# Dual CNN based Channel Estimation for MIMO-OFDM Systems

Peiwen Jiang, Chao-kai Wen, *Member, IEEE,* Shi Jin, *Senior Member, IEEE,* and Geoffrey Ye Li, *Fellow, IEEE*

***Abstract*—Recently, convolutional neural network (CNN)-based channel estimation (CE) for massive multiple-input multiple-output communication systems has achieved remarkable success. However, complexity even needs to be reduced, and robustness can even be improved. Meanwhile, existing methods do not accurately explain which channel features help the denoising of CNNs. In this paper, we first compare the strengths and weaknesses of CNN-based CE in different domains. When complexity is limited, the channel sparsity in the angle-delay domain improves denoising and robustness whereas large noise power and pilot contamination are handled well in the spatial-frequency domain. Thus, we develop a novel network, called dual CNN, to exploit the advantages in the two domains. Furthermore, we introduce an extra neural network, called HyperNet, which learns to detect scenario changes from the same input as the dual CNN. HyperNet updates several parameters adaptively and combines the existing dual CNNs to improve robustness. Experimental results show improved estimation performance for the time-varying scenarios. To further exploit the correlation in the time domain, a recurrent neural network framework is developed, and training strategies are provided to ensure robustness to the changing of temporal correlation. This design improves channel estimation performance but its complexity is still low.**

***Index Terms*—Deep learning, CNN, RNN, MIMO, channel estimation, robustness.**

## I. Introduction

MASSIVE multiple-input multiple-output (MIMO) systems have been widely used for high-data-rate transmission where the base stations (BSs) equipped with multiple antennas can serve multiple users simultaneously at the same frequency bands [1]. For massive MIMO systems, channel estimation (CE) is critical to exploit the full benefit and the accuracy of CE directly affects the performance of MIMO systems. However, its accuracy is limited by pilot resources, complexity, and interference. For example, the performance of least-square (LS) estimation is poor under low signal-to-noise ratio (SNR) whereas minimum mean-squared error (MMSE) CE needs complicated large matrix operations. Some robust and low complex MMSE-CE methods [2], [3] cannot perform well for channels with long delay spread and interference.

Deep learning (DL) can improve CE performance, especially in extreme environments [4]–[11]. At the outset, end-to-end deep fully connected networks [5] outperform conventional MMSE estimation under insufficient pilots and nonlinear distortion. A deep autoencoder-based CE [8] is designed for the wireless energy transfer system due to its superiority under nonlinear and nonconvex problems. DL-based receivers can also be applied to systems with low-resolution analog-to-digital converters [9], [10], [12] and limited radio frequency chains [13]. Decision-directed channel estimation can be enhanced by DNNs [14], [15] in time-varying channels. In [16], the pilot design and CE are optimized jointly to cope with a condition in which the pilot length is less than the number of antennas of all user terminals. In addition, CE networks can be jointly designed with beamforming [17] and precoding [18]. However, an end-to-end communication system is usually challenging to train jointly because the unknown channel truncates the gradient calculation. In [19], generative adversarial networks have been used to learn the channel effects; thus, the transmitter and the receiver are connected by the generative network. In [20], the labeled direction of arrival helps the offline and online training. Some novel architectures [7], [21], [22], called model-driven methods, combine DL with expert knowledge to reduce the training data and computation resources requirements. The low-complex frameworks in [23], [24] and the meta-learning frameworks in [25], [26] significantly reduce required training resources for online adjustment.

Convolutional neural networks (CNNs) have a great potential to exploit the correlation of the adjacent elements of channels in the spatial, time, and frequency domains. This CNN-based CE [27] outperforms the traditional MMSE method. The CNN in the spatial-frequency (SF) domain (referred to as SF-CNN) [28] can obtain better estimation performance for mmWave MIMO systems but with lower complexity than the traditional MMSE method. In [29], an untrained CNN has been applied to denoise the channels with pilot contamination because CNNs easily recover a 3-D channel from a smaller input of randomly chosen input tensor by compressing the correlated channels. The sparsity of channels in a transform-domain is a crucial feature, which motivates the employment of compressive sensing approaches [30]. Recently, this feature has been exploited to CNN-based CE [31] and channel state

Manuscript received Dec. 1, 2020; revised Jan. 29 and Apr. 7, 2021; accepted May 24, 2021.The work was supported in part by the National Key Research and Development Program 2018YFA0701602, the National Natural Science Foundation of China (NSFC) for Distinguished Young Scholars with Grant 61625106, and the NSFC under Grant 61941104. The associate editor coordinating the review of this paper and approving it for publication was C.-H. Lee. (Corresponding author: Shi Jin)

P. Jiang and S. Jin are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: PeiwenJiang@seu.edu.cn; jinshi@seu.edu.cn).

C.-K. Wen is with the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan (e-mail: chaokai.wen@mail.nsysu.edu.tw).

G. Y. Li is with the Department of Electrical and Electronic Engineering, Imperial Colledge London, London, UK (e-mail: geoffrey.li@imperial.ac.uk).

information feedback [32].

In this work, we demonstrate the advantages of denoising for CNN-based CE in the SF and AD domains. However, the decreased trainable parameters and complexity inevitably limit the receptive field in the CNN [33], thus aggravating estimation performance. Therefore, we propose a novel architecture called dual CNN to exploit the channel features in both domains. Through noise analysis, we understand the mechanism of the CNNs and develop a new framework and training strategy, called hyper dual CNN, to improve robustness. Furthermore, by exploiting the time domain correlation, the proposed work is further enhanced with a recurrent neural network (RNN).

The major contributions of this paper are summarized as follows:

1) To reduce the complexity, a new architecture, called dual CNN, is proposed by connecting the CNNs in the SF and AD domains. We compare the dual CNN performance with CNN-based CE in the SF and AD domains, called SFCNN and ADCNN, respectively, and find that ADCNN outperforms SFCNN when SNR is high, whereas SFCNN is more robust to the change of SNR. The dual CNN always performs better than the CNN-based CE in any single domain when their complexities are similar.

2) To improve the robustness, we develop a novel network called HyperNet, which adaptively detects the LS estimation scenario. The novel framework called hyper dual CNN consists of several SFCNNs, an ADCNN, and a HyperNet. This framework uses HyperNet to combine the existing CNNs to cope with the time-varying environment; thus, it requires no online training and has better performance under trained scenarios than networks without the HyperNet. Meanwhile, the feasibility under untrained scenarios is also guaranteed.

3) We further enhance the dual CNN by exploiting the temporal correlation through the RNN architecture. The proposed work, called dual RNN, directly exploits the correlation between blocks without demodulating the correlative blocks together. Specifically, this network uses the trained dual CNN as initialization, and a simple CNN called TimeNet is added to deliver the information from the previous blocks, which improves estimation performance but remains low complexity.

The rest of this paper is organized as follows. Section II introduces the system model, including conventional CE algorithms and classic DL-based CE architectures. The proposed networks are presented in Section III. In Section IV, we demonstrate the superiority of the proposed networks in terms of estimation performance, robustness, and complexity. Section V concludes this paper.

## II. System Model

After introducing the multiuser MIMO-OFDM system and conventional CE methods, we present the existing AI-aided channel estimators, including DNN- and CNN-based CE in this section. Besides, we analyze the complexity of the current methods and introduce some techniques to improve robustness.

### A. Multiuser MIMO-OFDM System

We consider a BS with $M$ antennas serving $N_{\text{ue}}$ users, each with a single antenna. OFDM modulation with $K$ subcarriers is used. The length of the transmit pilot sequence is $P$. The received signal at the $k$-th subcarrier of the BS is

$$\mathbf{Y}_k = \sum_{n=1}^{N_{\text{ue}}} \sqrt{\rho_{n,k}}\mathbf{h}_{n,k} \otimes \mathbf{x}_{n,k}^* + \mathbf{Z}, \tag{1}$$

where the channel between the BS and the $n$-th user, $\mathbf{h}_{n,k} \in \mathbb{C}^{M\times 1}$, is constant over $P$ time slots by virtue of block fading, $\mathbf{x}_{n,k} \in \mathbb{C}^{P\times 1}$ is the transmit pilot, $\rho_{n,k}$ is the transmit power, $\otimes$ and $(\cdot)^*$ represent Kronecker product and Hermitian transpose and $\mathbf{Z} \in \mathbb{C}^{M\times P}$ denotes the white Gaussian noise. To estimate the channel, the pilot sequence is orthogonal among different users from the same BS, yielding

$$\mathbf{x}^*{}_{n_1,k}\mathbf{x}_{n_2,k} = \begin{cases} P, & n_1 = n_2 \\ 0, & n_1 \neq n_2 \end{cases}. \tag{2}$$

Then, LS-CE can be expressed as

$$\widehat{\mathbf{h}}_{n,k,\text{LS}} = \frac{1}{\sqrt{\rho_{n,k}}P}\mathbf{Y}_k\mathbf{x}_{n,k}, \tag{3}$$

In the subsequent discussion, we denote the true and the estimated channels of the $n$-th user at all subcarriers as $\mathbf{H}_{n,\text{LS}}$ and $\widehat{\mathbf{H}}_{n,\text{LS}} \in \mathbb{C}^{M\times K}$, respectively. However, the pilot sequences of the users from different BSs are not orthogonal, which leads to pilot contamination.

