IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 25, 2019, accepted February 14, 2019, date of current version April 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2901066

Deep Learning-Based Channel Estimation for Doubly Selective Fading Channels

YUWEN YANG[1], FEIFEI GAO[1], (Senior Member, IEEE), XIAOLI MA[2], (Fellow, IEEE), AND SHUN ZHANG[3], (Member, IEEE)

[1]State Key Lab of Intelligent Technologies and Systems, Beijing National Research Center for Information Science and Technology, Department of Automation, Institute for Artificial Intelligence, Tsinghua University, Beijing 100084, China
[2]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[3]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China

Corresponding author: Feifei Gao (feifeigao@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61831013, Grant 61771274, and Grant 61531011, in part by the Beijing Municipal Natural Science Foundation under Grant 4182030 and Grant L182042, in part by the Guangdong Key Laboratory Project under Grant 2017B030314147, and in part by the Key Laboratory of Universal Wireless Communications (BUPT), Ministry of Education, China.

ABSTRACT In this paper, online deep learning (DL)-based channel estimation algorithm for doubly selective fading channels is proposed by employing the deep neural network (DNN). With properly selected inputs, the DNN can not only exploit the features of channel variation from previous channel estimates but also extract additional features from pilots and received signals. Moreover, the DNN can take the advantages of the least squares estimation to further improve the performance of channel estimation. The DNN is first trained with simulated data in an off-line manner and then it could track the dynamic channel in an online manner. To reduce the performance degradation from random initialization, a pre-training approach is designed to refine the initial parameters of the DNN with several epochs of training. The proposed algorithm benefits from the excellent learning and generalization capability of DL and requires no prior knowledge about the channel statistics. Hence, it is more suitable for communication systems with modeling errors or non-stationary channels, such as high-mobility vehicular systems, underwater acoustic systems, and molecular communication systems. The numerical results show that the proposed DL-based algorithm outperforms the existing estimator in terms of both efficiency and robustness, especially when the channel statistics are time-varying.

INDEX TERMS Deep learning, neural networks, channel estimation, doubly selective channel, LS oriented input, pre-training.

I. INTRODUCTION

The quality of channel estimation is crucial to the performance of wireless communication systems. Classic estimation methods, such as least squares (LS) [1] and minimum mean-square error (MMSE) [2], have been widely used to estimate block fading channels. In mobile scenarios, the transmitted signals often undergo doubly selective fading (i.e., both frequency- and time-selective fading) due to multipath effects and Doppler spread. Various models have been proposed to characterize the doubly selective channels, e.g., basis expansion model (BEM) [3], high order-motion (HOM) model [4], finite state Markov model (FSM) model [5], etc. Among these models, BEM has been widely adopted, in which doubly selective channels are expressed as superpositions of (known) time-invariant basis functions (e.g., Fourier basis functions [3], polynomials [6], wavelets [7], [8], etc) weighted by (unknown) time-varying coefficients [9]. Many BEM-based estimators have been developed for doubly selective fading channels, including LS [10], MMSE [11], recursive LS [12], linear MMSE (LMMSE) [3], etc.

It is not difficult to know that all classical estimators highly rely on tractable mathematically channel models, which are generally assumed to be linear, stationary, and follow Gaussian statistics. However, practical wireless communication systems may have many imperfections and unknown effects that cannot be well captured by accurate models, especially for doubly selective environments. As a result, the existing channel estimators always suffer from performance degradation in real applications. For example, experiment results in [13] and [14] have both demonstrated the error floors of LMMSE estimators.

The associate editor coordinating the review of this manuscript and approving it for publication was Ning Zhang.

Recently, deep learning (DL) has drawn attentions for its great success in computer vision (CV), automatic speech recognition (ASR), and natural language processing (NLP). There are two reasons promoting the applications of DL in various areas [15]. Firstly, DL-based algorithms are data-driven, and therefore are more robust to imperfections in real-world systems. Secondly, DL-based algorithms have low computational complexities, which only involve several layers of simple operations such as matrix-vector multiplications. With the rapid development of massively parallel processing architectures (e.g., graphic processing units (GPUs), specialized chips [16], etc), the execution of deep neural networks (DNNs) can be highly parallelized on concurrent architectures and is easily implemented with low-precision data types [17], which makes DL-based algorithms much more efficient. Motivated by these advantages, DL was introduced to physical layer and achieved superior performance over various issues [18]–[20].

One way to apply DL into physical layer is based on the ‘‘unfold’’ idea, where mutilayer networks are employed to approximate iterative algorithms. By unfolding the iterative algorithm as a chain of iterative operations and mimicking each iteration using a layer, the network with specialized structure can well approximate the iterative algorithm with the aid of off-line training [21]–[23]. For example, a multi-layer neural network for multiple input multiple output (MIMO) detection is designed in [22] by unfolding the orthogonal approximate message passing (OAMP) algorithm, which can achieve better performance than the classical OAMP algorithm with several epochs of training. However, the ‘‘unfold’’ idea is only feasible when the iterations have simple structures. If the iterations face computationally heavy operations, e.g., matrix inversion or singular value decomposition, then the ‘‘unfold’’ type of design is not amenable.

Another way to apply DL into physical layer is to treat the generic deep neural network (DNN) as a ‘‘black box’’, and try to learn the underlying relationships between its inputs and outputs [24]. As rigorously proved in the universal approximation theorem [25], a feed-forward network with a single hidden layer, known as shallow neural network, is capable of approximating any continuous functions defined on compact sets. Compared with the shallow neural network, DNN exhibits more powerful learning capability due to more hidden layers and neurons. Many DNN based approaches have been developed to address issues in wireless communications, such as beamforming [26], [27], channel state information (CSI) feedback [28]–[30], modulation recognition [31], [32], channel encoding and decoding [33]–[36], channel estimation and detection [37], [38]. Especially, the DNN based joint channel estimation and symbol detection algorithm for orthogonal frequency division multiplexing (OFDM) systems with frequency selective channels in [37] is shown to outperform the traditional MMSE estimator when imperfections and non-linearities of systems are taken into account. In [38], a learning assisted channel estimation algorithm was proposed for time-varying channels, and achieves better performance than the LS estimation. To the best of the authors' knowledge, there is no study on DL based channel estimation for doubly selective channels.

