Deep CNN-Based Channel Estimation for mmWave Massive MIMO Systems

Peihao Dong , *Student Member, IEEE*, Hua Zhang , *Member, IEEE*, Geoffrey Ye Li, *Fellow, IEEE*, Ivan Simões Gaspar , and Navid NaderiAlizadeh

Abstract—For millimeter wave (mmWave) massive multiple-input multiple-output (MIMO) systems, hybrid processing architecture is usually used to reduce the complexity and cost, which poses a very challenging issue in channel estimation. In this paper, deep convolutional neural network (CNN) is employed to address this problem. We first propose a spatial-frequency CNN (SF-CNN) based channel estimation exploiting both the spatial and frequency correlation, where the corrupted channel matrices at adjacent subcarriers are input into the CNN simultaneously. Then, exploiting the temporal correlation in time-varying channels, a spatial-frequency-temporal CNN (SFT-CNN) based approach is developed to further improve the accuracy. Moreover, we design a spatial pilot-reduced CNN (SPR-CNN) to save spatial pilot overhead for channel estimation, where channels in several successive coherence intervals are grouped and estimated by a channel estimation unit with memory. Numerical results show that the proposed SF-CNN and SFT-CNN based approaches outperform the non-ideal minimum mean-squared error (MMSE) estimator but with reduced complexity, and achieve the performance close to the ideal MMSE estimator that is very difficult to be implemented in practical situations. They are also robust to different propagation scenarios. The SPR-CNN based approach achieves comparable performance to SF-CNN and SFT-CNN based approaches while only requires about one-third of spatial pilot overhead at the cost of complexity. The results in this paper clearly demonstrate that deep CNN can efficiently exploit channel correlation to improve the estimation performance for mmWave massive MIMO systems.

Index Terms—mmWave massive MIMO, deep CNN, channel estimation, channel correlation.

I. INTRODUCTION

MILLIMETER wave (mmWave) communications can meet the high data rate demand due to its large bandwidth. Its high propagation loss can be well compensated by using massive multiple-input multiple-output (MIMO) technique [2]−[6]. However, Due to the limited physical space with closely placed antennas and prohibitive power consumption in mmWave massive MIMO systems, it is difficult to equip a dedicated radio frequency (RF) chain for each antenna. To reduce complexity and cost, phase shifter based two-stage architecture, usually called hybrid architecture, is widely used at both the transmitter and the receiver to connect a large number of antennas with much fewer RF chains [7], [8].

For mmWave massive MIMO systems with the hybrid architecture, channel estimation is a challenging problem. In [9], a hierarchical multi-resolution codebook has been designed, based on which an adaptive channel estimation algorithm has been developed by exploiting the channel sparsity. In [10], the structured sparsity in angle domain has been utilized to estimate the wideband channel for multi-user mmWave massive MIMO uplink. In [11], a channel estimation approach has been developed for mmWave massive MIMO orthogonal frequency division multiplexing (OFDM) systems with low-precision analog-to-digital converters (ADCs). For the mmWave MIMO-OFDM downlink, a channel parameter estimation for the angles, time delays, and fading coefficients has been proposed in [12] resorting to the low-rank tensor decomposition. Instead of estimating the mmWave MIMO channel directly, the method for singular subspace estimation has been proposed in [13], based on which a subspace decomposition algorithm has been further developed to design the hybrid analog-digital architecture.

Compared to the conventional methods, machine learning (ML) is more powerful to uncover the inherent characteristics inside data/signals collected in an end-to-end manner and thus can achieve better performance when addressing various problems in wireless communications [14]. In [15], deep learning (DL) has been successfully used in joint channel estimation and signal detection of OFDM systems with interference and nonlinear distortions. In [16], iterative channel estimation has been proposed for the lens mmWave massive MIMO systems, where denoising neural network (NN) is used in each iteration to update the estimated channel. To reduce the CSI feedback overhead of the frequency duplex division (FDD) massive MIMO system, DL has been employed in [17] to compress the channel into a low
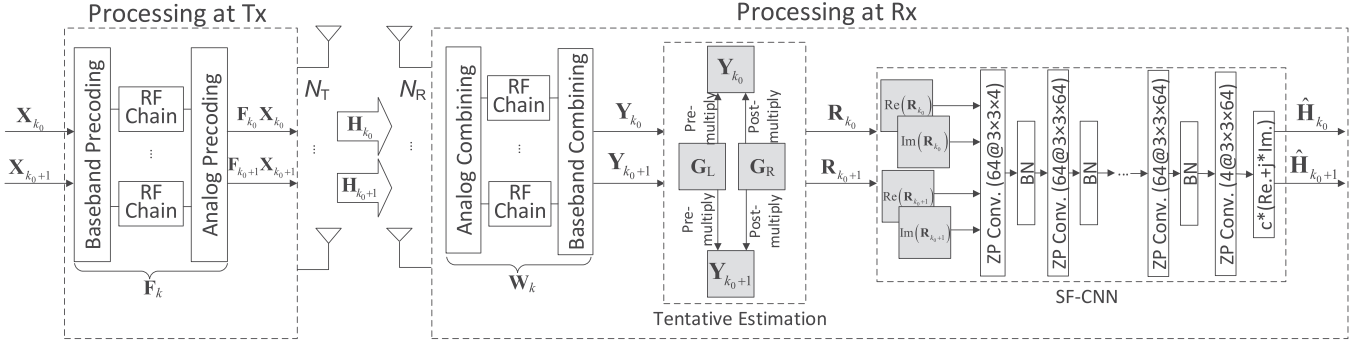
Fig. 1. SF-CNN based channel estimation.

dimensional codeword and then to perform recovery with high accuracy. Exploiting temporal correlation of the channel, long short-term memory (LSTM) based deep NN (DNN) has been introduced in [18] to develop a more efficient channel compression and recovery method for the CSI feedback. More related research can be also found in [19] and the references therein. In [20], DL has been applied to estimate channels in wireless power transfer systems, which outperforms the conventional scheme in terms of both estimation accuracy and harvested energy. In [21], supervised learning algorithms have been used to acquire the downlink CSI for FDD massive MIMO systems with reduced overheads for pilot and CSI feedback. In [22], the supervised learning has been exploited to perform blind detection for massive MIMO systems with low-precision ADCs.

The conventional channel estimation methods usually perform unsatisfactorily in the more practical complicated channel model and also suffer from high complexity. In contrast, the deep convolutional NN (CNN) is more capable to extract the inherent characteristics underlying the channel matrix from the large amount of data and provides the potential to estimate the channel more accurately with lower complexity by using the efficient parallel computing methods. In this paper, we use the deep CNN to address channel estimation for mmWave massive MIMO-OFDM systems. To exploit the correlation among channels at adjacent subcarriers in OFDM, we first propose a spatial-frequency CNN (SF-CNN) based channel estimation, where the tentatively estimated channel matrices at adjacent subcarriers are input into the CNN simultaneously [1]. To further exploit the temporal correlation, a spatial-frequency-temporal CNN (SFT-CNN) based channel estimation is developed, where the channel information of the previous coherence interval is utilized when estimating the channel matrices of the current coherence interval. The SFT-CNN based approach incorporates all types of channel correlation in a simple way and yields remarkable performance gains that can be used to significantly save the spatial pilot overhead due to large-scale arrays. Therefore, we propose a spatial pilot-reduced CNN (SPR-CNN) based channel estimation, where channels in several successive coherence intervals are grouped and estimated by a channel estimation unit (CEU) with memory. From the numerical results, the proposed SF-CNN and SFT-CNN based approaches outperform the non-ideal minimum mean-squared error (MMSE) estimator and achieve the performance very close to the ideal MMSE

estimator that is very difficult to be implemented in practical systems. They are also with lower complexity than the MMSE estimator and exhibit the unique robustness to maintain the fairly good performance when facing different channel statistics. The SPR-CNN based approach achieves comparable performance to SF-CNN and SFT-CNN based approaches by using only about one third of spatial pilot overhead and moderately increased complexity.

The rest of the paper is organized as follows. Section II describes the considered mmWave massive MIMO system, followed by the proposed SF-CNN based channel estimation in Section III. Section IV further develops the SFT-CNN and SPR-CNN based channel estimation. Numerical results are provided in Section V and finally Section VI gives concluding remarks.

Notations: In this paper, we use upper and lower case bold-face letters to denote matrices and vectors, respectively. $\| \cdot \|_F$, $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^{-1}$, and $\mathbb{E}\{\cdot\}$ represent the Frobenius norm, transpose, conjugate transpose, inverse, and expectation, respectively. $\mathcal{CN}(\mu, \sigma^2)$ represents circular symmetric complex Gaussian distribution with mean μ and variance σ^2. $\delta(\cdot)$ and $\lceil \cdot \rceil$ denote the delta function and ceiling function. $\mathbb{N}_+$ denotes the set of all positive integers.

II. SYSTEM MODEL

We consider a mmWave massive MIMO-OFDM system as in Fig. 1, where the transmitter is with N_T antennas and N_T^{RF} RF chains and the receiver is with N_R antennas and N_R^{RF} RF chains. Phase shifters are employed to connect a large number of antennas with a much fewer number of RF chains at both the transmitter and the receiver sides. We therefore assume $N_T \gg N_T^{RF}$ and $N_R \gg N_R^{RF}$.