LS estimation exploits no channel statistics. It has low complexity but poor performance. MMSE-CE improves performance by using the channel correlation in time, frequency, and antennas. Here, we assume that the channel is static within an OFDM block. For convenience, the $M \times K$ matrix is converted into an $MK \times 1$ channel vector by concatenating the columns, yielding

$$\underline{\widehat{\mathbf{h}}}_{n,\text{LS}} = \text{vec}\left(\widehat{\mathbf{H}}_{n,\text{LS}}\right), \tag{4}$$

where $\underline{\widehat{\mathbf{h}}}_{n,\text{LS}} \in \mathbb{C}^{MK\times 1}$. The linear MMSE (LMMSE) estimation of the channel vector is

$$\underline{\widehat{\mathbf{h}}}_{n,\text{LMMSE}} = \mathbf{R}\left(\mathbf{R} + \sigma^2\mathbf{I}_{MK}\right)^{-1}\underline{\widehat{\mathbf{h}}}_{n,\text{LS}} = \mathbf{W}_{\text{LMMSE}}\underline{\widehat{\mathbf{h}}}_{n,\text{LS}}, \tag{5}$$

where $\sigma^2$ denotes noise power and $\mathbf{R} \in \mathbb{C}^{MK\times MK}$ is the autocorrelation matrix of subcarriers and BS antennas, which is expressed as the expectation of the true channel vector $\mathbf{R} = E(\underline{\mathbf{h}}_n\underline{\mathbf{h}}_n^*)$ but usually obtained by time-averaging or according to channel models [34] in practice. Since the matrix multiplication requires $O\left((MK)^2\right)$ scalar operations, the MMSE estimator is much more complicated than the LS estimator. Moreover, $\mathbf{W}_{\text{LMMSE}}$ may need to be updated along with the change of scenarios where matrix inversion requires $O\left((MK)^3\right)$ multiplications. Here, the spatial correlation of the users is ignored. Otherwise, the matrix multiplication requires $O\left((MK)^2N\right)$ scalar operations for each user, where $N$ is the number of correlative user antennas.

To simplify the LMMSE estimator, the max delay of the channels, $l_{\max}$, is assumed offline and the correlation in an-

tennas is ignored. The robust LMMSE estimation at the $n$-th user and the $m$-th antenna ($\hat{\mathbf{h}}_{n,m,\mathrm{RLMMSE}} \in \mathbb{C}^{K\times 1}$) can easily be calculated by the fast Fourier transform (FFT) because the correlation in the frequency domain, $\mathbf{R}_f = E(\mathbf{h}_{n,m}\mathbf{h}_{n,m}^*)$, can be eigendecomposed [3] into

$$\mathbf{R}_f = \mathbf{D}E(\tilde{\mathbf{h}}_{n,m}\tilde{\mathbf{h}}_{n,m}^*)\mathbf{D}^* = \mathbf{D}\begin{pmatrix} \frac{1}{l_{\max}}\mathbf{I}_{l_{\max}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}\mathbf{D}^*, \quad (6)$$

where $\mathbf{D}$ can be approximated by the $K \times K$ discrete Fourier transform (DFT) matrix [2] and $\tilde{\mathbf{h}}_{n,m}$ is the true channel in the delay domain at the $n$-th user and the $m$-th antenna. The robust LMMSE estimation is expressed as

$$\begin{aligned} \underline{\widehat{\mathbf{h}}}_{n,m,\mathrm{RLMMSE}} &= \mathbf{R}_f\left(\mathbf{R}_f + \sigma^2\mathbf{I}_K\right)^{-1}\underline{\widehat{\mathbf{h}}}_{n,m,\mathrm{LS}} \\ &= \mathbf{W}_{\mathrm{RLMMSE}}\underline{\widehat{\mathbf{h}}}_{n,m,\mathrm{LS}}. \end{aligned} \quad (7)$$

As a result, the complexity of the robust LMMSE estimation for each user is reduced to $O(MK\log K)$. In the following, it is denoted as RLMMSE.

Compared with LMMSE, RLMMSE is less complicated but performs worse because RLMMSE does not exploit the spatial correlation and assumes that the power in the delay domain distributes uniformly.

### B. DL-based CE

In Fig. 1(a), the estimated channel using the classic fully connected DNN can be written as

$$\underline{\widehat{\mathbf{h}}}_{n,\mathrm{DNN}} = \mathbf{W}_L \cdots \beta\left(\mathbf{W}_2\beta\left(\mathbf{W}_1\underline{\widehat{\mathbf{h}}}_{n,\mathrm{LS}} + \mathbf{b}_1\right) + \mathbf{b}_2\right)\cdots + \mathbf{b}_L, \quad (8)$$

where $\mathbf{W}_i$ and $\mathbf{b}_i$ denote the real multiplicative parameter matrix and the additive parameter vector for the $i$-th hidden layer, and $\beta(\cdot)$ is a nonlinear activation function. For fully-connected DNN-based CE, the sizes of $\mathbf{W}_i$ and $\mathbf{b}_i$ increase with the numbers of antennas and subcarriers. The complexity of this architecture is larger than $O((MK)^2)$. The DL-based receiver reveals its superiority for extreme scenarios, such as insufficient pilots and nonlinear interference. However, complexity is the key restriction to many applications of DL in wireless communications. Moreover, too many trainable parameters are difficult to train and update with the change of scenarios. Thus, CNN-based receivers are used to simplify the architecture.

In Fig. 1(b), the CE module is usually designed as a CNN-based denoiser, where the channels are regarded as two-dimensional pictures with frequency and antennas as height and width, respectively. Therefore, the complexity is $O(\sum_{i=1}^{L}(cMKN_{i-1}N_i))$, where $N_i$ denotes the number of filters in the $i$-th layer, the filter size is $c$. The input of the $i$-th layer is $(M, K, N_{i-1})$, which means this input matrix has three dimensions with the sizes $M$, $K$, $N_{i-1}$, respectively.

Transfer learning is a common method for adapting the trained network to a new environment. According to [23]–[26], we can either reduce trainable parameters or exploit novel training strategies to save pilot resources online. Some architectures [35], [36] can adjust themselves without online transfer learning. The SNR feedback is utilized in [35] while an extra DNN, called hyper-net, to adjust all the trainable

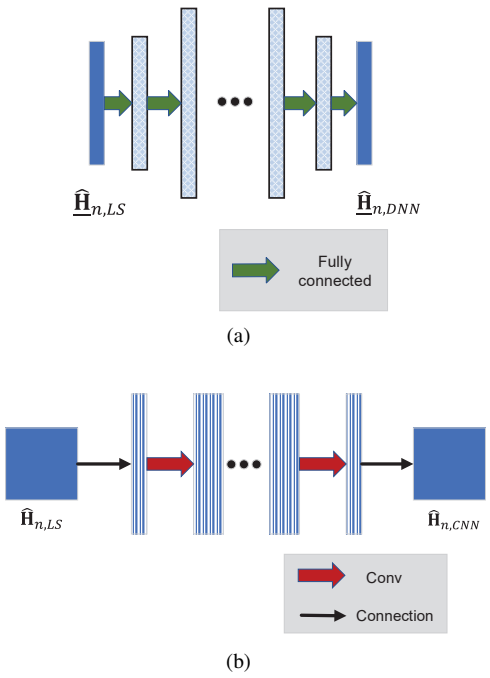



Fig. 1. (a) DNN-based CE. The channels are converted to a vector, and the correlation is fully utilized. (b) CNN-based CE. The channels are considered images, and the correlation of adjacent elements is more important.

weights in [36]. We take the DNN-based CE as an example to describe the architecture of hyper-net. For convenience, the process of a neural network is denoted as a function $f(\mathbf{a};\mathbf{b})$ in the following, where $\mathbf{a}$ is the input of the network and $\mathbf{b}$ contains all the trainable parameters of the network. Thus, Eq. (8) is rewritten as

$$\underline{\widehat{\mathbf{h}}}_{n,\mathrm{DNN}} = f_{\mathrm{DNN}}(\underline{\widehat{\mathbf{h}}}_{n,\mathrm{LS}};\mathbf{W}), \quad (9)$$

where $\mathbf{W}$ denotes $[\mathbf{W}_1,\cdots,\mathbf{W}_L;\mathbf{b}_1,\cdots,\mathbf{b}_L]$, $f_{\mathrm{DNN}}(\cdot;\cdot)$ is the process of the DNN-based CE. Then, a hyper-net is used to generate $\mathbf{W}$ with some key parameters as an input. The process is expressed as

$$\mathbf{W} = f_{\mathrm{hyper-net}}(l_{max},\sigma^2,\cdots;\mathbf{W}'), \quad (10)$$

where $\mathbf{W}'$ denotes the trainable parameters in hyper-net. Thus, the entire process is

$$\underline{\widehat{\mathbf{h}}}_{n,\mathrm{DNN}} = f_{\mathrm{DNN}}(\underline{\widehat{\mathbf{h}}}_{n,\mathrm{LS}}; f_{\mathrm{hyper-net}}(l_{max},\sigma^2,\cdots;\mathbf{W}')). \quad (11)$$

After $\mathbf{W}'$ is trained, the original trainable parameters, $\mathbf{W}$, are controlled by the key parameters, such as $l_{max}$ and $\sigma^2$. These key parameters are provided by the user, which is more convenient compared with retraining $\mathbf{W}$ online.

### C. Existing Challenges

*1) High complexity:* For the two classic DL-based CE methods mentioned above, the CNN has fewer trainable parameters, but their complexity remains large unless the filter size $c$ and the hidden layer size $N_i$ are reduced. For example, in [27], the CNN with several hidden layers of 64 $3\times 3$ filters

is applied to interpolation and denoising after LS-CE, where the input of the CNN is with 72 subcarriers and 14 time slots. This architecture exploits the correlation in the time and frequency domains; thus, the complexity of one hidden layer is $O(9\times14\times72\times64\times64)$. This CNN is much more complicated than the LMMSE without online updating ($O((72\times14)^2)$) and is larger than a couple of fully-connected layers.

*2) Robustness without online training:* Online training for DL-based CE and online updating for LMMSE are difficult due to substantial computational complexity and data requirements. CE networks are difficult to be enhanced by the hyper-net in Eq. (11) because its improvement is unstable if the key parameters cannot be obtained accurately and $\mathbf{W}$ is large. Self-attention architecture [37] has great potential because it learns to concentrate on the essential parts automatically. Although self-attention architecture is too complicated and cannot be chosen to expand the dual CNN, it still inspires us to combine several parallel networks trained under different scenarios with a weight vector; thus, the proposed network can pay attention to scenarios changing.

In this paper, we aim to simplify the CNN-based CE and develop a robust architecture without any online training.

## III. CE Based on Dual CNN Architecture

In this section, the basic network is expanded, and a novel framework called HyperNet is introduced for online adaption. The simple RNN architecture is added to exploit temporal correlation further.

### A. Basic Dual CNNs

To reduce computational complexity, the number of filters in the hidden layers should decrease; as a result, the network can only learn limited features. The small filter size and few hidden layers restrict receptive fields; thus, some global features of the channels are ignored.

