based machine learning," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Jul. 2021, pp. 571–575.

[28] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748–751, Oct. 2018.

[29] 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.

[30] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, Mar. 2019.

[31] A. M. Elbir and A. K. Papazafeiropoulos, "Hybrid precoding for multiuser millimeter wave massive MIMO systems: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 552–563, Jan. 2020.

[32] Q. Hu, Y. Cai, K. Kang, G. Yu, J. Hoydis, and Y. C. Eldar, "Two-timescale end-to-end learning for channel acquisition and hybrid precoding," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 163–181, Jan. 2022.

[33] 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, Mar. 2021.

[34] H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 77–83, Oct. 2019.

[35] H. Lee, S. H. Lee, and T. Q. S. Quek, "Deep learning for distributed optimization: Applications to wireless resource management," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2251–2266, Oct. 2019.

[36] M. Eisen, C. Zhang, L. F. O. Chamon, D. D. Lee, and A. Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2775–2790, Apr. 2019.

[37] S. Khobahi, N. Shlezinger, M. Soltanalian, and Y. C. Eldar, "LoRD-Net: Unfolded deep detection network with low-resolution receivers," *IEEE Trans. Signal Process.*, vol. 69, pp. 5651–5664, 2021.

[38] J. Liao, J. Zhao, F. Gao, and G. Y. Li, "A model-driven deep learning method for massive MIMO detection," *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1724–1728, Aug. 2020.

[39] 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, May 2020.

[40] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Trans. Signal Process.*, vol. 68, pp. 1702–1715, 2020.

[41] M. Borgerding, P. Schniter, and S. Rangan, "AMP-inspired deep networks for sparse linear inverse problems," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4293–4308, Aug. 2017.

[42] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.

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

[44] 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.

[45] Q. Hu, Y. Liu, Y. Cai, G. Yu, and Z. Ding, "Joint deep reinforcement learning and unfolding: Beam selection and precoding for mmWave multiuser MIMO with lens arrays," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2289–2304, Aug. 2021.

[46] Y. Liu, Q. Hu, Y. Cai, G. Yu, and G. Y. Li, "Deep-unfolding beamforming for intelligent reflecting surface assisted full-duplex systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 4784–4800, Jul. 2022.

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

[48] H. Sun, W. Pu, M. Zhu, X. Fu, T.-H. Chang, and M. Hong, "Learning to continuously optimize wireless resource in episodically dynamic environment," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 4945–4949.

[49] J. Kaleva, A. Tölli, 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.

[50] X. Rao and V. K. N. Lau, "Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3261–3271, Jun. 2014.

[51] A. Alkhateeb, G. Leus, and R. W. Heath, Jr., "Limited feedback hybrid precoding for multi-user millimeter wave systems," *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 6481–6494, Nov. 2015.

[52] Z. Lu, J. Wang, and J. Song, "Multi-resolution CSI feedback with deep learning in massive MIMO system," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[53] L. Bai *et al.*, "Channel modeling for satellite communication channels at *Q*-band in high latitude," *IEEE Access*, vol. 7, pp. 137691–137703, 2019.

Kai Kang received the B.S. degree in information and communication engineering from Zhejiang University, Hangzhou, China, in 2021, where he is currently pursuing the M.Sc. degree with the College of Information Science and Electronic Engineering. His current research interests include hybrid precoding for MIMO wireless systems and machine learning for physical layer signal processing.

Qiyu Hu (Student Member, IEEE) received the B.S. degree in information engineering from Zhejiang University, Hangzhou, China, in 2018, where he is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering. His current research interests mainly include unmanned aerial vehicle communications, optimization theory, deep-unfolding, machine learning for physical layer communication, and semantic communication.

Yunlong Cai (Senior Member, IEEE) received the M.Sc. degree in electronic engineering from the University of Surrey, Guildford, U.K., in 2006, and the Ph.D. degree in electronic engineering from the University of York, York, U.K., in 2010. From 2010 to 2011, he was a Post-Doctoral Fellow with the Electronics and Communications Laboratory, Conservatoire National des Arts et Metiers, Paris, France. Since February 2011, he has been with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China, where he is currently a Professor. From August 2016 to January 2017, he was a Visiting Scholar with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His research interests include transceiver design for multiple-antenna systems, cooperative and relay communications, UAV communications, and machine learning for communications. He is currently an Associate Editor of IEEE SIGNAL PROCESSING LETTERS.