According to [10], the $N_R \times N_T$ channel matrix between the receiver and the transmitter in the delay domain is given by

$$\mathbf{H}(\tau) = \sqrt{\frac{N_T N_R}{L}} \sum_{l=1}^{L} \alpha_l \delta(\tau - \tau_l) \mathbf{a}_R(\varphi_l) \mathbf{a}_T^H(\phi_l), \quad (1)$$

where L is the number of main paths, $\alpha_l \sim \mathcal{CN}(0, \sigma_\alpha^2)$ is the propagation gain of the lth path with σ_α^2 being the average power gain, τ_l is the delay of the lth path, φ_l and $\phi_l \in [0, 2\pi]$ are the azimuth angles of arrival and departure (AoA/AoD) at the receiver and the transmitter, respectively. For uniform linear array

(ULA), the corresponding response vectors can be expressed as

$$\mathbf{a}_{\text{R}}(\varphi_l) = \frac{1}{\sqrt{N_{\text{R}}}} \left[1, e^{-j2\pi \frac{d}{\lambda} \sin(\varphi_l)}, \dots, e^{-j2\pi \frac{d}{\lambda}(N_{\text{R}}-1)\sin(\varphi_l)} \right]^T,$$
(2)

$$\mathbf{a}_{\text{T}}(\phi_l) = \frac{1}{\sqrt{N_{\text{T}}}} \left[1, e^{-j2\pi \frac{d}{\lambda} \sin(\phi_l)}, \dots, e^{-j2\pi \frac{d}{\lambda}(N_{\text{T}}-1)\sin(\phi_l)} \right]^T,$$
(3)

where d and λ denote the distance between the adjacent antennas and carrier wavelength, respectively.

According to the channel model in (1), the frequency domain channel of the kth subcarrier in OFDM is given by

$$\mathbf{H}_k = \sqrt{\frac{N_{\text{T}}N_{\text{R}}}{L}} \sum_{l=1}^{L} \alpha_l e^{-j2\pi \tau_l f_s \frac{k}{K}} \mathbf{a}_{\text{R}}(\varphi_l)\mathbf{a}_{\text{T}}^H(\phi_l), \quad (4)$$

where f_s denotes the sampling rate and K is the number of OFDM subcarriers.

To estimate $\mathbf{H}_k$, the transmitter activates only one RF chains to transmit the pilot on one beam in each channel use while the receiver combines the received pilot by using all RF chains associated with different beams. In more detail, the transmitter transmits pilots, $x_{k,u}$, using M_{T} beamforming vectors, $\mathbf{f}_{k,u} \in \mathbb{C}^{N_{\text{T}}\times 1}$, $u = 1, \dots, M_{\text{T}}$. For the transmit pilot signal corresponding to each beamforming vector, $\mathbf{f}_{k,u}$, the receiver employs M_{R} combining vectors, $\mathbf{w}_{k,v} \in \mathbb{C}^{N_{\text{R}}\times 1}$, $v = 1, \dots, M_{\text{R}}$, to respectively process it. Since the receiver is equipped with $N_{\text{R}}^{\text{RF}}(< M_{\text{R}})$ RF chains, it can only use N_{R}^{RF} combining vectors in a channel use. Then, if the receiver uses all M_{R} combining vectors to process a beamforming vector carrying pilot, the required channel uses will be $\lceil \frac{M_{\text{R}}}{N_{\text{R}}^{\text{RF}}} \rceil$. So the total channel uses for processing all beamforming vectors will be $M_{\text{T}}\lceil \frac{M_{\text{R}}}{N_{\text{R}}^{\text{RF}}} \rceil$.[1] Then the pilot signal matrix associated with the kth subcarrier at the baseband of the receiver can be written as

$$\mathbf{Y}_k = \mathbf{W}_k^H \mathbf{H}_k \mathbf{F}_k \mathbf{X}_k + \tilde{\mathbf{N}}_k, \quad (5)$$

where $\mathbf{W}_k = [\mathbf{w}_{k,1}, \dots, \mathbf{w}_{k,M_{\text{R}}}]$ and $\mathbf{F}_k = [\mathbf{f}_{k,1}, \dots, \mathbf{f}_{k,M_{\text{T}}}]$ are combining matrix and beamforming matrix, respectively, $\mathbf{X}_k$ is an $M_{\text{T}} \times M_{\text{T}}$ diagonal matrix with its uth diagonal element being $x_{k,u}$. $\tilde{\mathbf{N}}_k = \mathbf{W}_k^H \mathbf{N}_k$ denotes the effective noise after combining and $\mathbf{N}_k$ is additive white Gaussian noise (AWGN) with $\mathcal{CN}(0, 1)$ elements before combining.

We consider the pilot insertion in both frequency and time domain. Specifically, adjacent Q $(Q \geq 2)$ subcarriers respectively place pilots with the same length at the beginning of a coherence interval to form a pilot subcarrier block and the rest of time slots in each coherence interval are used for data transmission. Two

---

[1]This pilot transmission process can capture the main paths in mmWave channels. Although simultaneously activating multiple RF chains with different beams at the transmitter can accelerate the pilot transmission process, it fails to capture the main paths and leads to poor channel estimation performance. Thus in lots of related work, only one RF chain is activated to transmit the pilot on one beam at the transmitter during each channel use to guarantee the performance of channel estimation algorithms [9], [11]. Similarly, in this paper, the pilot transmission mode that we consider also facilitates the CNN based channel estimation to achieve very good accuracy.

pilot subcarrier blocks are separated by Q_{d} $(Q_{\text{d}} \geq 0)$ subcarriers dedicated to data transmission. Pilots are utilized to estimate the channels of corresponding time-frequency positions. Based on the estimated channels, interpolation can be applied to get the channels at the positions without pilots. It is clear that the accuracy of interpolation is dependent on the accuracy of the estimated channels and the variation of channels. Therefore, we will focus on improving the accuracy of the pilot based channel estimation in this paper so that more reliable reference values can be provided for interpolation.

III. SF-CNN-BASED CHANNEL ESTIMATION

In this section, we first elaborate the SF-CNN based channel estimation, including an overview of the proposed approach, the offline training of SF-CNN, and the online deployment. Then the computational complexity for the online estimation is analyzed.

A. Algorithm Description

1) Channel Estimation Procedure: Fig. 1 illustrates the channel estimation procedure for adjacent Q $(= 2)$ subcarriers, k_0 and $k_0 + 1$, to simplify our discussion even if it is trivial to extend to the case with $Q > 2$. Without loss of generality, we assume the worst case that $\mathbf{W}_k = \mathbf{W}$, $\mathbf{F}_k = \mathbf{F}$, and $\mathbf{X}_k = \sqrt{P}\mathbf{I}$ for $k \in \{k_0, k_0 + 1\}$, where P denotes the transmit power. The pilot signal matrix, $\mathbf{Y}_k$, becomes

$$\mathbf{Y}_k = \sqrt{P}\mathbf{W}^H \mathbf{H}_k \mathbf{F} + \tilde{\mathbf{N}}_k. \quad (6)$$

Then $\mathbf{Y}_k$ goes through the tentative estimation (TE) module, which uses two matrices, $\mathbf{G}_{\text{L}}$ and $\mathbf{G}_{\text{R}}$, to process $\mathbf{Y}_k$ and outputs a coarse estimation of $\mathbf{H}_k$, that is,

$$\mathbf{R}_k = \mathbf{G}_{\text{L}}\mathbf{Y}_k \mathbf{G}_{\text{R}} = \sqrt{P}\mathbf{G}_{\text{L}}\mathbf{W}^H \mathbf{H}_k \mathbf{F} \mathbf{G}_{\text{R}} + \mathbf{G}_{\text{L}}\tilde{\mathbf{N}}_k \mathbf{G}_{\text{R}}, \quad (7)$$

where

$$\mathbf{G}_{\text{L}} = \begin{cases} \mathbf{W}, & M_{\text{R}} < N_{\text{R}}, \\ (\mathbf{W}\mathbf{W}^H)^{-1}\mathbf{W}, & M_{\text{R}} \geq N_{\text{R}}, \end{cases} \quad (8)$$

and

$$\mathbf{G}_{\text{R}} = \begin{cases} \mathbf{F}^H, & M_{\text{T}} < N_{\text{T}}, \\ \mathbf{F}^H(\mathbf{F}\mathbf{F}^H)^{-1}, & M_{\text{T}} \geq N_{\text{T}}. \end{cases} \quad (9)$$

The tentatively estimated channel matrices $\mathbf{R}_{k_0}$ and $\mathbf{R}_{k_0+1}$ are then input into the SF-CNN simultaneously, which outputs the estimated channel matrices $\hat{\mathbf{H}}_{k_0}$ and $\hat{\mathbf{H}}_{k_0+1}$ through the mapping relationship

$$\left\{ \hat{\mathbf{H}}_{k_0}, \hat{\mathbf{H}}_{k_0+1} \right\} = f_\Phi \left(\mathbf{R}_{k_0}, \mathbf{R}_{k_0+1}; \Phi \right), \quad (10)$$

where Φ denotes the parameter set of the SF-CNN.

2) SF-CNN Offline Training: For the proposed SF-CNN, the training set consisting of N_{tr} samples is generated according to certain channel model in the simulation environment with $(\underline{\mathbf{R}}_i, \underline{\mathbf{H}}_i)$ denoting the ith sample, where $\underline{\mathbf{R}}_i$ is the input data and $\underline{\mathbf{H}}_i$ is the target data. $\underline{\mathbf{R}}_i \in \mathbb{C}^{N_{\text{R}}\times N_{\text{T}}\times 2}$ is a three-dimensional matrix composed of $\mathbf{R}_{k_0'}^i, \mathbf{R}_{k_0'+1}^i \in \mathbb{C}^{N_{\text{R}}\times N_{\text{T}}}$, which are the tentatively estimated channel matrices at subcarrier k_0' and $k_0' + 1$ collected through (7) with $k_0' \in \{1, 2, \dots, K - 1\}$. $\underline{\mathbf{H}}_i \in$

TABLE I
ARCHITECTURE OF THE SF-CNN

Layer	Type	Number of filters	Size of filters	Activation function
1	Input	-	-	-
2∼10	Conv.	64	3×3	ReLU
11	Output	$2Q$	3×3	tanh

TABLE II
SF-CNN PARAMETER SETTINGS

l	$M_{1,l}$	$M_{2,l}$	F_l	N_{l-1}	N_l
1	16	32	3	4	64
$2 \sim 9$	16	32	3	64	64
10	16	32	3	64	4

$\mathbb{C}^{N_R \times N_T \times 2}$ is also a three-dimensional matrix composed of $\frac{\mathbf{H}_{k_0'}^i}{c}, \frac{\mathbf{H}_{k_0'+1}^i}{c} \in \mathbb{C}^{N_R \times N_T}$, where $\mathbf{H}_{k_0'}^i$ and $\mathbf{H}_{k_0'+1}^i$ are the corresponding true channel matrices. $c > 0$ is a scaling constant to make the value range of the real and imaginary parts of all the target data, $\underline{\mathbf{H}}_i$, match the activation function applied in the output layer of the SF-CNN. Then $\underline{\mathbf{R}}_i$ is fed into the SF-CNN to approximate the corresponding scaled true channels $\underline{\mathbf{H}}_i$.