Since channel matrix $\mathbf{H}_{\rm SF} \in \mathbb{C}^{M\times K}$ in the SF domain displays its correlation at adjacent subcarriers and antennas, the CNN denoiser in the image process is usually effective. Meanwhile, the channel can be easily converted to the AD domain by DFT. The frequency domain can be converted to the delay domain through inverse DFT (IDFT) and the spatial domain can be converted to angle domain through DFT [38], yielding

$$\mathbf{H}_{\rm AD} = D(\mathbf{H}_{\rm SF}) = \mathbf{D}_M \mathbf{H}_{\rm SF} \mathbf{D}_K^*, \tag{12}$$

where $\mathbf{D}_K$ is a $K\times K$ DFT matrix, $\mathbf{D}_M$ is an $M\times M$ DFT matrix. This domain transform process and its inverse process are denoted as $D(\cdot)$ and $D^{-1}(\cdot)$, respectively. The channel matrix in the AD domain, $\mathbf{H}_{\rm AD} \in \mathbb{C}^{M\times K}$, is sparse because the channel paths only spread in a limited area. The two domains have different features, which have been exploited to design a conventional channel estimator and a DNN-based estimator. By contrast, CNN focuses on the local features and cannot learn the two different features adequately from only one domain.

To solve the above issues, a dual CNN architecture, as shown in Fig. 2, is proposed. The domain transform processes, $D(\cdot)$ and $D^{-1}(\cdot)$, are introduced to help the network learn from different domains. This design combines the CNN and expert knowledge in wireless communications. Dual CNN remains low complexity by using a few $3\times3$ filters and only two hidden layers. To compensate for the performance loss, the CNN denoisers in the two domains, denoted as SFCNN and ADCNN, are connected. The input size is $(M, K, 2N)$, where the complex value is converted to real in the third dimension and $N$ is the number of the transmission antennas of the users. For example, if $N$ user antennas are correlative, they should be inputted together to exploit the spatial correlation. If $N$=1, the real part and the imaginary part of $\widehat{\mathbf{H}}_{n,\rm LS}$ in Section II are concatenated in the third dimension to form the real matrix $\widehat{\mathbf{H}}_{\rm LS}$. In the figure, the first two convolutions in the SF domain use $8N$ filters and the leaky ReLU activation function. The last convolution has $2N$ filters without an activation function. The output size of the SFCNN is $(M, K, 2N)$, and a skip connection adds the input to the output [39], which can be formulated as

$$\widehat{\mathbf{H}}_{\rm SF} = f_{\rm CNN}(\widehat{\mathbf{H}}_{\rm LS}; \boldsymbol{\theta}_{\rm SF}) + \widehat{\mathbf{H}}_{\rm LS}, \tag{13}$$

where $\boldsymbol{\theta}_{\rm SF}$ denotes the trainable parameters in the SFCNN and $f_{\rm CNN}(\cdot)$ represents the CNN processes. The ADCNNs have the same architecture, and its input is the transformed output of the SFCNN. The ADCNN in Fig. 2 further exploits channel features in the AD domain to improve estimation performance.

The output of SFCNN, $\widehat{\mathbf{H}}_{\rm SF}$, is first converted into the AD domain using $D(\cdot)$. Next, a CNN denoted as $f_{\rm CNN}(\cdot;\boldsymbol{\theta}_{\rm AD})$ is used to exploit channel features in the AD domain. Thus, the estimated channel, $\widehat{\mathbf{H}}$, in Fig. 2 can be expressed as

$$\widehat{\mathbf{H}} = D^{-1}(f_{\rm CNN}(D(\widehat{\mathbf{H}}_{\rm SF}); \boldsymbol{\theta}_{\rm AD}) + D(\widehat{\mathbf{H}}_{\rm SF})), \tag{14}$$

where $\boldsymbol{\theta}_{\rm AD}$ denotes the trainable parameters in the ADCNN.

The loss function is the mean-squared error, that is,

$$(\hat{\boldsymbol{\theta}}_{\rm SF}, \hat{\boldsymbol{\theta}}_{\rm AD}) = \underset{\boldsymbol{\theta}_{\rm SF}, \boldsymbol{\theta}_{\rm AD}}{\arg\min} ||\mathbf{H} - \widehat{\mathbf{H}}||_2^2. \tag{15}$$

When $N$=1, the complexity of the hidden layer in the SFCNN is $O(3^2MK(2\times8+8\times8+8\times2))$. The two FFTs in the network cost $O(2\times(MKlogK + KMlogM))$. Thus, the total complex multiplicative operations for each user is $O(8\times(108\times MK + MKlogK + KMlogM))$. The dual CNN estimator in Fig. 2 has lower complexity than LMMSE. When $K$ is large, the dual CNN in Fig. 2 is even simpler than RLMMSE.

Dual CNN adopts two CNN denoisers in the two domains to learn additional channel features with low complexity. However, robustness is still a problem because the architecture cannot update itself if the scenario changes.

### B. Hyper Dual CNN

In contrast to the deep unfolding network with a hyper-net in [40], dual CNN has at least thousands of trainable parameters to adapt. Meanwhile, some environment information, such as exact channel statistical state and noise power, which requires extra computation, will not be considered. Thus, we use an extra network to combine the parallel networks trained under different scenarios with the input of LS-CE. In the following,

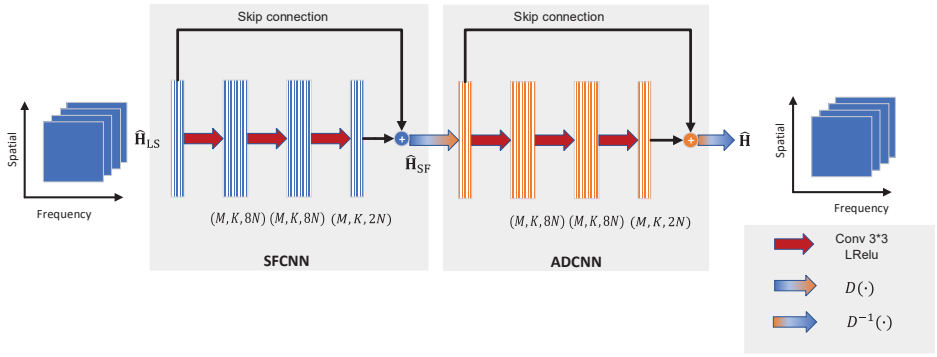



Fig. 2. Structure of a dual CNN, which contains an ADCNN and a SFCNN. The two CNNs are connected by DFT process.

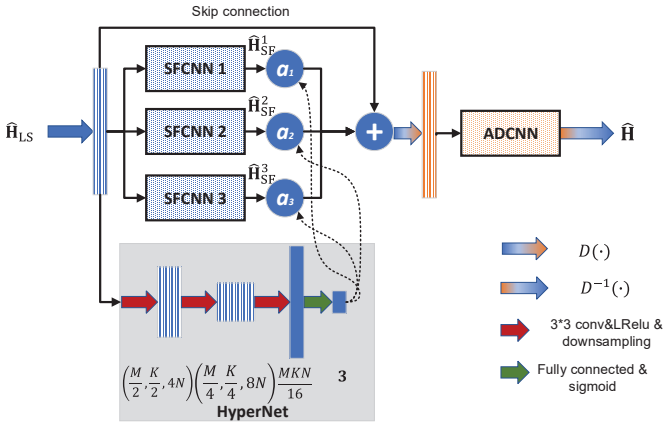



Fig. 3. Structure of a hyper dual CNN, which contains a HyperNet, several SFCNNs, and one ADCNN.

the three different scenarios in a spatial channel model (SCM) are considered an example.

Fig. 3 shows the structure of the hyper dual CNN. In the figure, additional SFCNNs are used for adaptation and HyperNet is designed as a classifier, which is stable under the noisy input because it only has few output parameters. SFCNN is affected more than ADCNN when the scenario changes due to sparsity in the AD domain. Therefore, most areas in the AD domain are only with noise power, and thus the ADCNN can remove a large part of the noise without the effect of the varying channels. On the contrary, noise and channel are added in the SF domain and the channel correlation is vitally important for the SFCNN denoiser.

The input of HyperNet is LS-CE, which is $(M, K, 2N)$ in real form. Then, the input is convoluted by $3 \times 3$ filters and doubled in the third dimension, and the activation function is ReLU. The downsampling process reduces the first two dimensions to half. The convolution and downsampling are repeated twice. In the end, a fully connected layer is used to output three parameters, and the sigmoid function limits the output from 0 to 1. The three output parameters are multiplied to the output of SFCNNs and control the contributions of the different SFCNNs. The process of HyperNet and trainable parameters are denoted as $f_{\mathrm{hyper}}(\cdot)$ and $\boldsymbol{\theta}_{\mathrm{hyper}}$, respectively. The output consists of three parameters $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3]$, each representing a scenario, and can be expressed as

$$\boldsymbol{\alpha} = f_{\mathrm{hyper}}(\widehat{\mathbf{H}}_{\mathrm{LS}}; \boldsymbol{\theta}_{\mathrm{hyper}}). \tag{16}$$

Then, based on Eq. (14), the output of the hyper dual CNN can be written as

$$\widehat{\mathbf{H}} = D^{-1}\left(f_{\mathrm{CNN}}\left(D\left(\sum_{i=1}^{3}\alpha_i\widehat{\mathbf{H}}^i_{\mathrm{SF}}\right);\boldsymbol{\theta}_{\mathrm{AD}}\right) + D\left(\sum_{i=1}^{3}\alpha_i\widehat{\mathbf{H}}^i_{\mathrm{SF}}\right)\right), \tag{17}$$

where $\widehat{\mathbf{H}}^i_{\mathrm{SF}}$ denotes the output of the $i$-th SFCNN.

In order to make the hyper dual CNN robust, three training steps are performed.