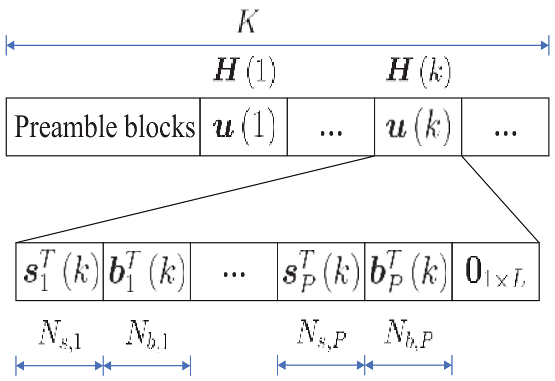

FIGURE 1. Transmitted signal structure.

In this paper, we propose an on-line DL-based estimator using DNN for doubly selective channels. The DNN is first trained with simulated data in an off-line manner and then tracks the dynamic channel in an on-line manner. We also design a pre-training approach for the DNN to acquire a desirable initialization, which can further improve the performance of DL-based estimator. Numerical results show that the DL-based estimator outperforms the conventional BEM-based channel estimator in all scenarios. These results also demonstrate the efficiency and robustness of the proposed DL-based estimator.

The rest of this paper is organized as follows. The system model for doubly selective channels is introduced in Section II. The proposed DL-based estimation algorithm is presented in Section III. Numerical results are presented in Section IV. Our main conclusions are given in Section V.

Notations: The bold and lowercase letters denote vectors while the bold and capital letters denote matrices. The notation $[\boldsymbol{X}]_{n,m}$ denotes the (n, m)th entry of the matrix $\boldsymbol{X}$, where the index m and n both start from 0. The notations $\Re[\cdot]$ and $\Im[\cdot]$, respectively, denote the real and imaginary parts of matrices, vectors or scales. The notations $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the integer floor and ceiling of x, respectively. The notation $|x|$ denotes the absolute value of x. The notation $\|\boldsymbol{x}\|_1$ denotes the L_1 norm of $\boldsymbol{x}$. The notations $(\cdot)^H$ and $(\cdot)^{-1}$ denote the Hermitian and inverse of the matrix, respectively. The notation $E[\cdot]$ represents the expectation with respect to all random variables within the brackets. The notation $\otimes$ represents Kronecker product. The notation $\mathrm{vec}(\cdot)$ represents the vectorization of the matrix.

II. SYSTEM MODEL AND CONVENTIONAL ALGORITHMS

In this section, we present the doubly selective channel model. Then, the conventional BEM based LS and linear MMSE (LMMSE) channel estimators are briefly reviewed as the baselines of channel estimation algorithms.

$$h(t;\tau)=\sum_{l}\sum_{\mu}a_{l,\mu}\exp\left[j\left(\phi_{l,\mu}+2\pi f_{\max}\cos\left(\beta_{l,\mu}t\right)\right)\right]\delta(\tau-\tau_l) \tag{3}$$

A. TRANSMITTED SIGNAL

The transmitted signal structure follows the current IEEE standards, as shown in Fig. 1. Each frame contains K blocks and can be split into two parts. The first part is preamble while the second part is the information blocks. Denote k and $L+1$ as the block index and the number of multipath, respectively. Each information block $\boldsymbol{u}(k)$ has the form $\left[\boldsymbol{u}_{sb}^T(k), \boldsymbol{0}_{1\times L}\right]^T$, where $\boldsymbol{u}_{sb}(k)$ contains N_s information symbols and N_b pilot symbols, while $\boldsymbol{0}$ is the zero padding to avoid the inter block interference (IBI).[1] Moreover, $\boldsymbol{u}_{sb}(k)$ is divided into P sub-blocks, each containing consecutive $N_{s,p}$ information symbols and $N_{b,p}$ pilot symbols, denoted as $\boldsymbol{s}_p(k)$ and $\boldsymbol{b}_p(k)$, respectively. There are $\sum_{p=1}^{P}N_{s,p}=N_s$, $\sum_{p=1}^{P}N_{b,p}=N_b$, and $N_s+N_b+L=N$, where N is the number of symbols in one information block. For simplicity, we set $N_{s,p}=N_s/P$ and $N_{b,p}=N_b/P$ for $1\le p\le P$. Therefore, the kth information block $\boldsymbol{u}(k)$ can be expressed as:

$$\begin{aligned}\boldsymbol{u}(k)&=\left[\boldsymbol{u}_{sb}^T(k),\boldsymbol{0}_{1\times L}\right]^T\\&=\left[\boldsymbol{s}_1^T(k),\boldsymbol{b}_1^T(k),\cdots,\boldsymbol{s}_P^T(k),\boldsymbol{b}_P^T(k),\boldsymbol{0}_{1\times L}\right]^T.\end{aligned} \tag{1}$$

B. CHANNEL MODELS

Typically, the time variation in rich scattering environment is characterized by Jakes model. It should be mentioned that the proposed DL-based estimator does not require any prior knowledge of the channel models, i.e., model free, and is applicable for any communication systems. The Jakes model is briefly introduced in the following.

1) JAKES MODEL

Let $h_J(t)$ denote the channel generated by Jakes' model [39]:

$$h_J(t)=\sum_{\mu}a_\mu\exp\left[j\left(\phi_\mu+2\pi f_{\max}\cos\left(\beta_\mu\right)t\right)\right], \tag{2}$$

where a_μ is the amplitude of the μth propagation path; ϕ_μ and β_μ are, respectively, the angle of arrival and random phase of the μth path; $f_{\max}=f_c v_{\max}/c$ is the maximum Doppler frequency with carrier frequency f_c, maximum mobile velocity $v_{\max}$, and the speed of light c. Both ϕ_μ and β_μ are mutually independent and uniformly distributed over $[-\pi,\pi)$.

2) DOUBLY-SELECTIVE CHANNELS

The time- and frequency-selective channel model can be given as Eq. (3), shown at the top of this page, where l is the index for multipath with $0\le l\le L$, while $a_{l,\mu}$, $\phi_{l,\mu}$, $\beta_{l,\mu}$, and τ_l are the path dependent amplitude, angle of arrival, phase and time delay, respectively. After sampling, the discrete time channel is given by

$$h(n;l)\triangleq h(nT_s;\tau), \tag{4}$$

where T_s is the sampling period.

C. RECEIVED SIGNAL

Denote $u(i)$ as the transmitted symbol at the ith slot, and the received symbol is given by

$$r(i)=\sum_{l=0}^{L}h(i;l)\,u(i-l)+w(i), \tag{5}$$

where $w(i)$ is the additive white Gaussian noise (AWGN), i.e., $w(i)\sim\mathcal{CN}\left(0,\sigma_w^2\right)$ with σ_w^2 denoting the noise variance.