For the mmWave massive MIMO systems, we use $N_T = 32$, $N_R = 16$ as a typical example. As shown in Fig. 1, the SF-CNN receives the tentatively estimated complex channel matrices, $\mathbf{R}_{k_0'}^i \in \mathbb{C}^{16 \times 32}$ and $\mathbf{R}_{k_0'+1}^i \in \mathbb{C}^{16 \times 32}$, as the input and separates their real and imaginary parts so that four 16×32 real-valued matrices are obtained. In the subsequent convolutional layer, the four matrices are processed by 64 $3 \times 3 \times 4$ convolutional filters with the rectified linear unit (ReLU) activation function to generate 64 16×32 real-valued matrices. Zero padding (ZP) is used when processing each feature matrix so that its dimension maintains unchanged after convolution. Then a batch normalization (BN) layer is added to avoid the gradient diffusion and overfitting. For the next eight convolutional layers, each uses 64 $3 \times 3 \times 64$ filters to operate ZP convolution with the feature matrices passed by the previous layer and outputs 64 16×32 real-valued feature matrices. ReLU activation function is applied for these eight layers, each of which is followed by a BN layer. The output layer uses four $3 \times 3 \times 64$ convolutional filters to process the 64 16×32 real-valued feature matrices and obtains the estimated real and imaginary parts of the scaled channel matrices at the k_0'th and $(k_0' + 1)$th subcarrier, i.e., $\frac{\text{Re}(\hat{\mathbf{H}}_{k_0'}^i)}{c}$, $\frac{\text{Im}(\hat{\mathbf{H}}_{k_0'}^i)}{c}$, $\frac{\text{Re}(\hat{\mathbf{H}}_{k_0'+1}^i)}{c}$, and $\frac{\text{Im}(\hat{\mathbf{H}}_{k_0'+1}^i)}{c}$. Hyperbolic tangent activation function is used in the output layer to map the output into interval $[-1, 1]$. After scaling up and combining the corresponding real and imaginary parts, the 16×32 complex-valued estimated channel matrices, $\hat{\mathbf{H}}_{k_0'}^i$ and $\hat{\mathbf{H}}_{k_0'+1}^i$, are obtained. Table I lists the detailed architecture of the SF-CNN.

The objective of the offline training for the SF-CNN is to minimize the MSE loss function

$$\text{MSE}_{\text{Loss}} = \frac{1}{N_{\text{tr}} c^2} \sum_{i=1}^{N_{\text{tr}}} \sum_{q=1}^{2} \left\| \mathbf{H}_{k_0'+q-1}^i - \hat{\mathbf{H}}_{k_0'+q-1}^i \right\|_F^2. \quad (11)$$

The design of the SF-CNN architecture draws from CNN based image processing and considers the specific channel estimation task in this paper. The SF-CNN is used for channel denoising and thus we set the size of feature maps of each layer as $N_R \times N_T$. Nine convolutional hidden layers are used to fully uncover the inherent structure of the channel. According to [23], we adopt multiple convolutional filters with a very small size,

i.e., 3×3, to achieve good channel estimation performance with the low complexity. Based on our simulation trials, further increasing the number of convolutional layers or the number of convolutional filters does not bring major performance improvement but causes much higher complexity for CNN training and testing.

3) Online Deployment Issue: After the offline training, the SF-CNN as well as the TE module will be deployed at the receiver to output the estimated channel matrices, $\hat{\mathbf{H}}_{k_0}, \hat{\mathbf{H}}_{k_0+1}, \ldots, \hat{\mathbf{H}}_{k_0+Q-1}$, by jointly processing the pilot matrices, $\mathbf{Y}_{k_0}, \mathbf{Y}_{k_0+1}, \ldots, \mathbf{Y}_{k_0+Q-1}$. If the actual channel model differs from what is used to generate the training set, a straightforward solution is fine-tuning. But it is difficult to collect the true channel and therefore the estimated channel is used instead. It is clear that using more power or longer pilot sequence makes the estimated channel closer to the true channel, which, however, increases the overhead for online fine-tuning. Fortunately, as shown by Fig. 6 and Fig. 7 in Section V, the offline trained SF-CNN is quite robust to most of the new channel statistics that are not observed before. This implies that further online fine-tuning only provides marginal performance improvement and hence might not be necessary.

B. Complexity Analysis

In this subsection, we analyze the computational complexity of the proposed SF-CNN based channel estimation in testing stage and compare it with the non-ideal MMSE using estimated covariance matrix. The required number of floating point operations (FLOPs) is used as the metric.

For the proposed approach, the FLOPs come from the TE module processing in (7) and SF-CNN. By assuming $M_T = N_T$ and $M_R = N_R$, the matrix product in (7) requires FLOPs of $C_{\text{TE}} \sim \mathcal{O}(Q N_T N_R (N_T + N_R))$ [24]. According to [25], the required FLOPs of SF-CNN processing is $C_{\text{SF-CNN}} \sim \mathcal{O}(\sum_{l=1}^{L_c} M_{1,l} M_{2,l} F_l^2 N_{l-1} N_l)$, where L_c is number of convolutional layers, $M_{1,l}$ and $M_{2,l}$ denote the number of rows and columns of each feature map output by the lth layer, F_l is the side length of the filters used by the lth layer, N_{l-1} and N_l denote the numbers of input and output feature maps of the lth layer. Specifically, these parameters are listed in Table II based on the SF-CNN offline training mentioned above. Then the computational complexity of the proposed SF-CNN based channel estimation is given by

$$C_{\text{SF-CNN-CE}}$$

$$\sim \mathcal{O}\left(Q N_T N_R (N_T + N_R) + N_T N_R \sum_{l=1}^{L_c} F_l^2 N_{l-1} N_l \right), \quad (12)$$

For the MMSE channel estimation, least-square (LS) channel estimation needs to be first performed causing FLOPs of $C_{LS} \sim \mathcal{O}(QN_T^2 N_R^2)$. The channel covariance matrix is then calculated based on the LS channel estimation once per channel realization, which requires the computational complexity of $C_{MMSE,1} \sim \mathcal{O}(Q^2 N_T^2 N_R^2)$ if considering both spatial and frequency channel statistics. Finally, the LS channel estimation is refined by the covariance matrix and the corresponding FLOPs is $C_{MMSE,2} \sim \mathcal{O}(Q^3 N_T^3 N_R^3)$. Therefore, the overall computational complexity of MMSE is

$$C_{MMSE} \sim \mathcal{O}(Q^3 N_T^3 N_R^3). \tag{13}$$

It is hard to compare $C_{SF\text{-}CNN\text{-}CE}$ with C_{MMSE} straightforwardly in general since the former depends on L_c, F_l, N_{l-1}, and N_l besides Q, N_T and N_R. If $N_T = 32$, $N_R = 16$, $Q = 2$ and other parameters for the SF-CNN are listed in Table II, the computational complexity of the proposed SF-CNN based approach is in the order of magnitude of 10^8 while MMSE needs a higher complexity in the order of magnitude of 10^9. In addition, the SF-CNN is able to run in a more efficient parallel manner and the runtime of a channel realization is only 1.47×10^{-4} seconds by using NVIDIA GeForce GTX 1080 Ti GPU. By comparison, the MMSE consumes the time of about 6.14×10^{-2} seconds per channel realization on the Intel(R) Core(TM) i7-3770 CPU.

IV. SFT-CNN AND SPR-CNN-BASED CHANNEL ESTIMATION

In this section, we first develop a SFT-CNN based channel estimation approach, which further incorporates channel temporal correlation into the SF-CNN. Then the SFT-CNN is modified to the SPR-CNN to mitigate the huge spatial pilot overhead caused by large-scale antenna arrays.

For time-varying channels, the frequency-domain channel at the kth subcarrier in (4) becomes [7]

$$\mathbf{H}_k(t) = \sqrt{\frac{N_T N_R}{L}} \sum_{l=1}^{L} \alpha_l e^{-j 2\pi (\tau_l f_s \frac{k}{K} - \nu_l t)} \mathbf{a}_R(\varphi_l) \mathbf{a}_T^H(\phi_l), \tag{14}$$

where ν_l denotes the Doppler shift of the lth path.

According to [26], [27], the temporal correlation between channels in successive coherence intervals can be modeled as Gauss-Markov distribution

$$\mathbf{H}_k[n] = \rho \mathbf{H}_k[n-1] + \sqrt{1-\rho^2} \mathbf{\Theta}[n], \quad n \in \mathbb{N}_+ \tag{15}$$

where $\mathbf{H}_k[n] = \mathbf{H}_k(nT)$ is the discrete-time version of $\mathbf{H}_k(t)$ with T denoting the length of the coherence interval, $0 \le \rho \le 1$ denotes the temporal correlation coefficient, and $\mathbf{\Theta}[n]$ is a random matrix accounting for the innovation process with unit variance for each entry. (15) clearly demonstrates that some inherence underlays the channel variation from the previous coherence interval to the current one and this correlation can be also exploited to improve the channel estimation accuracy in addition to the spatial and frequency correlation. In the following, we first elaborate the SFT-CNN based channel estimation.
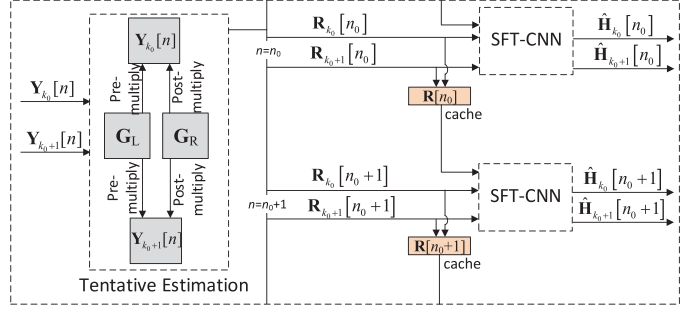

Fig. 2. SFT-CNN based channel estimation.