1) The ADCNN is trained first without the SFCNNs under an entire training set, where the channels from three main scenarios are mixed. The training process is formulated as

$$\hat{\boldsymbol{\theta}}_{\mathrm{AD}} = \arg\min_{\boldsymbol{\theta}_{\mathrm{AD}}} ||D^{-1}(f_{\mathrm{CNN}}(D(\widehat{\mathbf{H}}_{\mathrm{LS}});\boldsymbol{\theta}_{\mathrm{AD}}) + D(\widehat{\mathbf{H}}_{\mathrm{LS}})) - \mathbf{H}||_2^2. \tag{18}$$

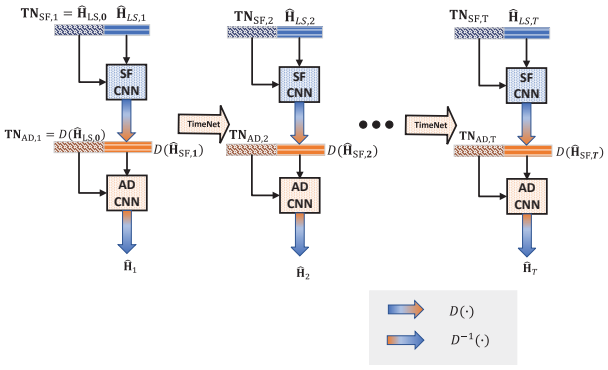
The end-to-end training is beneficial to improve performance when the training and the test scenarios are the same. In the same scenario, the ADCNN can better detect the channel power after the denoising of the SFCNN. However, training the CNNs together means the ADCNN relies on the pre-denoising in the SF domain. So, we train the ADCNN independently with the LS-CE input to improve the performance under untrained scenarios where SFCNNs may not work.

2) The SFCNNs are trained for three main scenarios successively. In this way, we obtain three SFCNNs, which can improve the performance if the test and the training scenarios are matched. However, the SFCNNs have poor robustness. The entire training process is expressed as

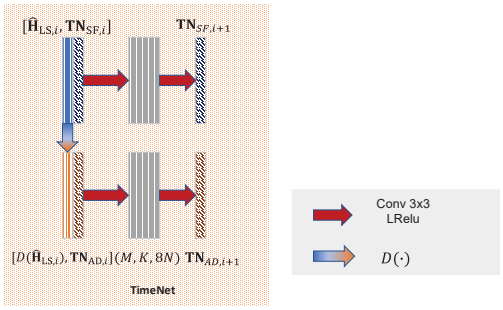
$$\hat{\boldsymbol{\theta}}^i_{\mathrm{SF}} = \arg\min_{\boldsymbol{\theta}^i_{\mathrm{SF}}} ||D^{-1}(f_{\mathrm{CNN}}(D(\widehat{\mathbf{H}}^i_{\mathrm{SF}});\hat{\boldsymbol{\theta}}_{\mathrm{AD}}) + D(\widehat{\mathbf{H}}^i_{\mathrm{SF}})) - \mathbf{H}||_2^2, \tag{19}$$

where $\boldsymbol{\theta}^i_{\mathrm{SF}}$ is the trainable parameters of the $i$-th SFCNN.

3) HyperNet with LS estimation as input is trained to output $\boldsymbol{\alpha}$, and other parameters are fixed. After training, HyperNet can be considered as a recognizer because it outputs different $\boldsymbol{\alpha}$ under different scenarios, which combines the SFCNNs




(a)




(b)

Fig. 4. Architecture of the dual RNN. (a) Overview of the RNN. (b) Details of TimeNet.

to adapt to scenario changes. The training process can be expressed as

$$\hat{\boldsymbol{\theta}}_{\text{hyper}} = \arg\min_{\boldsymbol{\theta}_{\text{hyper}}} ||\mathbf{H} - \widehat{\mathbf{H}}||_2^2. \tag{20}$$

After completing the training process offline, the hyper dual CNN is ready to work without any online training. This architecture can perform better than the dual CNN under the three scenarios. More importantly, it can still work under untrained scenarios due to the combination of the SFCNNs and the robust design of the ADCNN.

### C. Dual RNN

In addition to antennas and frequency, temporal correlation of channel state information among adjacent OFDM blocks can be exploited to improve CE further. Here, the channels in an OFDM block are assumed to be static and $T$ contiguous OFDM blocks are correlated. MMSE estimation, which maximizes spatial, frequency, and temporal correlations, should collect the pilots in the $T$ blocks and calculate them together.

A simple CNN called TimeNet is added to extract the correlation feature among OFDM blocks. The proposed architecture, called dual RNN[1], consists of dual CNNs and TimeNets at different blocks. These networks are connected as shown in Fig. 4(a). Here, two matrices at the $i$-th block, $\mathbf{TN}_{\text{SF},i}$ and $\mathbf{TN}_{\text{AD},i}$, are introduced to store temporal information in the SF and the AD domains from the previous OFDM blocks. The two matrices are delivered by the TimeNet. The temporal information at the first dual CNN directly utilizes the LS-CE in the previous time slot in the SF and AD domains as initialization, i. e., $\mathbf{TN}_{\text{SF},1} = \hat{\mathbf{H}}_{\text{LS},0}$ and $\mathbf{TN}_{\text{AD},1} = D(\hat{\mathbf{H}}_{\text{LS},0})$.

The dual CNN at the $i$-th block is changed to exploit $\mathbf{TN}_{\text{SF},i}$ and $\mathbf{TN}_{\text{AD},i}$. They are concatenated with the input of SFCNN and ADCNN. Thus, the output of SFCNN of the dual CNN at the $i$-th block is

$$\widehat{\mathbf{H}}_{\text{SF},i} = f_{\text{CNN}'}([\widehat{\mathbf{H}}_{\text{LS},i}, \mathbf{TN}_{\text{SF},i}]; \boldsymbol{\theta}_{\text{SF}}, \boldsymbol{\theta}'_{\text{SF}}) + \widehat{\mathbf{H}}_{\text{LS},i}, \tag{21}$$

where $f_{\text{CNN}'}(\cdot)$ represents the CNN process in the changed dual CNN and $\boldsymbol{\theta}'_{\text{SF}}$ denotes the extra trainable parameters because the size of the input layer in the third dimension is doubled and the trainable parameters in the filters of the first convolution is also doubled. For example, if the size of the input $\widehat{\mathbf{H}}_{\text{LS},i}$ is $(M, K, 2)$ and thus that of $[\widehat{\mathbf{H}}_{\text{LS},i}, \mathbf{TN}_{\text{SF},i}]$ is $(M, K, 4)$, the size of the eight $3\times3$ filters in the first layer of CNN process $f_{\text{CNN}}(\cdot)$ is $(3, 3, 2)$ and that of $f_{\text{CNN}'}(\cdot)$ is $(3, 3, 4)$. ADCNN is revised similarly, yielding

$$\widehat{\mathbf{H}}_i = D^{-1}(f_{\text{CNN}'}([D(\widehat{\mathbf{H}}_{\text{SF},i}), \mathbf{TN}_{\text{AD},i}]; \boldsymbol{\theta}_{\text{AD}}, \boldsymbol{\theta}'_{\text{AD}}) + D(\widehat{\mathbf{H}}_{\text{SF},i})). \tag{22}$$

The trained parameters in the dual CNN in Section III. A can be used to initiate the dual CNNs in the dual RNN, i. e., $\boldsymbol{\theta}_{\text{SF}} = \hat{\boldsymbol{\theta}}_{\text{SF}}$ and $\boldsymbol{\theta}_{\text{AD}} = \hat{\boldsymbol{\theta}}_{\text{AD}}$. Meanwhile, the extra parameters, $\boldsymbol{\theta}'_{\text{SF}}$ and $\boldsymbol{\theta}'_{\text{AD}}$, are set as 0, where the effect of the extra input, $\mathbf{TN}_i$, is eliminated. Thus, the performance of the dual RNN is equal to that of the dual CNN after initiation.

The details of TimeNet are shown in Fig. 4(b). TimeNet has one hidden layer in each domain and the first convolution uses $8N$ $3\times3$ filters and leaky ReLU activation function. The last convolution has $2N$ $3\times3$ filters with activation function. The output of TimeNet in the SF domain at the $i$-th block is $\mathbf{TN}_{\text{SF},i+1}$ while the input of this TimeNet is $\mathbf{TN}_{\text{SF},i}$ and the current LS-CE $\widehat{\mathbf{H}}_{\text{LS,SF},i}$. The output of TimeNet in the AD domain at the $i$-th block, $\mathbf{TN}_{\text{AD},i+1}$, is obtained by the same process with the input of $\widehat{\mathbf{H}}_{\text{LS},i}$ and $\mathbf{TN}_{\text{AD},i}$. The CNN processes of TimeNet in two domains are both denoted as $g_{\text{CNN}}(\cdot)$. The process of TimeNet in the SF domain can be expressed as

$$\mathbf{TN}_{\text{SF},i+1} = g_{\text{CNN}}([\widehat{\mathbf{H}}_{\text{LS},i}, \mathbf{TN}_{\text{SF},i}]; \boldsymbol{\omega}_{\text{SF}}) \tag{23}$$

and that in the AD domain

$$\mathbf{TN}_{\text{AD},i+1} = g_{\text{CNN}}([D(\widehat{\mathbf{H}}_{\text{LS},i}), \mathbf{TN}_{\text{AD},i}]; \boldsymbol{\omega}_{\text{AD}}), \tag{24}$$

where $[\widehat{\mathbf{H}}_{\text{LS},i}, \mathbf{TN}_{\text{SF},i}]$ means that $\widehat{\mathbf{H}}_{\text{LS},i}$ and $\mathbf{T}_{\text{SF},i}$ are connected in the third dimension with the trainable parameters, $\boldsymbol{\omega}_{\text{SF}}$ and $\boldsymbol{\omega}_{\text{AD}}$.

Then, the architecture is improved by training with the following loss function

$$Loss = \frac{1}{T}\sum_{i=1}^{T} ||\mathbf{H}_i - \widehat{\mathbf{H}}_i||_2^2. \tag{25}$$

[1]We choose the classic RNN rather than LSTM or GRU because this study concentrates on low complexity and the proposed dual RNN is a combination of the CNNs. Meanwhile, the temporal correlation coefficient of the channels usually decreases with time, where long-term memory may not be effective.