With the zero padding structure in Eq. (1), the received signal can be written in matrix-vector form as follows:

$$\boldsymbol{r}(k)=\boldsymbol{H}(k)\,\boldsymbol{u}_{sb}(k)+\boldsymbol{w}(k), \tag{6}$$

where $\boldsymbol{r}(k)=[r(kN),\cdots,r(kN+N-1)]^T$, $\boldsymbol{w}(k)=[w(kN),\cdots,w(kN+N-1)]^T$, and $\boldsymbol{H}(k)$ is an $N\times(N-L)$ Toepliz matrix with entries

$$[\boldsymbol{H}(k)]_{n,m}=\begin{cases}h(kN+n;n-m), & 0\le n-m\le L;\\ 0, & \text{otherwise.}\end{cases} \tag{7}$$

D. BEM-BASED LS AND LMMSE ESTIMATOR

With the basis expansion model (BEM), the discrete-time baseband equivalent channel $h(i;l)$ can be written as follows (see [40] for detailed derivations):

$$h(i;l)=\sum_{q=0}^{Q}h_{\mathrm{B},q}\left(\lfloor i/N\rfloor;l\right)e^{j\omega_q i},\quad l=0,1,\cdots,L, \tag{8}$$

where $\left\{h_{\mathrm{B},q}\left(\lfloor i/N\rfloor;l\right)\right\}_{q=0}^{Q}$ are the BEM coefficients, $\left\{e^{j\omega_q i}|\omega_q=2\pi\left(q-Q/2\right)/N\right\}_{q=0}^{Q}$ are Fourier bases, and $Q:=2\lceil f_{\max}NT_s\rceil$. For simplicity, Eq. (8) can be written in matrix-vector form as follows:

$$\boldsymbol{h}_l=\boldsymbol{B}\boldsymbol{h}_{\mathrm{B},l}, \tag{9}$$

where $\boldsymbol{B}$ is an $N\times(Q+1)$ matrix with $[\boldsymbol{B}]_{m,n}=e^{jmw_n}$, $\boldsymbol{h}_l=[h(kN;l),\cdots,h(kN+N-1;l)]^T$, and $\boldsymbol{h}_{\mathrm{B},l}=\left[h_{\mathrm{B},0}(k;l),\cdots,h_{\mathrm{B},Q}(k;l)\right]^T$.

Since the following channel estimation is based on a single block, the block index k in Eq. (6) is omitted. We can extract one sub-block of $\boldsymbol{r}$, denoted as $\boldsymbol{r}_b=\left[\boldsymbol{r}_1^b,\cdots,\boldsymbol{r}_P^b\right]^T$, that depends only on $\boldsymbol{H}$ and $\left\{\boldsymbol{b}_p\right\}_{p=1}^{P}$ as shown in Fig. 2. With the BEM in Eq. (9), $\boldsymbol{r}_b$ can be obtained as follows:

$$\boldsymbol{r}_b=\boldsymbol{\phi}_b\boldsymbol{h}_{\mathrm{B}}+\boldsymbol{w}_b, \tag{10}$$

[1] If OFDM modulation is adopted, zero padding can be replaced by cyclic prefix.

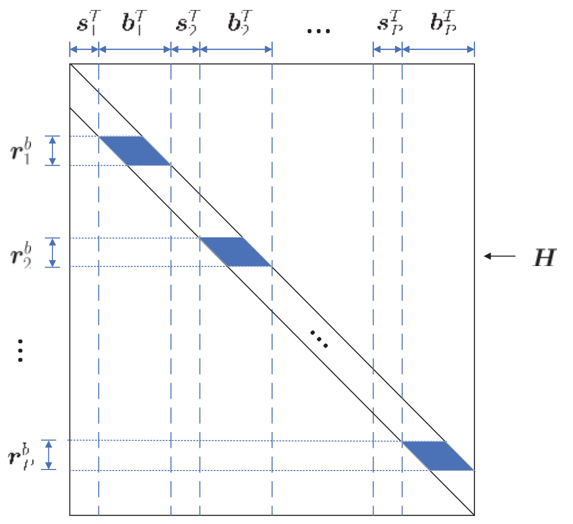

FIGURE 2. Illustration of extracting r_b from r.

where $\boldsymbol{w}_b$ denotes the corresponding noise vector, $\boldsymbol{\phi}_b$ is an $(N_b - PL) \times (Q+1)(L+1)$ matrix consisting of the pilot symbols and Fourier transform sub-matrices determined by pilot positions (see [42, eq. (9)] for specific expression), and $\boldsymbol{h}_{\mathrm{B}} = \left[\boldsymbol{h}_{\mathrm{B},0}^T, \boldsymbol{h}_{\mathrm{B},1}^T, \cdots, \boldsymbol{h}_{\mathrm{B},Q}^T\right]^T$ represents the BEM coefficients to be estimated with[2] $\boldsymbol{h}_{\mathrm{B},q} = \left[h_{\mathrm{B},q}(0), h_{\mathrm{B},q}(1), \cdots, h_{\mathrm{B},q}(L)\right]^T$ for $0 \leq q \leq Q$. Since there are $(Q+1)(L+1)$ unknown coefficients in $\boldsymbol{h}_{\mathrm{B}}$, the minimum number of pilot symbols for estimating doubly selective channels is $L + (Q+1)(L+1)$ to ensure the uniqueness of estimation.

The LS estimator can be written as [42]

$$\hat{\boldsymbol{h}}_{\mathrm{B,LS}} = \left(\boldsymbol{\phi}_b{}^H \boldsymbol{\phi}_b + \eta \boldsymbol{I}\right)^{-1} \boldsymbol{\phi}_b{}^H \boldsymbol{r}_b, \tag{11}$$

where $\boldsymbol{I}$ is an identity matrix with dimension $(Q+1)(L+1)$, and η is set to be a very small positive real number to ensure the matrix $\boldsymbol{\phi}_b{}^H \boldsymbol{\phi}_b + \eta \boldsymbol{I}$ being full rank. Once $\hat{\boldsymbol{h}}_{\mathrm{B,LS}}$ is obtained, one can first substitute Eq. (11) into Eq. (9) and obtain the LS estimated discrete-time baseband equivalent channel $\left\{\hat{\boldsymbol{h}}_{l,\mathrm{LS}}\right\}_{l=0}^{L}$. Then, the LS estimated channel matrix $\hat{\boldsymbol{H}}_{\mathrm{LS}}(k)$ can be obtained by substituting $\left\{\hat{\boldsymbol{h}}_{l,\mathrm{LS}}\right\}_{l=0}^{L}$ into Eq. (7).