A. SFT-CNN based Channel Estimation

As shown in Fig. 2, we still consider channel estimation at Q ($= 2$) adjacent subcarriers, k_0 and $k_0 + 1$, for ease of illustration. In time-varying channels, the received pilots after combining at the receiver in (6) becomes

$$\mathbf{Y}_k[n] = \sqrt{P} \mathbf{W}^H \mathbf{H}_k[n] \mathbf{F} + \tilde{\mathbf{N}}_k[n], \quad k \in \{k_0, k_0+1\}. \tag{16}$$

Similar to SF-CNN based channel estimation, $\mathbf{Y}_k[n]$ is then processed by the TE module, which generates the tentatively estimated channel matrices sequentially as

$$\mathbf{R}_k[n] = \sqrt{P} \mathbf{G}_L \mathbf{W}^H \mathbf{H}_k[n] \mathbf{F} \mathbf{G}_R + \mathbf{G}_L \tilde{\mathbf{N}}_k[n] \mathbf{G}_R. \tag{17}$$

Then a SFT-CNN further refines these tentatively estimated channel matrices by exploiting the spatial, frequency, and temporal correlation of channels simultaneously. As shown in Fig. 2, we capture S ($= 2$) successive coherence intervals, n_0 and ($n_0 + 1$), to describe the channel estimation procedure. In the n_0th coherence interval, the tentatively estimated channel matrices, $\mathbf{R}_{k_0}[n_0]$ and $\mathbf{R}_{k_0+1}[n_0]$, are input into the SFT-CNN. A copy of $\mathbf{R}_{k_0}[n_0]$ and $\mathbf{R}_{k_0+1}[n_0]$ is stored in the cache in order to be used in the next coherence interval. In ($n_0 + 1$)th coherence interval, the SFT-CNN receives tentatively estimated channel matrices, $\mathbf{R}_{k_0}[n_0 + 1]$ and $\mathbf{R}_{k_0+1}[n_0 + 1]$, as well as fetches $\mathbf{R}_{k_0}[n_0]$ and $\mathbf{R}_{k_0+1}[n_0]$ from the cache to perform joint processing and obtain the estimated channel matrices as

$$\left\{ \hat{\mathbf{H}}_{k_0}[n_0 + 1], \hat{\mathbf{H}}_{k_0+1}[n_0 + 1] \right\}$$
$$= f_\Psi(\mathbf{R}_{k_0}[n_0], \mathbf{R}_{k_0+1}[n_0], \mathbf{R}_{k_0}[n_0+1], \mathbf{R}_{k_0+1}[n_0+1]; \Psi), \tag{18}$$

where Ψ denotes the parameter set of the SFT-CNN. Meanwhile, the cache is updated by replacing $\mathbf{R}_{k_0}[n_0]$ and $\mathbf{R}_{k_0+1}[n_0]$ with $\mathbf{R}_{k_0}[n_0 + 1]$ and $\mathbf{R}_{k_0+1}[n_0 + 1]$. In each coherence interval, the same SFT-CNN is used since it has learned the general channel temporal correlation instead of the specific relationship between channels in two successive coherence intervals.

After summarizing the channel estimation procedure, we focus on the offline training of the SFT-CNN. Similar to SF-CNN, the training set consisting of N_{tr} samples is generated according to certain channel model in the simulation environment with $(\underline{\mathbf{R}}_i, \underline{\mathbf{H}}_i)$ denoting the ith sample. $\underline{\mathbf{R}}_i \in \mathbb{C}^{N_R \times N_T \times 4}$ is a three-dimensional matrix composed of the tentatively estimated channel matrices in the n_0'th and ($n_0' + 1$)th

TABLE III
ARCHITECTURE OF THE LSTM-CNN

Layer	Type	Number of filters	Size of filters	Activation function	Recurrent activation function
1	Input	-	-	-	-
2∼4	Conv. LSTM	4	3×3	tanh	tanh
5∼11	Conv.	64	3×3	ReLU	-
12	Output	$2Q$	3×3	tanh	-

coherence intervals collected through (17), that is $\underline{\mathbf{R}}_i = [\mathbf{R}^i_{k'_0}[n'_0], \mathbf{R}^i_{k'_0+1}[n'_0], \mathbf{R}^i_{k'_0}[n'_0+1], \mathbf{R}^i_{k'_0+1}[n'_0+1]]$ with $n'_0 \in \mathbb{N}_+$. $\underline{\mathbf{H}}_i \in \mathbb{C}^{N_R \times N_T \times 2}$ is also a three-dimensional matrix composed of the scaled true channel matrices in the $(n'_0 + 1)$th coherence interval, that is $\underline{\mathbf{H}}_i = \left[\frac{\mathbf{H}^i_{k'_0}[n'_0+1]}{c}, \frac{\mathbf{H}^i_{k'_0+1}[n'_0+1]}{c} \right]$. As before, $c > 0$ is the scaling constant to make the value range of the real and imaginary parts of all the target data, $\underline{\mathbf{H}}_i$, match the activation function applied in the output layer of the SFT-CNN. Then $\underline{\mathbf{R}}_i$ is fed into the SFT-CNN to approximate the corresponding scaled true channels $\underline{\mathbf{H}}_i$. The architecture of the SFT-CNN is similar to the SF-CNN except that it has the additional input from the previous coherence interval. With the estimated scaled channel matrices, $\frac{\hat{\mathbf{H}}^i_{k'_0}[n'_0+1]}{c}, \frac{\hat{\mathbf{H}}^i_{k'_0+1}[n'_0+1]}{c}$, the objective of the SFT-CNN offline training is to minimize the MSE loss function

$$\text{MSE}_{\text{Loss}} = \frac{1}{N_{\text{tr}} c^2} \sum_{i=1}^{N_{\text{tr}}} \sum_{q=1}^{2} \big\| \mathbf{H}^i_{k'_0+q-1}[n'_0+1] - \hat{\mathbf{H}}^i_{k'_0+q-1}[n'_0+1] \big\|^2_F. \quad (19)$$

Compared to SF-CNN, SFT-CNN only increases the computational complexity for the first convolutional layer by S times, which is a quite minor part in the total computational complexity according to (12) and Table II. In contrast, if further incorporating temporal correlation, the complexity of MMSE channel estimation in (13) becomes

$$C_{\text{MMSE}} \sim \mathcal{O}(S^3 Q^3 N_T^3 N_R^3), \quad (20)$$

which will be increased significantly even with $S = 2$. Therefore, SFT-CNN provides a simple and efficient way to utilize the channel spatial, frequency, and temporal correlation simultaneously to improve the channel estimation accuracy.

B. SFT-CNN or LSTM-CNN?

The LSTM-CNN would naturally come to mind when we consider temporal correlation. In this subsection, we will compare SFT-CNN and LSTM-CNN in terms of estimation accuracy and complexity, which explains why the SFT-CNN in used instead of LSTM-CNN.

1) Architecture: The LSTM-CNN consists of one input layer, three convolutional LSTM layers, seven convolutional layers, and one output layer, where the processing of each convolutional LSTM layer follows the operation in [28]. The detailed architecture of the LSTM-CNN is listed in Table III.

2) Channel Estimation Accuracy: To measure the channel estimation performance, we use the normalized MSE (NMSE),
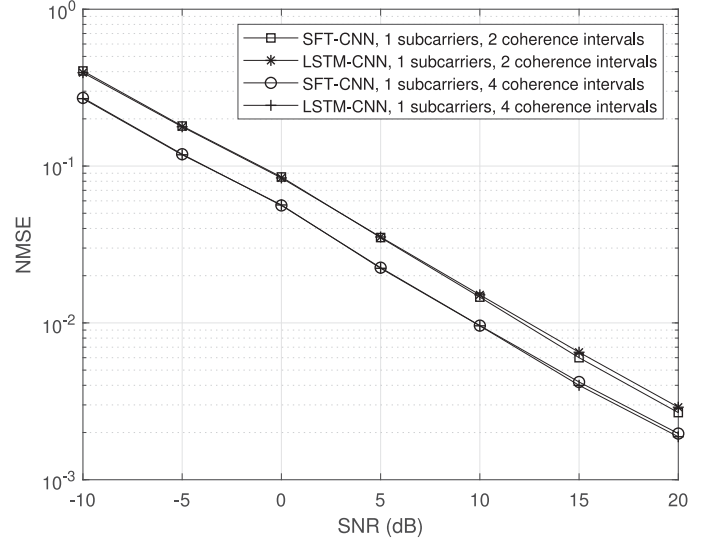

Fig. 3.   NMSE for SFT-CNN and LSTM-CNN based channel estimation.

TABLE IV
TIME COMPLEXITY OF TRAINING AND TESTING

	Training time seconds/epoch		Testing time seconds/channel realization	
	$N_c = 2$	$N_c = 4$	$N_c = 2$	$N_c = 4$
SFT-CNN	22	23	1.51×10^{-4}	1.7×10^{-4}
LSTM-CNN	34	48	3.53×10^{-4}	5.14×10^{-4}

defined as,

$$\text{NMSE} = \mathbb{E}_{\mathbf{H}} \left\{ \|\mathbf{H} - \hat{\mathbf{H}}\|^2_F / \|\mathbf{H}\|^2_F \right\}, \quad (21)$$

where $\mathbf{H}$ and $\hat{\mathbf{H}}$ refer to the true and estimated channels, respectively. Fig. 3 plots the NMSE performance of SFT-CNN and LSTM-CNN based channel estimation, where two and four coherence intervals are involved, respectively. From this figure, SFT-CNN and LSTM-CNN achieve almost the same performance in the whole signal-to-noise ratio (SNR) regime.