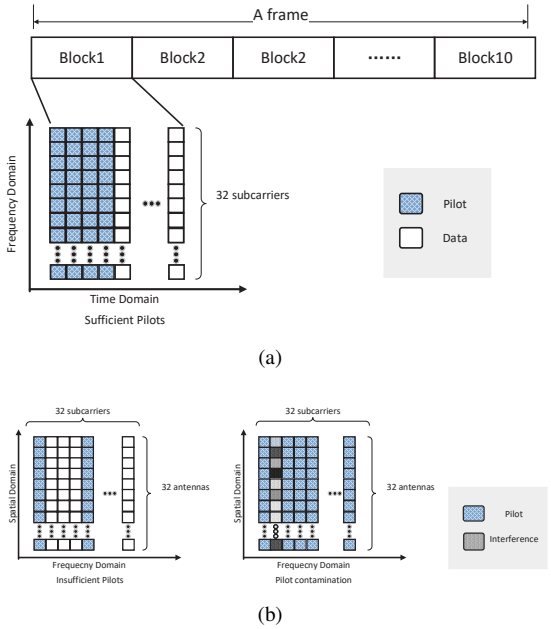



Fig. 5. (a) Frame structure transmitted by a single-antenna user. The pilot length $P$=4 is sufficient for four users. (b) Insufficient pilots: $P$=1 and each user occupies 1/4 subcarriers. Pilot contamination: $P$=4 and several subcarriers face interference.

The dual RNN can still demodulate block by block, and each block is only added with an extra CNN, called TimeNet. TimeNet only has one hidden layer in two domains, and thus it is with low complexity. This architecture is also easily applied to the hyper dual CNN because this design only introduces an extra part for time, and trained parameters from its original CE network are used for initialization.

## IV. Numerical Results

In this section, we demonstrate the numerical results under different scenarios and discuss the pros and cons of the CNN-based receivers through noise analysis. We also compare the complexity of the proposed networks and the competing ones.

### A. Configuration

The SCM channel model [41], [42] is used to generate channel realizations in three classic scenarios: urban micro, urban macro, and suburban macro. The max delay spread is six, and each path has 20 subpaths in default. The frame structure of the OFDM system is shown in Fig. 5(a). The BS has 32 antennas and serves four single-antenna users. The pilots are orthogonal for all users served by the same BS. Thus, pilot length $P$ is no less than $N_{\text{ue}}$ if the pilot is sufficient to occupy all 32 subcarriers.

Although each UE only transmits eight pilots, LS-CE can still be obtained through interpolation due to the frequency domain's correlation. Thus, the pilot length is limited to one, which is called insufficient pilot condition. The pilot sequences for the users served by the same BS are random in the frequency domain and orthogonal in the time domain. However, pilots from the users corresponding to different BSs are not necessarily orthogonal, which is called pilot contamination in massive MIMO literature. Fig. 5(b) shows insufficient pilots and pilot contamination in the SF domain. The insufficient pilot can be addressed by direct interpolation, whereas the injured resource elements need to be identified to address pilot contamination. In the following, 5% of the elements in the SF domain is injured, and the SIR is 5 dB.

We simulate 100,000 channel realizations under each scenario for training. The 10 channels in a frame are correlated, and the dual RNN has 10 dual CNNs connected by TimeNet. The training SNR is 10 dB. The optimizer for all networks is Adam [43] and the initial learning rate is 0.001. We utilize normalized mean-squared error (NMSE) to measure the CE performance, yielding

$$\text{NMSE} = E\left(\frac{||\mathbf{H} - \widehat{\mathbf{H}}||_2^2}{||\mathbf{H}||_2^2}\right), \tag{26}$$

where $\mathbf{H}$ and $\widehat{\mathbf{H}}$ are the true and estimated channels, respectively.

### B. Dual CNN

CNN-based CE treats the channels as pictures and learns the features in the SF and the AD domains. However, the performance of the existing CNN receivers is limited by complexity. Table II shows the relationship between the complexity and the NMSE performance under the urban micro scenario. The low-complexity SFCNN, ADCNN, and dual CNN have eight filters and four hidden layers, and they are compared with moderate- and high-complexity versions. The moderate-complexity versions increase the number of hidden layers from four to eight, and then the high-complexity versions increase the number of filters from eight to 128. Thus, the three networks have similar complexities.

For the same CNN architecture, increasing the number of filters demonstrates improved performance because the network can learn additional features. The high-complexity dual CNN is slightly better than LMMSE around its trained SNR, i.e., 10 dB. The high-complexity ADCNN has almost the same performance as the highly complex dual CNN when SNR $\geq$ 5 dB, whereas the moderate-complexity ADCNN reaches the performance of the moderate-complexity dual CNN when SNR $\geq$ 10 dB. This phenomenon indicates that ADCNN can be easily affected by noise power, especially when the complexity is low. By contrast, SFCNN is robust to the change of the noise power. The low- and moderate-complexity SFCNN has similar NMSE performance as RLMMSE and surpasses the low- and moderate-complexity ADCNN when SNR $\leq$ 5 dB.

However, increasing the number of hidden layers does not always bring performance gain. The increased number of layers usually increases the receptive field, which is essential in the SF domain to learn the adjacent areas' correlation but may not be useful for the sparse channel clusters in the AD domain. Thus, moderate-complexity ADCNN has no improvement compared with low-complexity ADCNN. Overall,

TABLE I. Settings of the proposed CNNs.

| Networks | Modules | Layer | Output dimensions | Activation function |
|---|---|---|---|---|
| Dual CNN | INPUT | $\widehat{\mathbf{H}}_{\mathrm{LS}}$ | (32,32,2) | / |
| | SFCNN | Conv1 | (32,32,8) | LReLU |
| | | Conv2 | (32,32,8) | LReLU |
| | | Conv3 | (32,32,2) | None |
| | | Conv3 output + $\widehat{\mathbf{H}}_{\mathrm{LS}}$ | (32,32,2) | / |
| | TRANS1 | $D(\cdot)$ | (32,32,2) | / |
| | ADCNN | Conv4 | (32,32,8) | LReLU |
| | | Conv5 | (32,32,8) | LReLU |
| | | Conv6 | (32,32,2) | None |
| | | Conv6 output + Trans1 output | (32,32,2) | / |
| | TRANS2 | $D^{-1}(\cdot)$ | (32,32,2) | None |
| HyperNet | INPUT | $\widehat{\mathbf{H}}_{\mathrm{LS}}$ | (32,32,2) | / |
| | CNN | Conv1+ Downsampling | (16,16,8) | LReLU |
| | | Conv2+ Downsampling | (8,8,16) | LReLU |
| | | Conv3+ Downsampling | (4,4,32) | LReLU |
| | RESHAPE | / | 512 | / |
| | DNN | Fully connected | 3 | Sigmoid |
| TimeNet | SF($\mathbf{TN}_{\mathrm{SF},i+1}$) | $[\widehat{\mathbf{H}}_{\mathrm{LS},i}, \mathbf{TN}_{\mathrm{SF},i}]$ | (32,32,4) | / |
| | | Conv1 | (32,32,8) | LReLU |
| | | Conv2 | (32,32,2) | None |
| | AD($\mathbf{TN}_{\mathrm{AD},i+1}$) | $[D(\widehat{\mathbf{H}}_{\mathrm{LS},i}), \mathbf{TN}_{\mathrm{AD},i}]$ | (32,32,4) | / |
| | | Conv3 | (32,32,8) | LReLU |
| | | Conv4 | (32,32,2) | None |

* The dual CNN has SFCNN and ADCNN modules. If SFCNN is mentioned as a CE method independently, it consists of INPUT, SFCNN modules in fact. Similarly, an independent CE method called ADCNN consists of Input, TRANS1, ADCNN and TRANS2 modules. Their numbers of filters and hidden layers are also adjusted for the comparison under different complex versions in Table. II.

TABLE II. The NMSE performance of the CNN based channel estimation with different number of filters in the hidden layers.

| | NMSE(dB) / SNR(dB) / Complexity | 0 | 5 | 10 | 15 |
|---|---|---|---|---|---|
| SFCNN | High (128 filters, 8 hidden layers) | -7.7 | -14.5 | -19.6 | -23.6 |
| | Moderate (8 filters, 8 hidden layers) | -8.7 | -14.3 | -18.9 | -22.5 |
| | Low (8 filters, 4 hidden layers) | -8.0 | -13.4 | -17.9 | -21.6 |
| ADCNN | High (128 filters, 8 hidden layers) | -6.8 | -17.1 | **-22.5** | **-26.9** |
| | Moderate (8 filters, 8 hidden layers) | -2.9 | -12.6 | -20.6 | -24.2 |
| | Low (8 filters, 4 hidden layers) | -3.6 | -13.0 | -20.5 | -24.2 |
| Dual CNN | High (128 filters, 8 hidden layers) | **-10.5** | **-17.4** | **-22.5** | -26.7 |
| | Moderate (8 filters, 8 hidden layers) | -9.7 | -16.4 | -21.8 | -25.8 |
| | Low (8 filters, 4 hidden layers) | -9.4 | -16.1 | -21.0 | -24.4 |
| RLMMSE | / | -7.4 | -12.4 | -17.4 | -22.4 |
| LMMSE | / | -12.22 | -17.1 | -22.2 | -27.1 |

the dual CNN always shows the best performance if their complexities are similar.

In the following, the low-complexity dual CNN is studied further. As shown in Fig. 6, the dual CNN is compared with the SFCNN and the ADCNN. Although they have the same number of hidden layers and filters, the dual CNN converges faster because the dual CNN has a smaller network size in each domain. Meanwhile, the domain transform modules exploit the expert knowledge to help the dual CNN learn features quickly. The ADCNN converges as fast as the SFCNN when training epochs < 200 but the ADCNN can reach better NMSE performance under the training SNR, i.e., 10 dB.

To investigate the denoising performance of different methods, the power distribution in the AD domain is displayed using gray images, and the sparsity of the channel power in Fig. 7(a) helps explain the noise power distribution after networks. In this simulation, SNR is set as 10 dB; thus, SFCNN is worse than ADCNN, whereas dual CNN is the best. The noise after SFCNN in Fig. 7(b) still has power in the green circle, where the delay is larger than six. This result means that SFCNN has no global insight because the max delay is the most critical feature exploited by RLMMSE. However, SFCNN can learn the correlation of antennas and reduce the noise in green circles, thus has a similar performance as RLMMSE. ADCNN, as shown in Fig. 7(c), removes more noise through the AD domain by detecting the channel power because most of the noise power and the channel power do not overlap. Its noise power is much less than that of SFCNN when the delay is larger than six. However, ADCNN also concentrates on local features, and a large noise power may

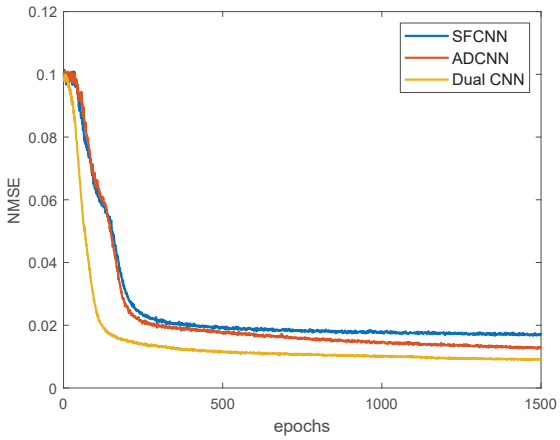



Fig. 6. Training process of the three competing CNN-based methods.