The LMMSE estimator is given as follows [43]:

$$\hat{\boldsymbol{h}}_{\mathrm{B,LMMSE}} = \frac{1}{\sigma_w^2}\left(\boldsymbol{R}_{h_{\mathrm{B}}}^{-1} + \frac{1}{\sigma_w^2}\boldsymbol{\phi}_b{}^H \boldsymbol{\phi}_b\right)^{-1} \boldsymbol{\phi}_b{}^H \boldsymbol{r}_b, \tag{12}$$

where $\boldsymbol{R}_{h_{\mathrm{B}}} = E\left[\boldsymbol{h}_{\mathrm{B}} \boldsymbol{h}_{\mathrm{B}}{}^H\right]$ is the covariance matrix of the BEM coefficients. The covariance matrix $\boldsymbol{R}_{h_{\mathrm{B}}}$ can be obtained in the following way: First, denote $\boldsymbol{R}_{h_l}$ as the covariance matrix of $\boldsymbol{h}_l$, which can be easily obtained based on the LS channel estimation in Eq. (11). Then, the covariance matrix

[2]Since block index k is omitted, we use $h_{\mathrm{B},q}(l)$ to denote $h_{\mathrm{B},q}(k;l)$ for simplicity.

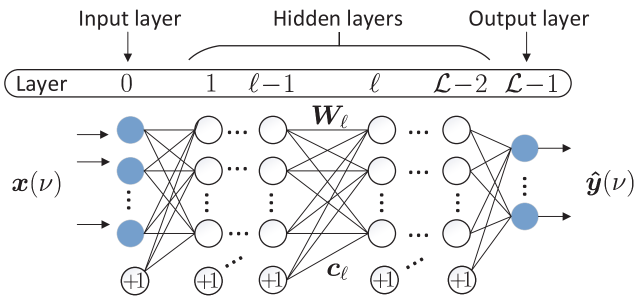

FIGURE 3. The structure of the DNN, where the circles labeled with "+1" are the bias units.

of $\boldsymbol{h}_{\mathrm{B},l}$, can be obtained from Eq. (9) as follows:

$$\boldsymbol{R}_{h_{\mathrm{B},l}} = \left(\boldsymbol{B}^H \boldsymbol{B}\right)^{-1} \boldsymbol{B}^H \boldsymbol{R}_{h_l} \boldsymbol{B} \left(\boldsymbol{B}^H \boldsymbol{B}\right)^{-1}. \tag{13}$$

Finally, $\boldsymbol{R}_{h_{\mathrm{B}}}$ can be obtained as follows:

$$\boldsymbol{R}_{h_{\mathrm{B}}} = \boldsymbol{R}_{h_{\mathrm{B},l}} \otimes \boldsymbol{I}, \tag{14}$$

where $\boldsymbol{I}$ is an identity matrix with dimension $L+1$.

III. DEEP LEARNING BASED CHANNEL ESTIMATION

In this section, we describe the architecture and learning mechanism of the DNN. Then, we present a detailed description of how the pre-training, training and testing stages are performed.

In the following subsections, we first describe the architecture and learning mechanism of the DNN. Then, we introduce the input data structure of the DL-based estimator as well as the generation of data sets required for pre-training, training and testing stages. Finally, we give a detailed description of how to the pre-training, training and testing stages are performed.

A. DEEP LEARNING ALGORITHM

The proposed DL-based channel estimator adopts the fully connected feedforward deep neural network with $\mathcal{L}$ layers, including one input layer, $\mathcal{L}-2$ hidden layers, and one output layer as shown in Fig. 3. The ℓth $(0 < \ell < \mathcal{L}-1)$ layer of the network consists of n_ℓ neurons and one bias unit (circle labeled with "+1"). Each neuron represents a nonlinear transform of a weighted summation of output values of the preceding layer. The nonlinear functions, i.e., the activation functions, used in the DNN can be the Sigmoid function $f_s(x) = 1/(1+\exp(-x))$, or the rectified linear unit (ReLU) function $f_r(x) = \max\{0, x\}$, etc. In the proposed DL-based channel estimator, we choose no activation function for neurons in the output layer and ReLU functions for neurons in the rest of layers.

As shown in Fig. 3, $\boldsymbol{W}_\ell$ is the $n_\ell \times n_{\ell-1}$ weight matrix associated with the $(\ell-1)$th and ℓth layers while $\boldsymbol{c}_\ell$ is the bias vector for the ℓth layer. Since a single execution of the DL algorithm is based on one batch of data, we denote V and ν $(0 \leq \nu \leq V-1)$ as the batch size and serial index, respectively. Let $\boldsymbol{x}(\nu)$ and $\boldsymbol{y}(\nu)$, respectively, represent the input and labels of the DNN at serial index ν. The output of the DNN is the estimate of $\boldsymbol{y}(\nu)$, which can be mathematically

$$\boldsymbol{g}_{\ell}\left(\boldsymbol{x}(\nu);\boldsymbol{\theta}_{\ell}\right)=\begin{cases}\boldsymbol{x}(\nu), & \ell=1;\\ \boldsymbol{f}_{r}\left(\boldsymbol{W}_{\ell}\left(\boldsymbol{g}_{\ell-1}\left(\boldsymbol{x}(\nu);\boldsymbol{\theta}_{\ell-1}\right)\right)+\boldsymbol{c}_{\ell}\right), & 2\leq\ell<\mathcal{L}-1;\\ \boldsymbol{W}_{\mathcal{L}-1}\left(\boldsymbol{g}_{\mathcal{L}-2}\left(\boldsymbol{x}(\nu);\boldsymbol{\theta}_{\mathcal{L}-2}\right)\right)+\boldsymbol{c}_{\mathcal{L}-1}, & \ell=\mathcal{L}-1\end{cases} \tag{16}$$

expressed as

$$\hat{\boldsymbol{y}}(\nu)=\boldsymbol{g}_{\mathcal{L}-1}\left(\cdots\boldsymbol{g}_{1}\left(\boldsymbol{x}(\nu);\boldsymbol{\theta}_{1}\right);\boldsymbol{\theta}_{\mathcal{L}-1}\right), \tag{15}$$

where $\boldsymbol{\theta}_{\ell}\triangleq\{\boldsymbol{W}_{\ell},\boldsymbol{c}_{\ell}\}$ represents the parameters of the ℓth layer. Moreover, $\boldsymbol{g}_{\ell}\left(\boldsymbol{x}(\nu);\boldsymbol{\theta}_{\ell}\right)$ is the output of the ℓth layer, which can be written as Eq. (16), shown at the top of this page, with $\boldsymbol{f}_r$ denoting the vector-form of the ReLU function.[3]