Guanding Yu (Senior Member, IEEE) received the B.E. and Ph.D. degrees in communication engineering from Zhejiang University, Hangzhou, China, in 2001 and 2006, respectively. He joined Zhejiang University in 2006 and is currently a Professor with the College of Information Science and Electronic Engineering. From 2013 to 2015, he was also a Visiting Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His research interests include 5G communications and networks, integrated sensing and communications, mobile edge computing/learning, and machine learning for wireless networks. He received the 2016 IEEE ComSoc Asia–Pacific Outstanding Young Researcher Award. He regularly sits on the Technical Program Committee (TPC) Boards of prominent IEEE conferences such as ICC, GLOBECOM, and VTC. He has served as the Symposium Co-Chair for IEEE Globecom 2019 and the Track Chair for IEEE VTC 2019'Fall. He has served as a Guest Editor for the Special Issue on Full-Duplex Communications of *IEEE Communications Magazine*; an Editor for IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Green Communications and Networking Series and Machine Learning in Communications and Networks Series), IEEE WIRELESS COMMUNICATIONS LETTERS, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, and IEEE ACCESS; and a Lead Guest Editor for the Special Issue on LTE in Unlicensed Spectrum of *IEEE Wireless Communications Magazine*.

Jakob Hoydis (Senior Member, IEEE) received the Diploma degree in electrical engineering from RWTH Aachen University, Germany, and the Ph.D. degree from Supélec, France. He is a Principal Research Scientist with NVIDIA working on the intersection of machine learning and wireless communications. Prior to this, he was the Head of a research department with Nokia Bell Labs, France; and the Co-Founder of the social network SPRAED. He is one of the maintainers and core developers of Sionna, a GPU-accelerated open-source link-level simulator for next-generation communication systems. He is a coauthor of the textbook *Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency* (2017). He was a recipient of the 2019 VTG IDE Johann-Philipp-Reis Prize, the 2019 IEEE SEE Glavieux Prize, the 2018 IEEE Marconi Prize Paper Award, the 2015 IEEE Leonard G. Abraham Prize, the IEEE WCNC 2014 Best Paper Award, the 2013 VDE ITG Förderpreis Award, and the 2012 Publication Prize of the Supélec Foundation. He has received the 2018 Nokia AI Innovation Award, as well as the 2018 and 2019 Nokia France Top Inventor Awards. From 2019 to 2021, he was the Chair of the IEEE COMSOC Emerging Technology Initiative on Machine Learning as well as an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. Since 2019, he has been an Area Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Machine Learning in Communications and Networks Series).

Yonina C. Eldar (Fellow, IEEE) received the B.Sc. degree in physics and the B.Sc. degree in electrical engineering from Tel Aviv University (TAU), Tel Aviv-Yafo, Israel, in 1995 and 1996, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2002. She is currently a Professor with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel. She was previously a Professor with the Department of Electrical Engineering, Technion. She is also a Visiting Professor with MIT, a Visiting Scientist with the Broad Institute, and an Adjunct Professor with Duke University; and was a Visiting Professor with Stanford University. She is an author of the book *Sampling Theory: Beyond Bandlimited Systems* and a coauthor of five other books published by Cambridge University Press. Her research interests are in the broad areas of statistical signal processing; sampling theory and compressed sensing; learning and optimization methods; and their applications to biology, medical imaging, and optics.

Dr. Eldar was a member of the Young Israel Academy of Science and Humanities and the Israel Committee for Higher Education. She is a member of the Israel Academy of Sciences and Humanities (elected 2017) and the IEEE Sensor Array and Multichannel Technical Committee and serves on several other IEEE committees. She is a fellow of EURASIP. In the past, she was a Signal Processing Society Distinguished Lecturer and a member of the IEEE Signal Processing Theory and Methods and Bio Imaging Signal Processing technical committees. She was a Horev Fellow of the Leaders in Science and Technology Program at the Technion and a fellow of Alon. She has received many awards for excellence in research and teaching, including the IEEE Signal Processing Society Technical Achievement Award in 2013, the IEEE/AESS Fred Nathanson Memorial Radar Award in 2014, and the IEEE Kiyo Tomiyasu Award in 2016. She received the Michael Bruno Memorial Award from the Rothschild Foundation, the Weizmann Prize for Exact Sciences, the Wolf Foundation Krill Prize for Excellence in Scientific Research, the Henry Taub Prize for Excellence in Research (twice), the Hershel Rich Innovation Award (three times), the Award for Women with Distinguished Contributions, the Andre and Bella Meyer Lectureship, the Career Development Chair at the Technion, the Muriel and David Jacknow Award for Excellence in Teaching, and the Technion Award for Excellence in Teaching (two times). She received several best paper awards and best demo awards together with her research students and colleagues including the SIAM Outstanding Paper Prize, the UFFC Outstanding Paper Award, the Signal Processing Society Best Paper Award, and the *IET Circuits, Devices and Systems Premium Award*. She was selected as one of the 50 most influential women in Israel and Asia and is a Highly Cited Researcher. She was the co-chair and the technical co-chair of several international conferences and workshops. She served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the *EURASIP Journal on Advances in Signal Processing*, the *SIAM Journal on Matrix Analysis and Applications*, and the *SIAM Journal on Imaging Sciences*. She is the Editor-in-Chief of *Foundations and Trends in Signal Processing*.