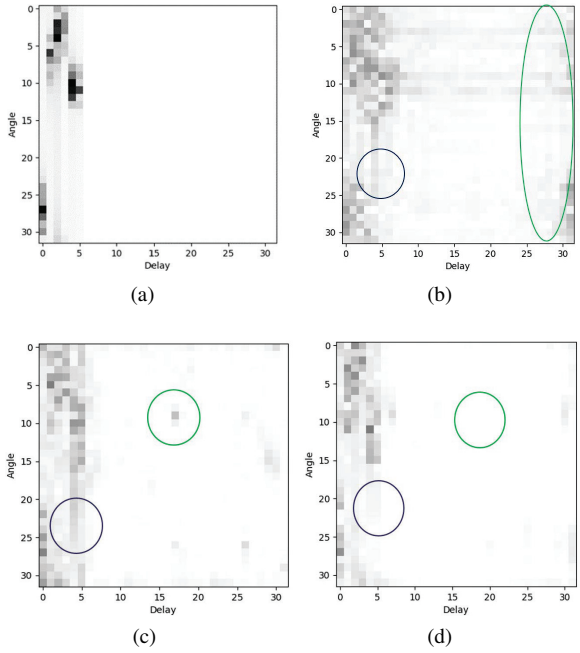
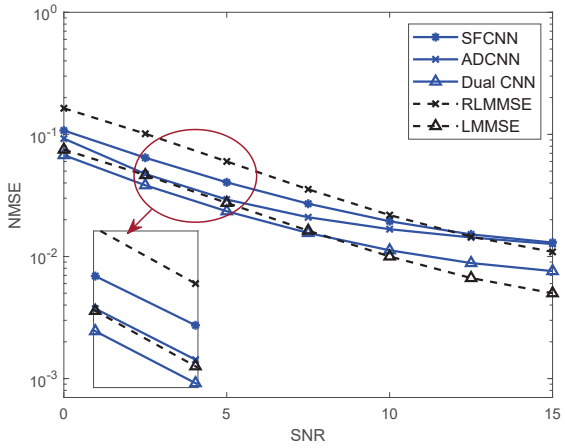



Fig. 7. (a) Channel power distribution. (b) Noise power after SFCNN. (c) Noise power after ADCNN. (d) Noise power after dual CNN. The blue circles emphasize the different reserved noise power at delay $\leq 6$, whereas the green circles compare the denoising performance at delay $> 6$.

be regarded as the channel power by mistake as in the green circle, which can be removed by SFCNN. Error detection explains the poor performance of ADCNN under low SNR. Thus, dual CNN can improve its performance because the denoising of SFCNN reduces the possibility of error detection in ADCNN.

The performance of SFCNN is worse than that of ADCNN because of the white noise and the channel overlap in the SF domain. However, when pilot contamination is considered, the superiority of the SFCNN is observed because the interference spreads over the SF domain. SFCNN can patch these corrupted



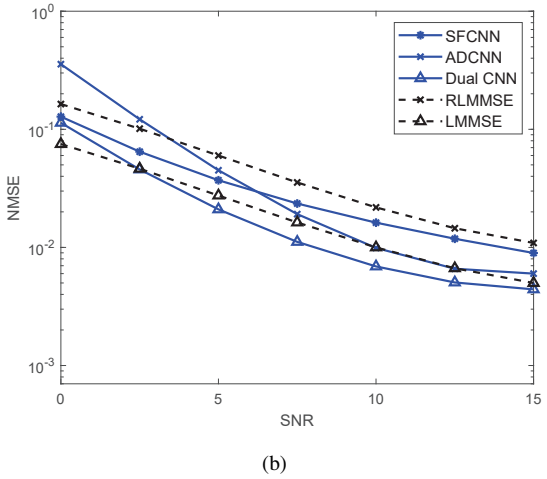




Fig. 8. NMSE performance under pilot contamination. (a) Training SNR is 5 dB. (b) Training SNR is 10 dB.

areas in the SF domain due to the correlation of adjacent areas. In Fig. 8(a), we train the three networks under SNR=5 dB. Dual CNN still outperforms the other two methods and is better than LMMSE when SNR $\leq$ 7 dB. SFCNN is also nearly 3 dB better than RLMMSE when the SNR is 0 dB. This result demonstrates that DL-based methods can outperform conventional methods under interference. ADCNN is better than SFCNN when SNR is low and the gap becomes smaller with the increase in SNR. This phenomenon means that ADCNN mistakenly takes the channel power as noise when trained under low SNR. In Fig. 8(b), ADCNN surpasses SFCNN at SNR=6 dB, whereas ADCNN surpasses SFCNN at SNR=5 dB under white noise in Table II. Therefore, SFCNN is better when handling pilot contamination. Hence, the gap between ADCNN and dual CNN is large. Besides, dual CNN is better than LMMSE under an SNR of 5-12 dB.

The above tests verify that the proposed dual CNN is always better than the CNN methods in a single domain when the complexity is similar. The channel features in different domains facilitate the estimation and dual CNN combines their advantages.

### C. Robustness Analysis and Performance of Hyper Dual CNN

To test the robustness of the networks, we analyze the noise power distribution in Fig. 9. The networks are trained under the urban micro scenario, tested under the suburban macro scenario, and SNR=15 dB. When the training scenario and the test scenario are the same, the performance of ADCNN is close to dual CNN. However, when the scenario is mismatched, the denoising performance of the SFCNN is weak, as shown in Fig. 9(a). The change of the correlation of the channel has excellent effects on SFCNN. By contrast, ADCNN still works well due to the sparsity of the channel in the AD domain. Dual CNN becomes worse than ADCNN under the error noise introduced by the mismatched SFCNN. The error cannot be distinguished from the channel, especially when they spread at the same area, as in the circle in Figs. 9(b) and (c).

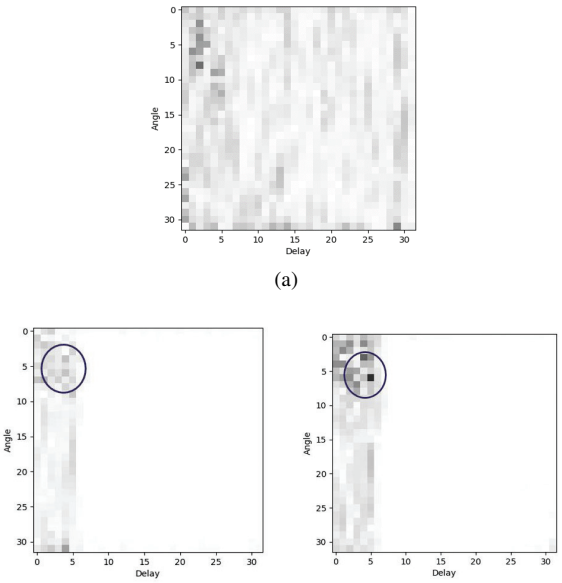



Fig. 9. Noise power distribution under the mismatch scenario. (a) Noise power after SFCNN. (b) Noise power after ADCNN. (c) Noise power after dual CNN.

The proposed hyper dual CNN can solve the above issue. The training strategy is described in Section III B. In Fig. 10(a), we simulate an untrained scenario for the above networks by increasing the max delay from 6 to 12 paths. The larger max delay spread causes a serious error because RLMMSE and the DNN ignore the channel power when the delay is larger than six. Refined RLMMSE estimates $l_{max}$ online and updates its CE matrix $\mathbf{W}_{\text{RLMMSE}}$ automatically. Refined LMMSE needs to recalculate $\mathbf{R}$ and $\mathbf{W}_{\text{RLMMSE}}$. Although Refined LMMSE achieves the best performance in Fig. 10(a), its complexity is much larger than other methods. The hyper dual CNN works without any online training and performs better than the refined RLMMSE when SNR $\leq$ 15 dB. The adaptive parameters $\boldsymbol{\alpha}$ are close to [0 0 0] when SNR $\geq$ 5 dB, indicating that SFCNNs cannot deal with untrained scenarios and only ADCNN is chosen for use. This phenomenon also illustrates the weakness of ADCNN in Fig. 7(c), where the large noise power is regarded as a channel power, which is

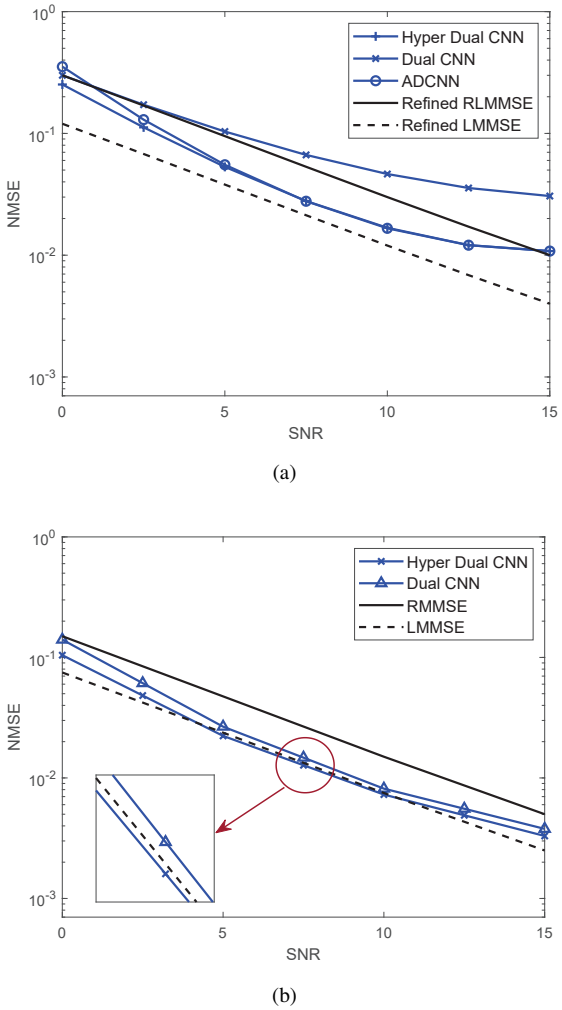



Fig. 10. NMSE performance of hyper dual CNN. (a) Under the untrained scenario with longer delay spread. (b) Under the mixed channels of the three scenarios.

beneficial for robustness. The channel power should never be ignored by ADCNN even though it appears out of the assumption in the offline training set. The hyper dual CNN is better than ADCNN when SNR $\leq$ 5 dB. Thus, HyperNet combines the existing SFCNNs and brings performance gain when the noise power is large. For example, the output of HyperNet $\boldsymbol{\alpha}$ is [0.141 0.042 0.007] when SNR=0 dB.