For expression simplicity, we define $\boldsymbol{\theta}\triangleq\{\boldsymbol{\theta}_{\ell}\}_{\ell=1}^{\mathcal{L}-1}$ as the set of parameters to be optimized. The optimal $\boldsymbol{\theta}$ can be obtained by minimizing the loss function Loss$(\boldsymbol{\theta})$ through off-line training, and Loss$(\boldsymbol{\theta})$ can be written as:

$$\text{Loss}(\boldsymbol{\theta})=\frac{1}{VL_y}\sum_{\nu=0}^{V-1}\left\|\hat{\boldsymbol{y}}(\nu)-\boldsymbol{y}(\nu)\right\|_1, \tag{17}$$

where L_y is the length of the vector $\boldsymbol{y}(\nu)$. In addition, we adopt the L_1 norm (absolute error) instead of L_2 norm (squared error) as loss function because L_1 norm removes the outliers and thus makes DNN converges faster.

Various optimization algorithms can be used to minimize *Loss*$(\boldsymbol{\theta})$ by iteratively updating the parameters $\boldsymbol{\theta}$, i.e., stochastic gradient descent [44], root mean square prop [45], adaptive moment estimation (ADAM) [46], etc. We adopt ADAM as the optimization algorithm for the proposed DNN, which is straightforward to implement, invariant to diagonal rescaling of the gradients, and computationally efficient.

B. IMPLEMENTATION OF THE DL-BASED CHANNEL ESTIMATION ALGORITHM

As illustrated in Fig. 4, the proposed DL-based estimation algorithm has three stages, i.e., the pre-training, training, and testing stages. The DNN is trained off-line in both the pre-training and training stages. While in the testing stage, the channels can be dynamically tracked by the DNN with only pilots known, and then the transmitted symbols are detected.

1) PRE-TRAINING STAGE

Let us collect the system data in one block (i.e., the transmitted signals, received signals, etc) as the training data at one serial index. Denote D as the number of transmitted frames in one training batch. As shown in Fig. 1, each signal frame contains K blocks. Hence, the serial length in one batch is $V=DK$ and the serial index ν can be described as $\nu=dK+k$, where $d=0,1,\cdots,D-1$ and $k=0,1,\cdots,K-1$. Since the implementation of the proposed estimator is based on one transmitted frame, we can omit the index d and use the block index k instead of ν to represent the serial index for simpler illustration.

As shown in Fig. 4, the raw input data of DNN at pre-training stage can be expressed as follows:

$$\boldsymbol{x}_{\text{rp}}(k)=\left[\boldsymbol{u}_{sb}(k)^T,\boldsymbol{r}(k)^T,\right.$$
$$\left.\text{vec}\left(\hat{\boldsymbol{H}}_{\text{LS}}^{sb}(k)\right)^T,\text{vec}\left(\hat{\boldsymbol{H}}(k-1)\right)^T\right]^T, \tag{18}$$

where $\hat{\boldsymbol{H}}_{\text{LS}}^{sb}(k)$ represents the LS estimate of $\boldsymbol{H}(k)$ with $\boldsymbol{u}_{sb}(k)$ known, and $\hat{\boldsymbol{H}}(k-1)$ is the previous estimated channel. Since DL-based algorithms can only work in real domain, the raw input data should be reshaped. Define $\boldsymbol{f}_{\text{R}}(\boldsymbol{z})$ as the input reshaping function, i.e.,

$$\boldsymbol{f}_{\text{R}}(\boldsymbol{z})=\left[\Re\{\boldsymbol{z}\}^T,\Im\{\boldsymbol{z}\}^T\right]^T. \tag{19}$$

Then, the real input data of the DNN in the pre-training stage are given as:

$$\boldsymbol{x}_{\text{p}}(k)=\boldsymbol{f}_{\text{R}}(\boldsymbol{x}_{\text{rp}}(k)). \tag{20}$$

In addition, the relationship between the output of DNNs $\hat{\boldsymbol{y}}(k)$ and the DL-based estimated channels $\hat{\boldsymbol{H}}(k)$ can be written as follows:

$$\hat{\boldsymbol{H}}(k)=\text{vec}^{-1}\left(\boldsymbol{f}_{\text{R}}^{-1}(\hat{\boldsymbol{y}}(k))\right), \tag{21}$$

where $\boldsymbol{f}_{\text{R}}^{-1}(\cdot)$ and $\text{vec}^{-1}(\cdot)$ are the inverse function of $\boldsymbol{f}_{\text{R}}(\cdot)$ and $\text{vec}(\cdot)$, respectively.

In the pre-training stage, we first initialize the weights of the DNN as random variables that follow the truncated normal distribution[4] with normalized variance.[5] Such kind of initialization of weights is recommended for the DNN with ReLu used as activation functions [47], [48]. The biases of the DNN are initialized to be small constants closed to 0. Let us denote the initial parameters of the DNN as $\boldsymbol{\theta}^{(i)}$. Then, we minimize *Loss*$(\boldsymbol{\theta})$ by the ADAM algorithm [46] until the DNN converges and then denote the converged parameters of the DNN as $\boldsymbol{\theta}^{(p)}$.

One major motivation for designing the pre-training approach is to obtain a desirable initialization for the training stage. Generally, when the DNN is randomly initialized, optimization algorithms such as ADAM may be stuck in

[3] Given $\boldsymbol{x}=[x_1,\cdots,x_n]^T$, there is $\boldsymbol{f}_r(\boldsymbol{x})=[f_r(x_1),\cdots,f_r(x_n)]^T$.

[4] The truncated normal distribution is similar to normal distribution except that values more than two standard deviations from the mean are discarded and re-drawn.

[5] We normalize the variance of weights in the following manner: First, generating the weights of neurons in the ℓth $(2\leq\ell\leq\mathcal{L}-1)$ layer as truncated normal variables with variance 2. Then, we divide the weights of neurons by the number of neurons in the $(\ell-1)$th layer.