3) Training and Testing Complexity: With the same NMSE performance, the complexity becomes an important metric for method selection. Table IV provides the time complexity of training and testing for SFT-CNN and LSTM-CNN on the NVIDIA GeForce GTX 1080 Ti GPU, where N_c denotes the number of coherence intervals involved in the CNNs. It is clear that LSTM-CNN consumes much more time than SFT-CNN in both training and testing stages.

Based on the above discussion, the LSTM-CNN has no performance advantages in exploiting channel temporal correlation but requires much higher complexity for training and testing. Since the SFT-CNN can also capture the channel temporal correlation efficiently with relatively low complexity, this simple pure CNN architecture is preferable for the channel estimation task in this paper.

C. SPR-CNN-Based Channel Estimation

Large-scale array antennas at both the transmitter and the receiver incur huge pilot overhead in spatial domain. In this
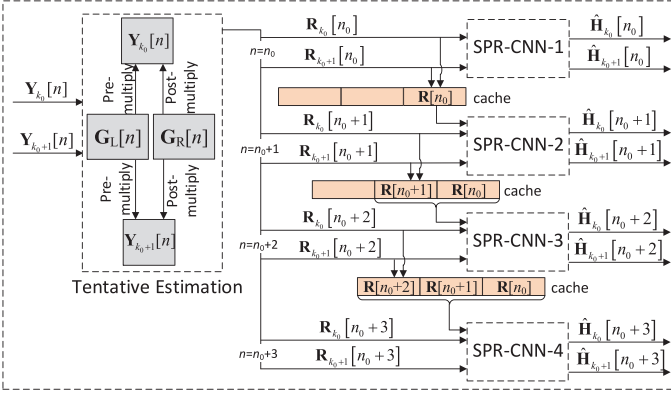
Fig. 4. SPR-CNN-based channel estimation.

subsection, we design the SPR-CNN based channel estimation, which uses much fewer pilots but still guarantees the fairly good accuracy.

The basic idea of the SPR-CNN based channel estimation can be summarized as follows:

1) Group D successive coherence intervals as a CEU, in which channel correlation is utilized to reduce the spatial pilot overhead. Different CEUs are non-overlapped.

2) In the first coherence interval of each CEU, use full spatial pilot overhead for channel estimation.[2] Then the pilot overhead is reduced in the subsequent coherence intervals.

3) For the first coherence interval, the receiver uses the currently received pilots to estimate the current channels. For the rest of coherence intervals, the receiver uses the currently and all previously received pilots in this CEU to jointly estimate the current channels.

Here is the detailed channel estimation procedure. Different beamforming and combining matrices are employed in different coherence intervals of each CEU. As shown in Fig. 4, the received pilots after combining at the receiver in (16) becomes

$$\mathbf{Y}_k[n] = \sqrt{P}\mathbf{W}^H[n]\mathbf{H}_k[n]\mathbf{F}[n] + \tilde{\mathbf{N}}_k[n], \qquad (22)$$

where $\mathbf{F}[n] \in \mathbb{C}^{N_T \times M_T[n]}$ and $\mathbf{W}[n] \in \mathbb{C}^{N_R \times M_R[n]}$ denote the beamforming matrix and combining matrix, respectively, in the nth coherence interval. The corresponding spatial pilot overhead is given by

$$p[n] = M_T[n]\left\lceil\frac{M_R[n]}{N_R^{\mathrm{RF}}}\right\rceil. \qquad (23)$$

From (23), the spatial pilot overhead can be saved by reducing $M_T[n]$ or/and $M_R[n]$.

$\mathbf{Y}_k[n]$ is first processed by the TE module and the tentatively estimated channel matrix is given by

$$\begin{aligned}\mathbf{R}_k[n] &= \mathbf{G}_L[n]\mathbf{Y}_k[n]\mathbf{G}_R[n]\\ &= \sqrt{P}\mathbf{G}_L[n]\mathbf{W}^H[n]\mathbf{H}_k[n]\mathbf{F}[n]\mathbf{G}_R[n]\\ &\quad + \mathbf{G}_L[n]\tilde{\mathbf{N}}_k[n]\mathbf{G}_R[n], \qquad (24)\end{aligned}$$

where $\mathbf{G}_L[n]$ and $\mathbf{G}_R[n]$ are also changed along with the coherence interval index and are expressed as

$$\mathbf{G}_L[n] = \begin{cases}\mathbf{W}[n], & M_R[n] < N_R,\\ (\mathbf{W}[n]\mathbf{W}^H[n])^{-1}\mathbf{W}[n], & M_R[n] \geq N_R,\end{cases} \qquad (25)$$

and

$$\mathbf{G}_R[n] = \begin{cases}\mathbf{F}^H[n], & M_T[n] < N_T,\\ \mathbf{F}^H[n](\mathbf{F}[n]\mathbf{F}^H[n])^{-1}, & M_T[n] \geq N_T.\end{cases} \qquad (26)$$

Then the $\mathbf{R}_k[n]$ is processed by the CNN estimation part. We consider that every $D\,(= 4)$ successive coherence intervals are grouped as a CEU and capture a certain CEU with the n_0th to the $(n_0 + 3)$th coherence intervals, as shown in Fig. 4. There are four SFT-CNNs with different input and output, called SPR-CNN-1, 2, 3, and 4, respectively. In the n_0th coherence interval, full spatial pilot overhead, i.e., $M_T[n] = N_T$ and $M_R[n] = N_R$, is used to provide accurate channel information for all coherence intervals of this CEU. After the TE module, $\mathbf{R}_{k_0}[n_0]$ and $\mathbf{R}_{k_0+1}[n_0]$ are input into the SPR-CNN-1, which generates the finally estimated channel matrices, $\hat{\mathbf{H}}_{k_0}[n_0]$ and $\hat{\mathbf{H}}_{k_0+1}[n_0]$. Meanwhile, a copy of $\mathbf{R}_{k_0}[n_0]$ and $\mathbf{R}_{k_0+1}[n_0]$ is stored in the cache to provide additional channel information for the channel estimation of subsequent coherence intervals. In the $(n_0 + 1)$th coherence interval, the dimensions of $\mathbf{F}[n]$ and $\mathbf{W}[n]$ will be reduced to save the pilot overhead, i.e., $M_T[n] < N_T$ and $M_R[n] < N_R$.[3] $\mathbf{R}_{k_0}[n_0]$ and $\mathbf{R}_{k_0+1}[n_0]$ stored in the cache along with $\mathbf{R}_{k_0}[n_0 + 1]$ and $\mathbf{R}_{k_0+1}[n_0 + 1]$ are simultaneously input into SPR-CNN-2 to obtain $\hat{\mathbf{H}}_{k_0}[n_0 + 1]$ and $\hat{\mathbf{H}}_{k_0+1}[n_0 + 1]$. A copy of $\mathbf{R}_{k_0}[n_0 + 1]$ and $\mathbf{R}_{k_0+1}[n_0 + 1]$ is also stored in the cache in addition to $\mathbf{R}_{k_0}[n_0]$ and $\mathbf{R}_{k_0+1}[n_0]$. All matrices stored in the cache are used for the joint channel estimation of the $(n_0 + 2)$th coherence interval. Channel estimations in the $(n_0 + 2)$th and $(n_0 + 3)$th coherence intervals are similar to that in the $(n_0 + 1)$th coherence interval. After channel estimation in the $(n_0 + 3)$th coherence interval, the cache will be emptied and then used for the next CEU. From Fig. 4, four different SPR-CNNs are employed for respective coherence intervals in a CEU and reused for all CEUs. The architecture and training process of the SPR-CNNs are similar to SFT-CNN except that the numbers of input matrices are different for different SPR-CNNs. An intuitional description of the SPR-CNN based channel estimation is given by Algorithm 1.

Among the four SPR-CNNs, SPR-CNN-4 has the highest complexity with the most input matrices. But it just increases the complexity of the first convolutional layer by $D\,(= 4)$ times compared to the SF-CNN in Section III, which causes limited impact on the total computational complexity. Therefore, SPR-CNN based channel estimation saves the spatial pilot overhead effectively while only increases the complexity moderately.

We design SPR-CNN based channel estimation aiming to reduce the pilot overhead in spatial domain significantly while still guarantee fairly good channel estimation accuracy. In each

---

[2]Full spatial pilot overhead means that the number of beamforming vectors is equal to the number of transmit antennas and the number of combining vectors is equal to the number of receive antennas.

[3]Different pilot overheads may be used in different coherence intervals of a CEU. But for each coherence interval, it and its counterparts in other CEUs should use the same pilot overhead.

Algorithm 1: SPR-CNN based Channel Estimation.