Apart from the untrained scenario, the hyper dual CNN is designed to improve the performance when the CE method needs to face many different scenarios. For comparison, the dual CNN, which is the traditional method to improve robustness, is trained under the mixed training set of three scenarios. In Fig. 10 (b), when the training set and the test set are matched, HyperNet helps improve performance under each scenario. Dual CNN without HyperNet sacrifices nearly 1 dB NMSE performance to balance the estimation precision of the three scenarios. LMMSE obtains its $\mathbf{R}$ from the mixed scenarios and performs a little worse than that in a single scenario, as shown in Table II; thus, the hyper dual CNN surpasses the LMMSE when SNR= 7- 10 dB.

The robustness analysis explains that mismatch happens when the scenario changes. We use additional SFCNNs to reduce the dependence on the channel correlation, and the hyper dual CNN is introduced to adapt to the changing environments without online training. The proposed architecture and training strategy are low in complexity and perform well under the trained and untrained scenarios.

### *D. Performance of Dual RNN*

The dual RNN delivering $\mathbf{TN}_{\mathrm{SF},i}$ and $\mathbf{TN}_{\mathrm{AD},i}$ is always better than the RNN only delivering in one domain. Meanwhile, using the CE from the output of the prior dual CNN, $\widehat{\mathbf{H}}_i$, as the input of TimeNet is better than the LS estimation, $\mathbf{H}_{\mathrm{LS,SF},i}$. However, $\widehat{\mathbf{H}}_i$ brings low robustness under the mismatch scenario because the estimation error is delivered by the RNN framework. Thus, LS estimation is a robust choice as the input of TimeNet. Apart from dual RNN, we also combine TimeNet, HyperNet, and the dual CNN, called hyper dual RNN. Additional TimeNets are not necessary because the changing scenarios have minimal effect on the hyper dual RNN. However, the changing temporal correlation is a challenge. The temporal correlation, which is affected by the velocity of the movements, changes continuously.

We assume the upper limit; thus, the channels are correlated in 10 blocks at least. The temporal correlation in the training set continuously changes from the assumption bound to static. Fig. 11(a) compares the performance of the hyper dual CNN. The test environment is the same as Fig. 10(a), where three scenarios are mixed. Dual RNN (T) means that the dual RNN is trained and tested exactly under the same environment, and $T$ blocks are correlative. The hyper dual RNN is trained for the environment where scenarios and temporal correlation change. Then, the hyper dual RNN ($T$) is tested under the mixed scenarios, and $T$ blocks are correlated. The hyper dual RNN ($T$ = 10) is almost 2 dB worse than the dual RNN ($T$ = 10), and the performance gap is larger when the correlated blocks is 50. The hyper dual RNN loses performance to adapt to the changing environment. The number of relative blocks has a slight influence on the hyper dual RNN, so it is more robust than the dual RNN without any online redefinition.

We also test the proposed networks with the insufficient pilot in Fig. 11(b). Each user uses 1/4 subcarriers to transmit pilots; thus, four users occupy an OFDM symbol of 32 subcarriers in total. Scatter pilots mean a user occupies different groups of eight subcarriers in contiguous blocks so that all the subcarriers are estimated once by pilots every four blocks. The dual CNN performs better than LMMSE when SNR $\leq$ 10 dB, demonstrating the superiority of the AI-aided methods under extreme environments. The dual RNN achieves better performance than the dual CNN and always surpasses LMMSE. The scatter pilots help the dual RNN to reduce the interpolation error and improve the performance further.

Finally, the performance of the proposed methods with different BS antennas is shown in Table III. They are both trained and tested under the scenarios of urban micro. The training SNR is 10 dB, and 10 blocks are correlated in the test set. The pilot is sufficient, and only the white noise exists,

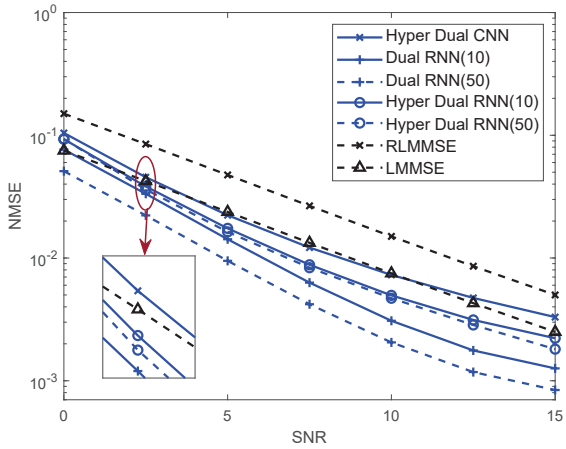



Fig. 11. (a) NMSE performance of different RNN designs. (b) NMSE performance under insufficient pilots.

so the dual CNN cannot surpass LMMSE. The dual RNN has a chance to perform better than LMMSE because it exploits temporal correlation. RLMMSE has the lowest complexity, and it only exploits correlation in the frequency domain; thus, its NMSE performance remains unchanged under different antennas. The LMMSE and the proposed methods always perform better with the increase of antennas. The dual CNN and the dual RNN under high SNR increase more than that under low SNR. The dual RNN has almost no performance gain under 0 dB SNR when the number of BS antennas $M$ is increased from 32 to 64. In contrast, the dual RNN surpasses the LMMSE under 15 dB SNR when $M$=64, but its performance is worse than the LMMSE when $M$=32 and $M$=48. This phenomenon is due to the better sparsity in AD domain with the increase of antennas. Thus, the dual CNN can remove more noise in the AD domain when SNR is high. However, the dual CNN and RNN are weak in handling a large noise power in the AD domain, which restrains their performance when SNR is low.

In the abovementioned test, the dual RNN exploits the temporal correlation and thus obtains better performance. The hyper dual RNN is proposed to address the changing

TABLE III. NMSE performance of the proposed methods under different number of BS antennas.

| | NMSE(dB) / SNR(dB) / Methods | 0 | 5 | 10 | 15 |
|---|---|---|---|---|---|
| M=64 | Dual RNN | -9.6 | **-19.0** | **-25.0** | **-28.8** |
| | Dual CNN | -9.8 | -17.5 | -23.2 | -27.0 |
| | LMMSE | **-13.9** | -18.7 | -23.9 | -28.7 |
| M=48 | Dual RNN | -9.4 | **-18.2** | **-23.8** | -27.3 |
| | Dual CNN | -9.6 | -16.7 | -21.9 | -25.5 |
| | LMMSE | **-13.0** | -17.8 | -23.0 | **-27.8** |
| M=32 | Dual RNN | -9.2 | **-17.6** | **-22.8** | -26.2 |
| | Dual CNN | -9.4 | -16.1 | -21.0 | -24.4 |
| | LMMSE | **-12.2** | -17.1 | -22.2 | **-27.1** |
| RLMMSE | / | -7.4 | -12.4 | -17.4 | -22.4 |

environments, and it also outperforms the conventional MMSE methods and CNN-based methods.

### *E. Complexity Analysis*

TABLE IV. Forward complexity analysis for proposed networks and competing methods.

| | FLOPs | Parameters |
|---|---|---|
| SFCNN | 4.1M | 2050 |
| ADCNN | 4.3M | 2050 |
| Dual CNN | 3.7M | 1764 |
| Dual RNN | 6.1M | 2936 |
| HyperNet | 0.055M | 4584 |
| RLMMSE | 0.16M | / |
| LMMSE | 8.4M | / |
| One Layer DNN | 8.4M | 4.2M |

Table IV compares the complexity of the number of floating-point multiplication-adds (FLOPs) [44] and parameters to estimate a channel for each user when $M = K = 32$. SFCNN, ADCNN, and dual CNN with the same number of hidden layers have similar complexities. Although the complexity of the CNN-based methods only relates to the first order of $M$ and $K$, the choice of filter size and input and output channels is also essential. Three CNN-based methods still cost approximately 1/2 of hardware resources compared with the LMMSE. However, dual CNN is better than the other two CNN-based methods and approaches the performance of LMMSE under white noise. When considering pilot contamination and insufficient pilot, CNN-based CE can surpass LMMSE. Furthermore, when the scenarios vary, the LMMSE costs impractical high complexity, whereas CNN-based methods can be enhanced with HyperNet, which only costs minimal resources. The dual RNN exploits the temporal correlation but its complexity is still lower than that of LMMSE. With the help of temporal correlation, the dual RNN performs better than LMMSE under white noise, let alone under insufficient pilots and pilot contamination conditions. The one-layer DNN is common when designing a simple DNN-based CE method [21], [23] because an end-to-end DNN costs too much time and resources to train. The number of its trainable parameters is $O((MK)^2)$; thus, DNN-based architecture is hard to realize when antennas and subcarriers are large. Moreover, 4.2 M parameters are impossible to refine under changing scenarios.

Overall, complexity analysis suggests that proposed networks cost fewer resources than the conventional methods and competing DL-based architectures. Moreover, the introduced HyperNet and training strategy improves the robustness without online refining.