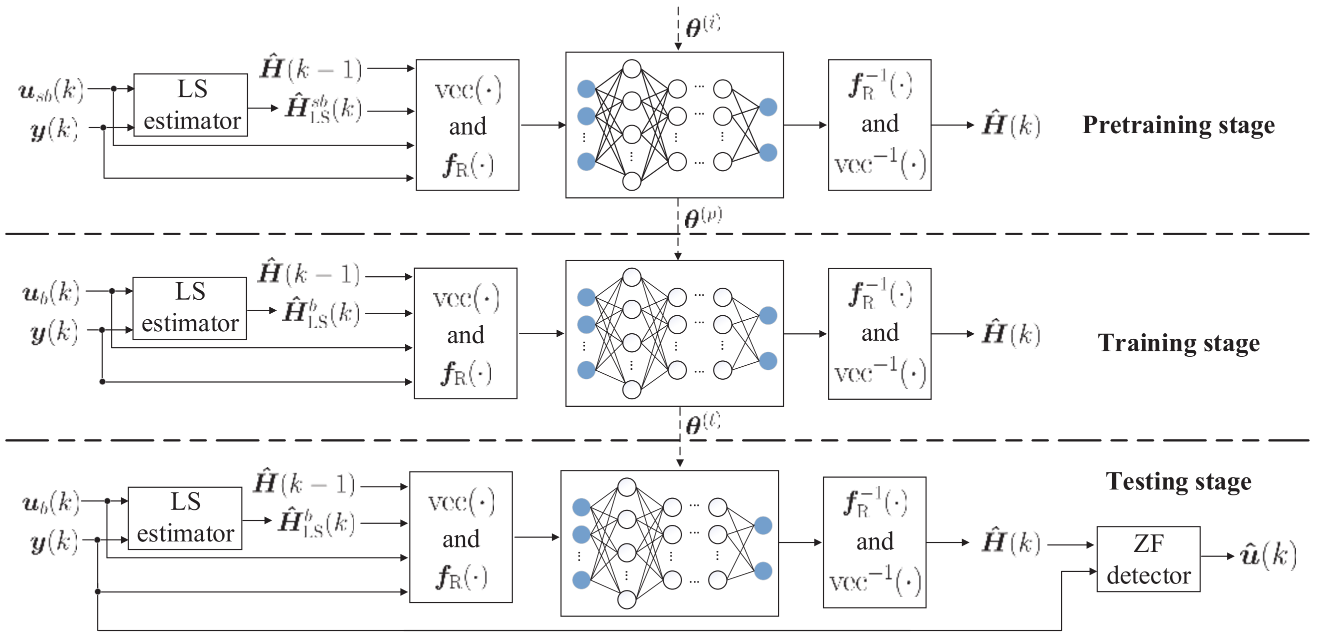
$\boldsymbol{\theta}^{(i)}$
$\boldsymbol{u}_{sb}(k)$
$\boldsymbol{y}(k)$
LS estimator
$\hat{\boldsymbol{H}}(k-1)$
$\hat{\boldsymbol{H}}_{\mathrm{LS}}^{sb}(k)$
vec(·) and $f_{\mathrm{R}}(\cdot)$
$f_{\mathrm{R}}^{-1}(\cdot)$ and $\mathrm{vec}^{-1}(\cdot)$
$\hat{\boldsymbol{H}}(k)$
Pretraining stage
$\boldsymbol{\theta}^{(p)}$
$\boldsymbol{u}_{b}(k)$
$\hat{\boldsymbol{H}}_{\mathrm{LS}}^{b}(k)$
Training stage
$\boldsymbol{\theta}^{(t)}$
Testing stage
ZF detector
$\hat{\boldsymbol{u}}(k)$

FIGURE 4. **Block diagram of the proposed DL-based estimation algorithm, where** vec(·) **and** $f_{\mathrm{R}}(\cdot)$ **represent the vectorization and reshaping function given in Eqs.** (18)-(23) **and Eq.** (19)**, respectively.** $\mathrm{vec}^{-1}(\cdot)$ **and** $f_{\mathrm{R}}^{-1}(\cdot)$ **represent the inverse function of** vec(·) **and** $f_{\mathrm{R}}(\cdot)$**, respectively.**

a low-performance local minimum, resulting in the limited performance of DNNs [49]. One possible way to circumvent this problem is to render the DNN well-initialized, i.e., selecting the initial parameters of DNNs close to the optimal solution [50]. Such way is feasible in channel estimation since one can first transmit training frames, where the pilot and information symbols are both known, to obtain a much more accurate estimation of the channels. Then, the DNN can use these accurate estimates to achieve a more desirable initialization. Moreover, the pre-training approach can help alleviate vanishing gradient problems, and thus makes the DNN converge faster [51]. The improvements brought by pre-training will be evaluated in Section IV-B2.

2) TRAINING STAGE

Define $\boldsymbol{u}_b(k)$ as the sub-block consisting of all pilot symbols in the kth block, which can be written as

$$\boldsymbol{u}_b(k) = \left[\mathbf{0}_{1\times N_{s,1}}, \boldsymbol{b}_1^T(k), \cdots, \mathbf{0}_{1\times N_{s,P}}, \boldsymbol{b}_P^T(k)\right]^T. \quad (22)$$

Then, the raw input data vector of the DNN in the training stage is given as:

$$\boldsymbol{x}_{\mathrm{rt}}(k) = \Big[\boldsymbol{u}_b(k)^T, \boldsymbol{r}(k)^T, \mathrm{vec}\left(\hat{\boldsymbol{H}}_{\mathrm{LS}}^{b}(k)\right)^T, \mathrm{vec}\left(\hat{\boldsymbol{H}}(k-1)\right)^T\Big]^T, \quad (23)$$

where $\hat{\boldsymbol{H}}_{\mathrm{LS}}^{b}(k)$ is the LS estimate of $\boldsymbol{H}(k)$ with $\boldsymbol{u}_b(k)$ known. The real input data of the DNN in the training stage can be written as follows:

$$\boldsymbol{x}_{\mathrm{t}}(k) = \boldsymbol{f}_{\mathrm{R}}(\boldsymbol{x}_{\mathrm{rt}}(k)). \quad (24)$$

The difference between the pre-training and training stages is that information symbols and pilots are both known and added as input data of the DNN in the pre-training stage while only pilots are known in the training stage, which is more practical in real-world systems. The existing work [38] simply used the transmitted pilots $\boldsymbol{u}_b(k)$, received signal $\boldsymbol{y}(k)$, and pervious estimated channel $\hat{\boldsymbol{H}}(k-1)$ as input, which cannot handle the doubly selective channels as shown in Fig. 5. The proposed DL-based estimator originally adopts the LS estimation $\hat{\boldsymbol{H}}_{\mathrm{LS}}^{b}(k)$ as part of the input such that the DNN can take the advantages of the LS estimation to further improve the performance. The effectiveness of the designed input data structure will be validated in Section IV-B3.