Input: The total number of CEUs M_{CEU}, the number of coherence intervals in each CEU D, spatial pilot overhead from the second to Dth coherence intervals of each CEU

Output: Estimated channel matrices

Procedure:

 1:    Initialize the CEU and coherence interval indices as $m = 1$ and $d = 1$;

 2:    Train D different SPR-CNNs for the first to Dth coherence intervals;

 3:    **for** $m \in [1, M_{\text{CEU}}]$ **do**

 4:      **for** $d \in [1, D]$ **do**

 5:        **if** $d = 1$ **then**

 6:          Use full spatial pilot overhead;

 7:          Tentatively estimate channels according to (24);

 8:          Store the tentatively estimated channel matrices in the cache;

 9:          Input the tentatively estimated channel matrices of the first coherence interval into SPR-CNN-1 to obtain the estimated channel matrices of the first coherence interval;

10:        **else**

11:          Use reduced spatial pilot overhead;

12:          Tentatively estimate channels according to (24);

13:          Store the tentatively estimated channel matrices in the cache (Invalid for $d = D$);

14:          Input the tentatively estimated channel matrices from the first to dth coherence intervals into SPR-CNN-d to obtain the estimated channel matrices of the dth coherence interval;

15:        **end if**

16:      **end for**

17:      Empty the cache and reset $d = 1$

18:    **end for**

19:    **return** the estimated channel matrices of M_{CEU} CEUs

---

CEU with D coherence intervals, full pilot overhead is used in the first coherence interval while reduced pilot overhead is used in the remaining coherence intervals. The first coherence interval provides complete channel information that is quite helpful to channel estimation of all coherence intervals in this CEU. It is clear that the average pilot overhead of a CEU reduces as D increases. However, increasing D will also degrade the average channel estimation accuracy since the effect of the complete channel information provided by the first coherence interval weakens along with the evanescent temporal correlation. Therefore, our architecture of SPR-CNN includes four coherence intervals in a CEU to achieve a good tradeoff between the pilot overhead and the estimation accuracy. In addition, if using the LSTM-CNN to perform the channel estimation in Fig. 4, four LSTM-CNNs with the architecture listed in Table III are needed. As discussed in Section IV-B, the LSTM-CNN has no performance superiority over the simple SFT-CNN architecture but causes much higher complexity and thus is not suitable for the SPR-CNN based channel estimation.

V. NUMERICAL RESULTS

In this section, we present simulation results of the proposed CNN based channel estimation approaches and compare them with non-ideal MMSE using the estimated covariance matrix and ideal MMSE using the true covariance matrix. We set the number of antennas at the transmitter, $N_{\text{T}} = 32$, the number of antennas at the receiver, $N_{\text{R}} = 16$, and the numbers of RF chains at the transmitter and the receiver $N_{\text{T}}^{\text{RF}} = N_{\text{R}}^{\text{RF}} = 2$. $\mathbf{F}$(or $\mathbf{F}[n]$) and $\mathbf{W}$(or $\mathbf{W}[n]$) are set as the first M_{T} (or $M_{\text{T}}[n]$) columns of an $N_{\text{T}} \times N_{\text{T}}$ discrete Fourier transform (DFT) matrix and the first M_{R} (or $M_{\text{R}}[n]$) columns of an $N_{\text{R}} \times N_{\text{R}}$ DFT matrix. In Section V-A and Section V-B, we set $M_{\text{T}} = 32$ and $M_{\text{R}} = 16$. The settings of $M_{\text{T}}[n]$ and $M_{\text{R}}[n]$ will be introduced in Section V-C.

The channel data are generated according to the 3rd Generation Partnership Project (3GPP) TR 38.901 Release 15 channel model [29]. Specifically, we use the clustered delay line models with the carrier frequency, $f_c = 28\,\text{GHz}$, the sampling rate, $f_s = 100\,\text{MHz}$, the number of main paths, $L = 3$, and the number of subcarriers, $K = 64$.

For the SF-CNN, the training set, validation set, and testing set contain 81,000, 9,000, and 19,000 samples, respectively. The parameters of each layer are set as Table I. Adam is used as the optimizer. The epochs are set as 800 while the corresponding learning rates are 10^{-4} for the first 200 epochs, 5×10^{-5} for the next 400 epochs, and 10^{-5} for the last 200 epochs, respectively.[4] The batch size is 128. The scaling constant is set as $c = 2$. The SFT-CNN and SPR-CNN use the same parameters as the SF-CNN except that the numbers of input matrices are different.

A. SF-CNN-Based Channel Estimation

Figure 5 shows the NMSE performance versus SNR of the proposed SF-CNN based channel estimation and MMSE channel estimation over two adjacent subcarriers in the urban micro (UMi) street non-line of sight (NLOS) scenario.[5] The performance of the SF-CNN based approach at single subcarrier is also plotted to demonstrate that frequency correlation is helpful to improve the channel estimation accuracy. Through offline training, the SF-CNN based channel estimation outperforms the non-ideal MMSE with estimated covariance matrix significantly yet requiring lower estimation complexity according to this figure and Section III-B. Moreover, the performance of the SF-CNN based approach is very close to the ideal MMSE with true covariance matrix, especially at the low and medium SNRs.

The robustness of the MMSE and proposed SF-CNN based approaches is shown in Fig. 6. The joint channel estimation over two subcarriers is considered. The SF-CNN is trained in

---

[4]Although optimizer Adam can adaptively adjust the learning rate for each epoch by considering the momentum as well as the gradients, we still manually set the learning rate at the specific epochs to speed up the convergence of offline training.

[5]We train different neural networks separately for different SNRs based on the same channel dataset. Specifically, we first generate a channel dataset according to the 3GPP TR 38.901 Release 15 channel model. Then we use these channel data to generate the received pilot signals in different SNRs, which are used to train the neural networks for corresponding SNRs. The testing set is generated similarly.
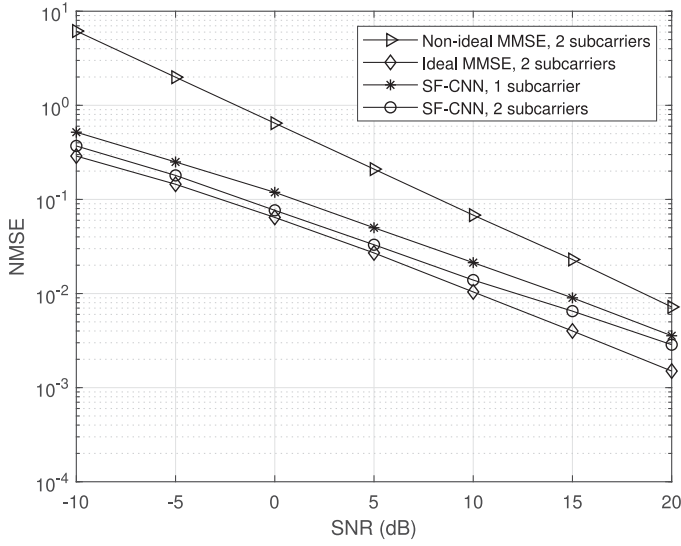
Fig. 5. NMSE versus SNR for the SF-CNN based channel estimation and MMSE channel estimation.

Fig. 7. Robustness of SF-CNN based approach to different numbers of main paths.
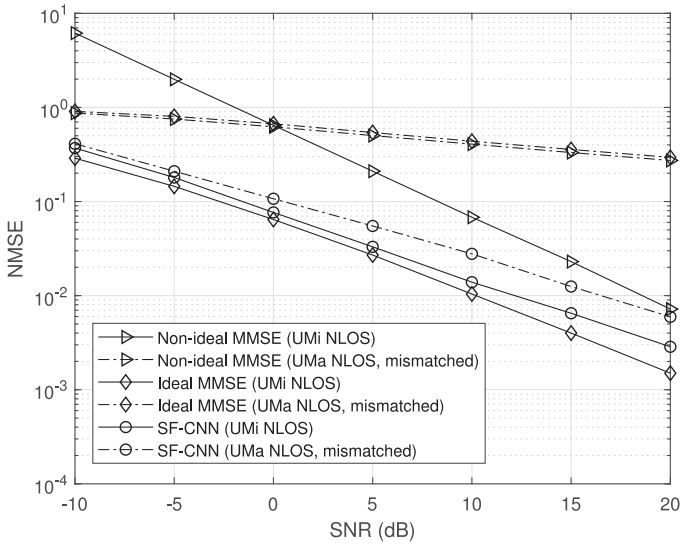

Fig. 6. Robustness of MMSE and SF-CNN based approaches to different scenarios.

the UMi street NLOS scenario and is tested in both UMi street NLOS scenario and urban macro (UMa) NLOS scenario. For the MMSE, its covariance matrix is calculated in the UMi street NLOS scenario and then the channel matrix is estimated in both UMi street NLOS scenario and UMa NLOS scenario. According to [29], the power, delay, and angle profiles of UMa NLOS scenario are quite different from those of UMi street NLOS scenario. The channel statistics are unknown to both SF-CNN and MMSE when they predict the channels in the UMa NLOS scenario. From this figure, the SF-CNN based channel estimation exhibits good robustness when facing the significantly different channel statistics. Even under the mismatched UMa NLOS scenario, the SF-CNN based approach still outperforms the non-ideal MMSE without mismatch. In contrast, due to the high dependence on channel statistics, both the ideal and non-ideal
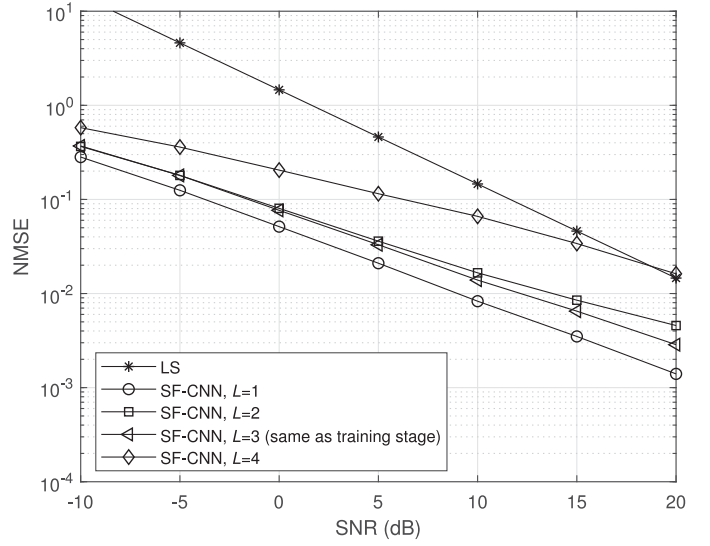
MMSE fail to cope with the change of channel statistics and suffer significant performance loss.