## V. Conclusions

In this paper, we first developed a CNN-based CE called dual CNN to take advantage of in the SF and AD domains. The channel's sparsity in the AD domain enables the CNN to handle most of the white noise, whereas the channel correlation in the SF domain helps ease interference. The SF domain's correlation also reduces the noise power so that the ADCNN has less possibility to be confused when distinguishing the channel and noise. Thus, the dual CNN has better performance and robustness than estimation in a single domain. We also introduced HyperNet, which does not require online training but performs better than the dual CNN and RLMMSE under the trained and untrained scenarios. We proposed an RNN framework to improve the CE performance by exploiting the temporal correlation of adjacent OFDM blocks. This framework is initiated with a trained dual CNN and learns to perform better than dual CNN. The robust design in this framework stabilizes its performance as long as the temporal correlation is larger than the assumption in the training set.

## References


[1] 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.

[2] Y. Li, L. J. Cimini, and N. R. Sollenberger, "Robust channel estimation for OFDM systems with rapid dispersive fading channels," *IEEE Trans. Commun.*, vol. 46, no. 7, pp. 902–915, Jul. 1998.

[3] O. Edfors, M. Sandell, J.-J. Van de Beek, S. K. Wilson, and P. O. Borjesson, "OFDM channel estimation by singular value decomposition," *IEEE Trans. Commun.*, vol. 46, no. 7, pp. 931–939, Jul. 1998.

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

[5] 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.

[6] Z.-J. Qin, H. Ye, G. Y. Li, and B.-H. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, Apr. 2019.

[7] H. He, S. Jin, C. 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.

[8] J. Kang, C. Chun, and I. Kim, "Deep-learning-based channel estimation for wireless energy transfer," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2310–2313, Nov. 2018.

[9] E. Balevi and J. G. Andrews, "One-bit OFDM receivers via deep learning," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4326–4336, Jun. 2019.

[10] S. Gao, P. Dong, Z. Pan, and G. Y. Li, "Deep learning based channel estimation for massive MIMO with mixed-resolution ADCs," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1989–1993, Nov. 2019.

[11] Q. Hu, F. Gao, H. Zhang, S. Jin, and G. Y. Li, "Deep learning for MIMO channel estimation: Interpretation, performance, and comparison," *arXiv preprint arXiv:1911.01918*, 2019.

[12] Y. Zhang, M. Alrabeiah, and A. Alkhateeb, "Deep learning for massive MIMO with 1-bit ADCs: When more antennas need fewer pilots," *IEEE Wireless Commun. Lett.*, vol. 9, no. 8, pp. 1273–1277, Aug. 2020.

[13] H. He, C. 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.

[14] M. Mehrabi, M. Mohammadkarimi, M. Ardakani, and Y. Jing, "Decision directed channel estimation based on deep neural network $k$ -step predictor for MIMO communications in 5G," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2443–2456, Aug. 2019.

[15] C. Chun, J. Kang, and I. Kim, "Deep learning-based channel estimation for massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1228–1231, Apr. 2019.

[16] C.-J. Chun, J.-M. Kang, and I.-M. Kim, "Deep learning-based joint pilot design and channel estimation for multiuser MIMO channels," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1999–2003, Nov. 2019.

[17] A. M. Elbir and K. V. Mishra, "Online and offline deep learning strategies for channel estimation and hybrid beamforming in multi-carrier mm-Wave massive MIMO systems," *arXiv preprint arXiv:1912.10036*, 2019.

[18] W. Ma, C. Qi, Z. Zhang, and J. Cheng, "Sparse channel estimation and hybrid precoding using deep learning for millimeter wave massive MIMO," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2838–2849, May 2020.

[19] H. Ye, L. Liang, G. Y. Li, and B.-H. Juang, "Deep learning-based end-to-end wireless communication systems with conditional GANs as unknown channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3133–3143, May 2020.

[20] 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, Jun. 2018.

[21] X. Gao, S. Jin, C.-K. Wen, and G. Y. Li, "ComNet: Combination of deep learning and expert knowledge in OFDM receivers," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2627–2630, Dec. 2018.

[22] M. van Lier, A. Balatsoukas-Stimming, H. Corporaaal, and Z. Zivkovic, "OptComNet: Optimized neural networks for low-complexity channel estimation," *arXiv preprint arXiv:2002.10493*, 2020.

[23] J. Liu, K. Mei, X. Zhang, D. Ma, and J. Wei, "Online extreme learning machine-based channel estimation and equalization for OFDM systems," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1276–1279, Jul. 2019.

[24] P. Jiang, T. Wang, B. Han, X. Gao, J. Zhang, C.-K. Wen, S. Jin, and G. Y. Li, "Artificial intelligence-aided OFDM receiver: Design and experimental results," *arXiv preprint arXiv:1812.06638*, 2018.

[25] H. Mao, H. Lu, Y. Lu, and D. Zhu, "RoemNet: Robust meta learning based channel estimation in OFDM systems," in *Proc. IEEE Int. Conf. Commun. (ICC), Shanghai, China*, May 2019.

[26] S. Park, O. Simeone, and J. Kang, "End-to-end fast training of communication links without a channel model via online meta-learning," *arXiv preprint arXiv:2003.01479*, 2020.

[27] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 652–655, Apr. 2019.

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

[29] E. Balevi, A. Doshi, and J. G. Andrews, "Massive MIMO channel estimation with an untrained deep neural network," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2079–2090, Jan. 2020.

[30] Z. Gao, L. Dai, S. Han, C. I, Z. Wang, and L. Hanzo, "Compressive sensing techniques for next-generation wireless communications," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 144–153, Jun. 2018.

[31] S. Liu, Z. Gao, J. Zhang, M. D. Renzo, and M. S. Alouini, "Deep denoising neural network assisted compressive channel estimation for mmWave intelligent reflecting surfaces," *IEEE Trans. on Veh. Technol.*, vol. 69, no. 8, pp. 9223–9228, Aug. 2020.

[32] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Convolutional neural network-based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2827–2840, Apr. 2020.

[33] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 4898–4906.

[34] K.-C. Hung and D. W. Lin, "Pilot-based LMMSE channel estimation for OFDM systems with power-delay profile approximation," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 150–159, Jan. 2010.

[35] J. Kang, C. Chun, and I. Kim, "Deep learning based channel estimation for MIMO systems with received SNR feedback," *IEEE Access*, vol. 8, pp. 121 162–121 181, 2020.

[36] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Syst.*, Dec 2017, pp. 5998–6008.

[38] C. Wen, S. Jin, K. Wong, J. Chen, and P. Ting, "Channel estimation for massive MIMO using gaussian-mixture bayesian learning," *IEEE Trans. on Wireless Commun.*, vol. 14, no. 3, pp. 1356–1368, Oct. 2015.

[39] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[40] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep HyperNetwork-based MIMO detection," *arXiv preprint arXiv:2002.02750*, 2020.

[41] D. S. Baum, J. Hansen, J. Salo, G. Del Galdo, M. Milojevic, and P. Kyösti, "An interim channel model for beyond-3G systems: extending the 3GPP spatial channel model (SCM)," in *Proc. IEEE 61st Vehicular Technology Conference 2005 (VTC2005-Spring)*, vol. 5, 2015, pp. 3132–3136.

[42] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *MIMO-OFDM Wireless Communications with MATLAB*. John Wiley & Sons, 2010.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[44] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *arXiv preprint arXiv:1611.06440*, 2016.


**Peiwen Jiang** received his B.S. degree from Southeast University, Nanjing, China in 2019. He is currently working towards his Ph.D. degree in information and communications engineering, Southeast University, China. His research interests include deep learning based channel estimation, signal detection, and semantic transmission in communications.

**Chao-Kai Wen** (Member, IEEE) received the Ph.D. degree from the Institute of Communications Engineering, National Tsing Hua University, Taiwan, in 2004. He was with Industrial Technology Research Institute, Hsinchu, Taiwan and MediaTek Inc., Hsinchu, Taiwan, from 2004 to 2009, where he was engaged in broadband digital transceiver design. Since 2009, he jointed the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, where he is currently Professor. His research interests center around the optimization in wireless multimedia networks.

**Shi Jin** (Senior Member, IEEE) received the B.S. degree in communications engineering from Guilin University of Electronic Technology, Guilin, China, in 1996, the M.S. degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2003, and the Ph.D. degree in information and communications engineering from the Southeast University, Nanjing, in 2007. From June 2007 to October 2009, he was a Research Fellow with the Adastral Park Research Campus, University College London, London, U.K. He is currently with the faculty of the National Mobile Communications Research Laboratory, Southeast University. His research interests include space time wireless communications, random matrix theory, and information theory. He served as an Associate Editor for the IEEE Transactions on Wireless Communications, and IEEE Communications Letters, and IET Communications. Dr. Jin and his co-authors have been awarded the 2011 IEEE Communications Society Stephen O. Rice Prize Paper Award in the field of communication theory and a 2010 Young Author Best Paper Award by the IEEE Signal Processing Society.

**Geoffrey Ye Li** (Fellow, IEEE) has been a Chair Professor at Imperial College London, UK, since 2020. Before moving to Imperial, he was with Georgia Institute of Technology in Georgia, USA, as a Professor for twenty years and with AT&T Labs - Research in New Jersey, USA, as a Principal Technical Staff Member for five years. His general research interests include statistical signal processing and machine learning for wireless communications. In the related areas, he has published over 500 journal and conference papers in addition to over 40 granted patents. His publications have been cited over 46,000 times and he has been recognized as a Highly Cited Researcher, by Thomson Reuters, almost every year. Dr. Geoffrey Ye Li was awarded IEEE Fellow for his contributions to signal processing for wireless communications in 2005. He won several prestigious awards from IEEE Signal Processing Society (Donald G. Fink Overview Paper Award in 2017), IEEE Vehicular Technology Society (James Evans Avant Garde Award in 2013 and Jack Neubauer Memorial Award in 2014), and IEEE Communications Society (Stephen O. Rice Prize Paper Award in 2013, Award for Advances in Communication in 2017, and Edwin Howard Armstrong Achievement Award in 2019). He also received the 2015 Distinguished ECE Faculty Achievement Award from Georgia Tech. He has been involved in editorial activities for over 20 technical journals, including the founding Editor-in-Chief of IEEE JSAC Special Series on ML in Communications and Networking. He has organized and chaired many international conferences, including technical program vice-chair of the IEEE ICC'03, general co-chair of the IEEE GlobalSIP'14, the IEEE VTC'19 (Fall), and the IEEE SPAWC'20.