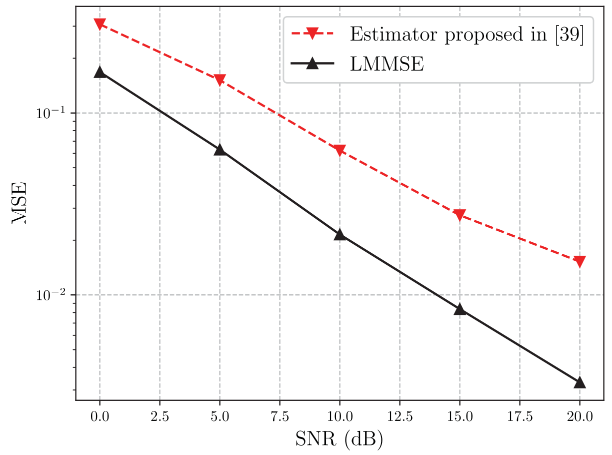

FIGURE 5. **Performance comparison between the LMMSE estimator and estimator proposed in [38].**

We collect the transmitted pilot symbols $\boldsymbol{u}_b(k)$, received signals $\boldsymbol{y}(k)$ and LS estimation $\hat{\boldsymbol{H}}_{\mathrm{LS}}^{b}(k)$ as the input data such that the DNN can not only take the advantages of the LS estimation but also extract more features about the channel, from $\boldsymbol{u}_b(k)$ and $\boldsymbol{y}(k)$. We add $\hat{\boldsymbol{H}}(k-1)$ as part of inputs to capture the underlying features of channel variation. At each block index, the DNN implicitly learns time correlation of the time-varying channels from the previous estimated channel and then merges them with the currently obtained data to further improve the accuracy of channel estimation.

In the training stage, we first load $\boldsymbol{\theta}^{(p)}$ as initial parameters of the DNN and then minimize $Loss\left(\boldsymbol{\theta}\right)$ by the ADAM algorithm until the DNN converges. The parameters of DNN after the training stage are denoted by $\boldsymbol{\theta}^{(t)}$.

3) TESTING STAGE

The input of the DNN in the testing stage has the same structure with that in the training stage. We load the trained parameters $\boldsymbol{\theta}^{(t)}$, and then pass the input data through the trained DNN and obtain the estimated channel. Based on the DL-based estimation channel $\hat{\boldsymbol{H}}(k)$, we use the zero-forcing (ZF) detector to obtain the estimation of $\boldsymbol{u}(k)$, which is denoted by $\hat{\boldsymbol{u}}(k)$ as shown in Fig. 4. In the testing stage, we generate the testing sets with channels following the same statistics as the training stage to evaluate the bit error rate (BER) performance of the DL-based estimator. We also generate the testing sets with channels following different statistics from the training stage to test the robustness of the DL-based estimation algorithm.

C. COMPLEXITY ANALYSIS

The required number of floating point operations (FLOPs) is used as the metric of complexity. For the proposed DL-based estimator, the FLOPs come from the LS estimation processing in Eq. (11) and the DNN processing in Eq. (15). The total number of FLOPs in the LS estimation is $N_Q^2\left(N_b - PL\right) + 2N_Q^3 + N_Q^2$ with $N_Q = (Q+1)(L+1)$ while in the DNN processing it is $\sum_{\ell=1}^{\mathcal{L}-1} n_{\ell-1}n_\ell$. Therefore, The complexity of DL-based estimator can be written as follows:

$$C_{\text{DL-based}} \sim O\left(N_Q^3 + \sum_{\ell=1}^{\mathcal{L}-1} n_{\ell-1}n_\ell\right). \tag{25}$$

IV. SIMULATION RESULTS

In this section, we investigate the performance of the proposed DL-based estimator for doubly selective channels. We first present the configuration and default parameters of the simulation system. Then, hyper-parameter selections and the structure of the input data for the DL-estimator are both evaluated. Finally, the performances of DL-based and BEM-LMMSE estimators are compared and analyzed.

A. SIMULATION SETUP

The proposed DL-based estimator is implemented on one computer with one Nvidia GeForce GTX 1080 Ti Graphical Processing Units (GPU) and 32 GB of memory. Keras 2.2.0 with TensorFlow 1.4.0. as backhaul is employed as the deep learning framework.

Unless otherwise specified, the parameters of the channel model follow the default setting as shown in Tab. 1. The minimum value of Doppler-spread Q is $Q_{\min} = 2\left\lceil f_{\max}NT_s\right\rceil$.

TABLE 1. Default channel parameters.

Parameter	Value
Carrier frequency f_c	5.2 GHZ
Modulation Mode	QPSK
Number of blocks K	11
Number of preamble blocks	2
Number of symbols in one block N	32
Maximum mobile velocity $v_{\max}$	15 m/s
Sampling period T_s	4μs

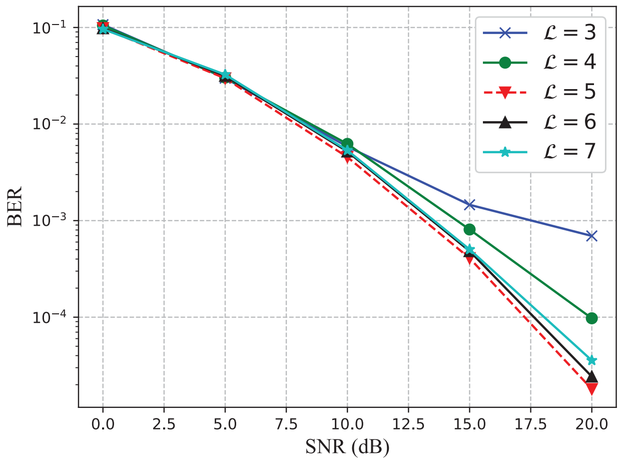

FIGURE 6. Performance of the DL-based algorithms with different numbers of layers.

B. HYPER-PARAMETER SELECTION AND STRUCTURE DESIGNING FOR THE DL-BASED ESTIMATOR

1) HYPER-PARAMETER SELECTION

The default parameters of the DL-based estimator are given in Tab. 2. The numbers of neurons of the input and output layers are consistent with the lengths of input and output data vectors, respectively. It has been widely identified that the hyper-parameter selections are crucial to the performance of DL [52]. The hyper-parameters for the ADAM algorithm, including the learning rate α, exponential decay rates (β_1, β_2), and disturbance term ϵ, follow those in [46]. Generally, these hyper-parameters for the ADAM algorithm have intuitive interpretations and typically require little tuning.