The number of main paths, L, may change in the real scenario. Here, we test the robustness of the SF-CNN based approach to different numbers of main paths, which is plotted in Fig. 7 with $L = 3$ in the training stage and $L = 1, 2, 3, 4$ in the testing stage. The case of $L = 3$ in the testing stage is matched with the training stage and thus acts as the baseline for comparison. LS channel estimation is also plotted as an upper bound. The SF-CNN based approach with $L = 1$ performs better than the baseline, which is because the channel structure becomes quite simple and thus facilitates the CNN denoising. When L is increased to 2, the performance is slightly degraded compared to the baseline. Although there is a significant performance degeneration when L becomes 4, the proposed approach still outperforms LS channel estimation when SNR < 18 dB. In fact, $L = 1$ and 2 can be regarded as two special cases of $L = 3$. The offline trained SF-CNN with $L = 3$ is able to identify the channels with $L = 1$ and 2 and thus exhibits good robustness. In contrast, the case of $L = 4$ leads to a significant change of channel structure, to which the offline trained model is less robust.

B. SFT-CNN-Based Channel Estimation

Figure 8 shows the NMSE performance versus SNR of the SF-CNN, SFT-CNN, and MMSE based channel estimation approaches in the UMi street NLOS scenario. The MMSE and proposed SFT-CNN based approaches utilize the channel information of the previous coherence interval to jointly estimate the current channels over two subcarriers while the SF-CNN based approach does not incorporate temporal correlation. By comparing the circle and square curves, we can clearly see the effect of temporal correlation on improving the NMSE performance with the SFT-CNN based channel estimation. The comparison between the star and square curves demonstrates that just utilizing the additional channel information of adjacent subcarrier
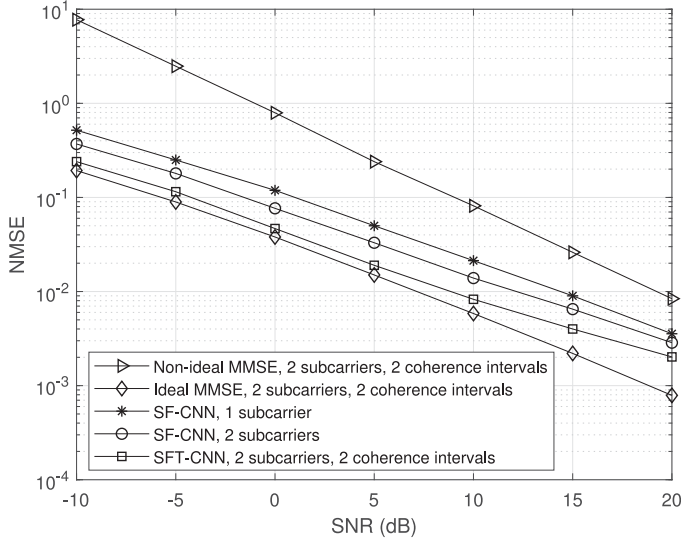
Fig. 8.   NMSE versus SNR for the SFT-CNN based channel estimation and MMSE channel estimation.
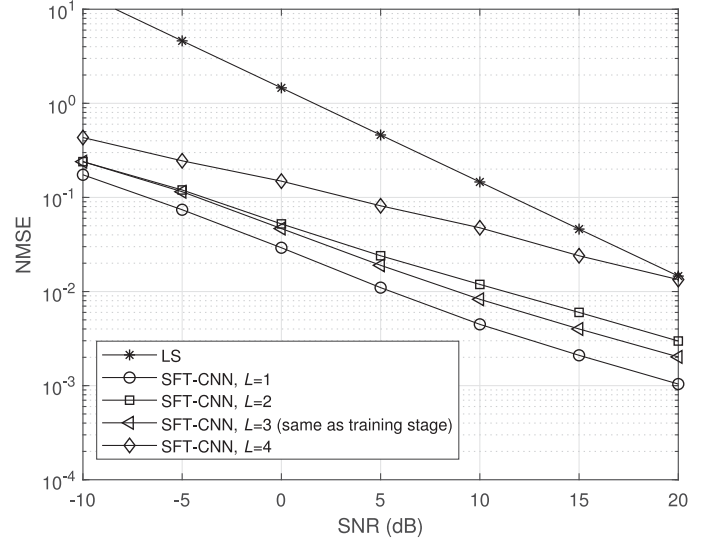

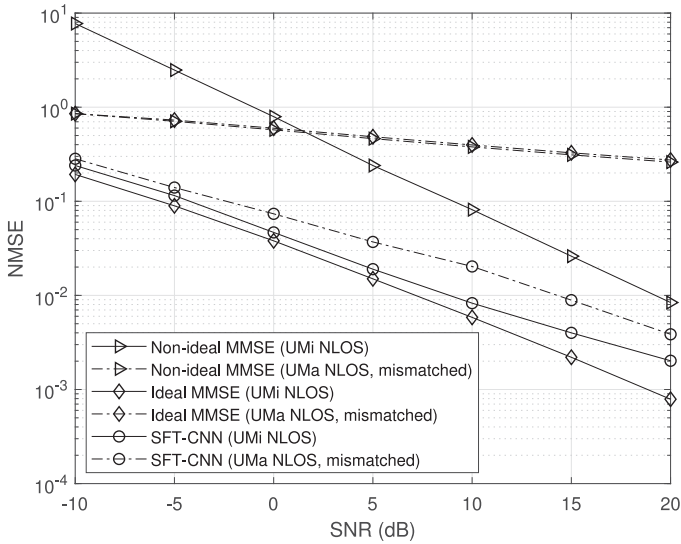Fig. 10.   Robustness of SFT-CNN based approach to different numbers of main paths.

Fig. 9.   Robustness of MMSE and SFT-CNN based approaches to different scenarios.

and coherence interval can improve the estimation accuracy remarkably while only increases the complexity slightly. With the same channel correlation information, the SFT-CNN based approach outperforms the non-ideal MMSE significantly in both performance and complexity at the cost of offline training. It is even very close to the ideal MMSE at the low and medium SNRs.

In Fig. 9, we demonstrate the robustness of the MMSE and proposed SFT-CNN based approaches using the channel correlation of two subcarriers and two coherence intervals. Similar to Fig. 6, both the covariance matrix calculation of the MMSE and the offline training of the SFT-CNN are performed in the UMi street NLOS scenario with the maximum Doppler spread $f_d = 1,400$ Hz. When the testing scenario is unchanged, all these

approaches can achieve good performance. If we test them in a mismatched scenario, i.e., UMa NLOS scenario with $f_d = 1,800$ Hz in Fig. 9, the performance of both ideal and non-ideal MMSE severely degrades since the mismatched covariance matrix is counterproductive when refining the LS estimated channels. In contrast, the SFT-CNN has learned the more inherent channel structure and thus exhibits superior robustness to the significantly different scenarios.

The robustness of SFT-CNN based approach to different numbers of main paths is evaluated in Fig. 10, which exhibits similar results to Fig. 7. Nevertheless, temporal correlation is still helpful to improve the channel estimation accuracy even with mismatched number of main paths by comparing Fig. 7 and Fig. 10.[6]

C. SPR-CNN-Based Channel Estimation

In Fig. 11, we verify the effectiveness of the SPR-CNN based channel estimation in the UMi street NLOS scenario, where every $D \,(= 4)$ successive coherence intervals are grouped as a CEU. In each CEU, we set $M_T[n] = N_T = 32$, $M_R[n] = N_R = 16$, for $d = 1$, and $M_T[n] = 16$, $M_R[n] = 4$, for $d = 2, 3, 4$. So the average spatial pilot overhead of the SPR-CNN based approach is $p_{\text{SPR-CNN}} = \frac{\sum_{d=1}^{4} p[d]}{4} = 88$ and the ratio of it over the full pilot overhead is $r = \frac{p_{\text{SPR-CNN}}}{p_{\text{full}}} = \frac{88}{256} \approx \frac{1}{3}$. For fair comparison, the ideal and non-ideal MMSE based approaches in Fig. 11 also use the above parameter settings. The SF-CNN and SFT-CNN based approaches with full pilot overhead are also plotted for comparison. From the figure, the SPR-CNN based

---

[6]Robustness is an important issue since a robust DNN can avoid the complex online fine-tuning process. There are infinite channel statistics or scenarios in reality and a DNN robust to as many cases as possible is expected to design. This is a challenging task and needs more research attention since the settings of the NN architecture and various hyper-parameters are still unclear for the complicated mmWave MIMO channels.
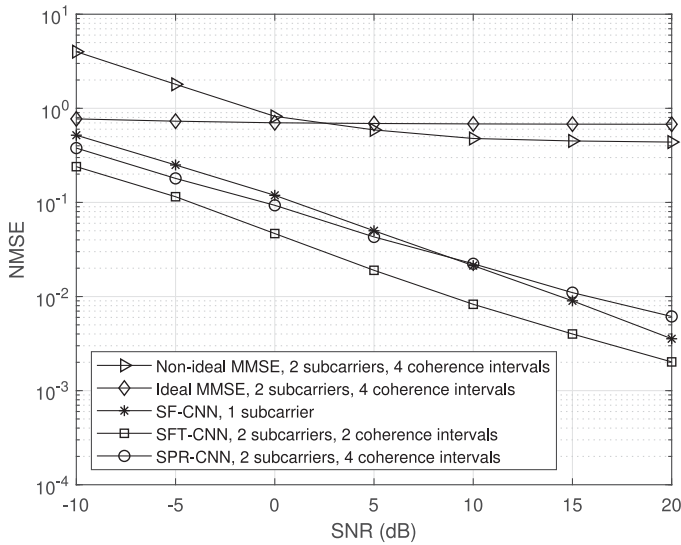
Fig. 11. NMSE versus SNR for the SPR-CNN based channel estimation and MMSE channel estimation.

channel estimation achieves comparable performance to the SF-CNN and SFT-CNN based approaches, especially at the low and medium SNR, but only requires about one third of pilot overhead at the cost of complexity. This means that the additional frequency and temporal correlation has been efficiently utilized by the SPR-CNN based approach to reduce the spatial pilot overhead significantly. On the contrary, both the ideal and non-ideal MMSE perform poorly even if using the same channel correlation information as the SPR-CNN based approach, which reveals that the proposed approach is able to tolerate the reduction of spatial pilot overhead.

VI. CONCLUSION

In this paper, we have developed the deep CNN based channel estimation approaches for mmWave massive MIMO-OFDM systems. The SF-CNN based channel estimation is first proposed to simultaneously utilize spatial and frequency correlation. To further incorporate the temporal correlation in the real scenario, we develop the SFT-CNN based approach. Finally, considering the huge spatial pilot overhead caused by massive antennas, we design the SPR-CNN based channel estimation to mitigate this problem. Numerical results show that the proposed SF-CNN and SFT-CNN based approaches outperform the non-ideal MMSE estimator yet requiring lower complexity and achieve the performance very close to the ideal MMSE estimator. Even if the channel statistics are different, the proposed approaches can still achieve fairly good performance. The SPR-CNN based channel estimation is efficient to save the spatial pilot overhead significantly with minor performance loss.

The proposed CNN based channel estimation approaches can be further improved in terms of the robustness to as many channel statistics as possible. To achieve this aim, the complicated mmWave MIMO channel structure needs to be deeply dissected to provide insights for the design of DNN architecture and hyper-parameters tuning.

REFERENCES

[1] P. Dong, H. Zhang, G. Y. Li, N. NaderiAlizadeh, and I. Gaspar, "Deep CNN for wideband mmWave massive MIMO channel estimation using frequency correlation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Brighton, U.K., May 2019, pp. 4529–4533.

[2] A. L. Swindlehurst, E. Ayanoglu, P. Heydari, and F. Capolino, "Millimeter-wave massive MIMO: The next wireless revolution?" *IEEE Commun. Mag.*, vol. 52, no. 9, pp. 56–62, Sep. 2014.

[3] 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.

[4] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.

[5] B. Wang, F. Gao, S. Jin, H. Lin, and G. Y. Li, "Spatial- and frequency-wideband effects in millimeter-wave massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3393–3406, Jul. 2018.

[6] M. Jian, F. Gao, Z. Tian, S. Jin, and S. Ma, "Angle-domain aided UL/DL channel estimation for wideband mmWave massive MIMO systems with beam squint," *IEEE Trans. Wireless Commun.*, to be published.

[7] R. W. Heath Jr., N. Gonzlez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE J. Sel. Topics in Signal Process.*, vol. 10, no. 3, pp. 436–453, Apr. 2016.

[8] L. Liang, W. Xu, and X. Dong, "Low-complexity hybrid precoding in massive multiuser MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 653–656, Dec. 2014.

[9] A. Alkhateeb, O. E. 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.

[10] Z. Gao, C. Hu, L. Dai, and Z. Wang, "Channel estimation for millimeter-wave massive MIMO with hybrid precoding over frequency-selective fading channels," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1259–1262, Jun. 2016.

[11] Y. Wang, W. Xu, H. Zhang, and X. You, "Wideband mmWave channel estimation for hybrid massive MIMO with low-precision ADCs," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 285–288, Feb. 2019.

[12] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and R. S. Blum, "Low-rank tensor decomposition-aided channel estimation for millimeter wave MIMO-OFDM systems," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 7, pp. 1524–1538, Jul. 2017.

[13] H. Ghauch, T. Kim, M. Bengtsson, and M. Skoglund, "Subspace estimation and decomposition for large millimeter-wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 528–542, Apr. 2016.

[14] Z. Qin, H. Ye, G. Y. Li, and B.-H. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, Apr. 2019.

[15] 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.

[16] 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.

[17] 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.

[18] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 416–419, Apr. 2019.

[19] 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.*, to be published.

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

[21] P. Dong, H. Zhang, and G. Y. Li, "Machine learning prediction based CSI acquisition for FDD massive MIMO downlink," in *Proc. IEEE Global Commun. Conf.*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.

[22] Y.-S. Jeon, S.-N. Hong, and N. Lee, "Blind detection for MIMO systems with low-resolution ADCs using supervised learning," in *Proc. IEEE Int. Conf. Commun.*, Paris, France, May 2017, pp. 1–6.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv: 1409.1556*, 2014.

[24] R. Hunger, "Floating point operations in matrix-vector calculus," Technische Universität München, Associate Institute for Signal Processing, Tech. Rep. v1.3, 2007.

[25] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 5353–5360.

[26] V. Va, J. Choi, and R. W. Heath Jr., "The impact of beamwidth on temporal channel variation in vehicular channels and its implications," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5014–5029, Jun. 2017.

[27] J. Choi, D. Love, and P. Bidigare, "Downlink training techniques for FDD massive MIMO systems: Open-loop and closed-loop training with memory," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 802–814, Oct. 2014.

[28] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[29] 3GPP, "5G; Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), Tech. Rep. 38.901 V15.0.0, Jun. 2018.
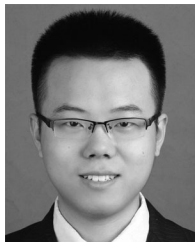
Peihao Dong (S'15) received the B.S. degree in communication engineering from Shandong University, Weihai, China, in 2012 and the M.S. degree in communication and information system from Shandong University, Jinan, China, in 2015. He is currently pursuing the Ph.D. degree at the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China. Since October 2017, he has been a Visiting Student with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His current research interests include massive MIMO, machine learning for wireless communications and relaying transmission.
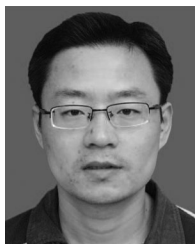
Hua Zhang (M'04) received the B.S. and M.S. degrees from the Department of Radio Engineering, Southeast University, Nanjing, China, in 1998 and 2001, respectively, and the Ph.D. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, in 2004. From 2001 to 2004, he was a Graduate Research Assistant with the Georgia Institute of Technology. From 2004 to 2005, he was a Senior System Engineer with Skyworks Solutions Inc., Irvine, CA, USA. From 2005 to 2006, he was a Staff Engineer with MaxLinear Inc., Carlsbad, CA, USA. Since August 2006, he has been an Associate Professor with the School of Information Science and Engineering, Southeast University, Nanjing, China. Prof. Zhang was the recipient of the best paper awards of IEEE Microwave, Antenna, Propagation, and EMC Technologies in 2013, and IEEE Global Communication Conference in 2014. His current research interests include massive MIMO, software defined radio, and cooperative communications.

Geoffrey Ye Li (S'93–M'95–SM'97–F'06) received the B.S.E. and M.S.E. degrees from the Department of Wireless Engineering, Nanjing Institute of Technology, Nanjing, China, in 1983 and 1986, respectively, and the Ph.D. degree from the Department of Electrical Engineering, Auburn University, Auburn, AL, USA, in 1994.

He was a Teaching Assistant and then a Lecturer with Southeast University, Nanjing, China, from 1986 to 1991, a Research and Teaching Assistant with Auburn University from 1991 to 1994, and a Postdoctoral Research Associate with the University of Maryland, College Park, MD, USA, from 1994 to 1996. He was with AT&T Labs—Research at Red Bank, NJ, USA, as a Senior and then a Principal Technical Staff Member from 1996 to 2000. Since 2000, he has been with the School of Electrical and Computer Engineering, Georgia Institute of Technology as an Associate Professor and then a Full Professor. He has authored and coauthored more than 500 journals and conference papers in addition to over 40 granted patents on the topics of his research interests, which include statistical signal processing and machine learning for wireless communications. His publications have been cited over 35 000 times and he has been recognized as the World's Most Influential Scientific Mind, also known as a Highly-Cited Researcher, by Thomson Reuters almost every year. Prof. Li was the recipient of IEEE Fellow for his contributions to *Signal Processing for Wireless Communications* in 2005. He won the 2010 IEEE ComSoc Stephen O. Rice Prize Paper Award, 2013 IEEE VTS James Evans Avant Garde Award, 2014 IEEE VTS Jack Neubauer Memorial Award, 2017 IEEE ComSoc Award for Advances in Communication, and 2017 IEEE SPS Donald G. Fink Overview Paper Award. He also received 2015 Distinguished Faculty Achievement Award from the School of Electrical and Computer Engineering, Georgia Tech. He has been involved in editorial activities for over 20 technical journals for the IEEE, including Founding Editor-in-Chief of IEEE 5G Tech Focus. He has organized and chaired many international conferences, including Technical Program Vice-Chair of IEEE International Conference on Communications (2003), Technical Program Co-Chair of IEEE Signal Processing Advances in Wireless Communications (2011), General Chair of IEEE Signal and Information Processing (2014), Technical Program Co-Chair of IEEE Vehicular Technology Conference [2016 (Spring)], and General Co-Chair of IEEE Vehicular Technology Conference [2019 (Fall)].

Ivan Simões Gaspar received the Dr.-Ing. degree from Technische Universitt Dresden, Germany, in 2016, and the M.Sc. and B.S.E.E. degrees from Inatel, Brazil, in 2006 and 2003, respectively. From 2003 to 2011, he was a Technical Supervisor and Product Manager with the R&D Department, Hitachi Kokusai Linear. From 2008 to 2011, he collaborated as a Lecturer at Inatel. From 2012 to 2015, he was a Research Associate with the Vodafone Chair, focusing on the 5GNOW project, the NI Lead User Program, and pioneering the research of GFDM. Since 2016, he has been with Intel, Santa Clara, CA, USA, working on future wireless communication systems with great interest on advancing the science and technology of intelligent machines.

Navid NaderiAlizadeh received the B.S. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2011; the M.S. degree in electrical and computer engineering from Cornell University, Ithaca, NY, USA, in 2014; and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2016. He is currently a Research Scientist at Intel Labs, Santa Clara, CA, USA. His research interests include development and analysis of model-based and learning-based radio resource allocation algorithms for 5G and beyond. Dr. Naderializadeh ranked first in the Iranian Nationwide University entrance exam in 2007. He was the recipient of Jacobs Scholarship in 2011. He was selected as a 2015-16 Ming Hsieh Institute Ph.D. Scholar. He has also been a finalist in the Shannon Centennial Student Competition at Nokia Bell Labs in 2016.