In Fig. 6, we investigate the performance of the DL-based estimators with different number of layers. The performance of the DL-based estimator first improves and then degrades as the number of layers $\mathcal{L}$ increases. From Fig. 6, we observe that the optimal number of layers is 5, which is also the default value of $\mathcal{L}$ for the DL-based estimator in our simulations. Theoretically, the learning capability of the DNN improves as the number of layers increases. In fact, due to the vanishing gradient and pathology degradation, the training of the DNN becomes more challenging as the network goes deeper [53]. Furthermore, when signal-to-noise ratio (SNR) is low, i.e., less than 10 dB, there are no significant gaps between the DL-based estimators with different number of layers. In this case, we can further reduce the number of layers of the DNN for lower complexity. This also indicates

TABLE 2. Default DNN parameters.

Parameter	Value
Number of neurons in hidden layers	512, 256, 128
Learning rate α	0.001
Exponential decay rates (β_1, β_2)	(0.9, 0.999)
Disturbance term ϵ	1e-08
Batch size	128
Activation function	None for the output layer and ReLu for the rest
Loss function	Mean absolute error

that the optimal architectures for the DNN should be chosen according to the practical system configurations and channel conditions.

2) PERFORMANCE EVALUATION FOR THE PRE-TRAINING APPROACH

As mentioned in Section III-B1, we design the pre-training approach to improve the performance of the DL-based estimator. For the DL-based estimator without pre-training approach, the DNN is loaded with the initial parameters $\boldsymbol{\theta}^{(i)}$ instead of $\boldsymbol{\theta}^{(p)}$ at the beginning of training stage.

Fig. 7 compares the performance of the DL-based estimators with and without pre-training, where $N_b = 12$ and $L = 3$. It can be observed that the improvement in BER performance brought by pre-training becomes significant as SNR increases. While the improvement is negligible when SNR is less than 10 dB. This is because that in lower SNR region, additional training frames in the pre-training stage are less effective to provide a well guided initialization for the DL-based estimator. The results show that when SNR is low, the pre-training cannot enhance the performance of DL-based estimator, and therefore is unnecessary. However, when SNR is high, the pre-training can assist the DL-based estimator in the training stage to obtain better initial parameters, and therefore can further improve the performance of DL-based estimator.

3) PERFORMANCE EVALUATION FOR THE INPUT DATA STRUCTURE

In this simulation, the impacts of different input data structures on the BER performance are analyzed. We consider three simplified structure designs corresponding to the DL-based estimator without $\hat{\boldsymbol{H}}(k-1)$, $\{\boldsymbol{u}_b(k), \boldsymbol{r}(k)\}$, and $\hat{\boldsymbol{H}}^b_{\text{LS}}(k)$, respectively.

As shown in Fig. 8, the gap between the DL-based estimator with and without $\hat{\boldsymbol{H}}(k-1)$ increases as SNR increases and the performance loss reaches 7dB when SNR is 20 dB. This indicates the DL-based estimator can learn to capture the underlying features of channel variation to further improve the performance of channel tracking with the aid of $\hat{\boldsymbol{H}}(k-1)$. Furthermore, the performance of the DL-based estimators without $\{\boldsymbol{u}_b(k), \boldsymbol{r}(k)\}$ and without $\hat{\boldsymbol{H}}_{\text{LS}}(k)$ both exhibit severe degradation. The performance losses of the DL-based estimators without $\{\boldsymbol{u}_b(k), \boldsymbol{r}(k)\}$ and without $\hat{\boldsymbol{H}}_{\text{LS}}(k)$ are 5 dB and 7 dB, respectively, when SNR is 20 dB. This implies the DL-based estimator can not only take the advantages of the channel features obtained by the LS estimator but also extract additional channel features from $\boldsymbol{u}_b(k)$ and $\boldsymbol{y}(k)$ to further enhance the estimation performance. The above discussions validate the justifiability of the proposed DL-based estimator.

C. PERFORMANCE COMPARISONS WITH LMMSE ESTIMATOR

1) PERFORMANCE OF CHANNEL TRACKING

The proposed DL-based estimator can be applied to general doubly selective channels. Here, we consider a special case, i.e., the time selective channel, where the number of paths is equal to 1. Fig. 9 plots the tracking of channel using DL-based and LMMSE algorithms, where SNR is set to be 20 dB and $N_b = 12$. It can be observed that the DL-based estimator exhibits more accurate tracking for the amplitude of channels

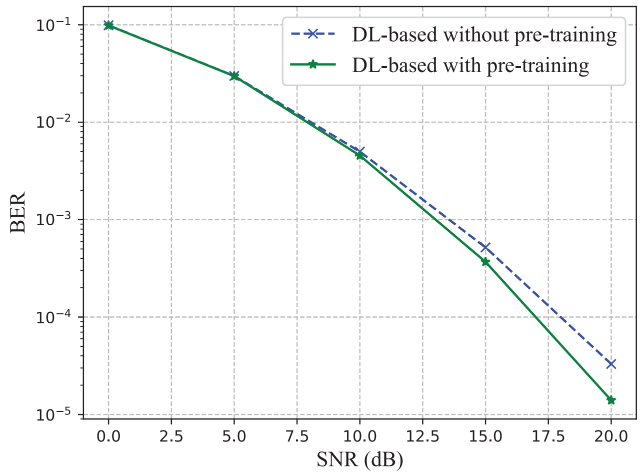

FIGURE 7. Performance comparison between the DL-based estimators with and without pre-training.

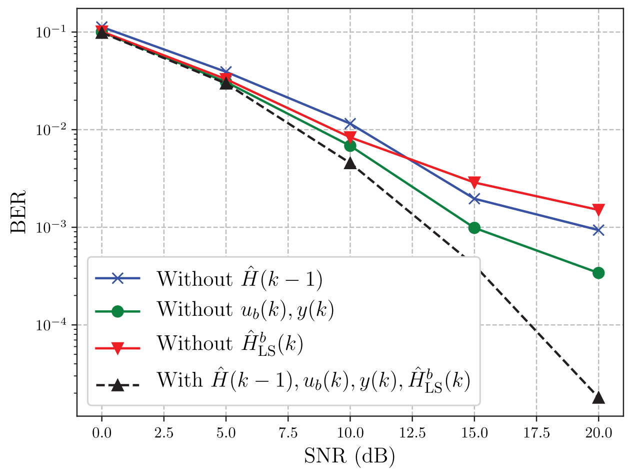

FIGURE 8. Performance comparison between the DL-based estimators with different input data structures.

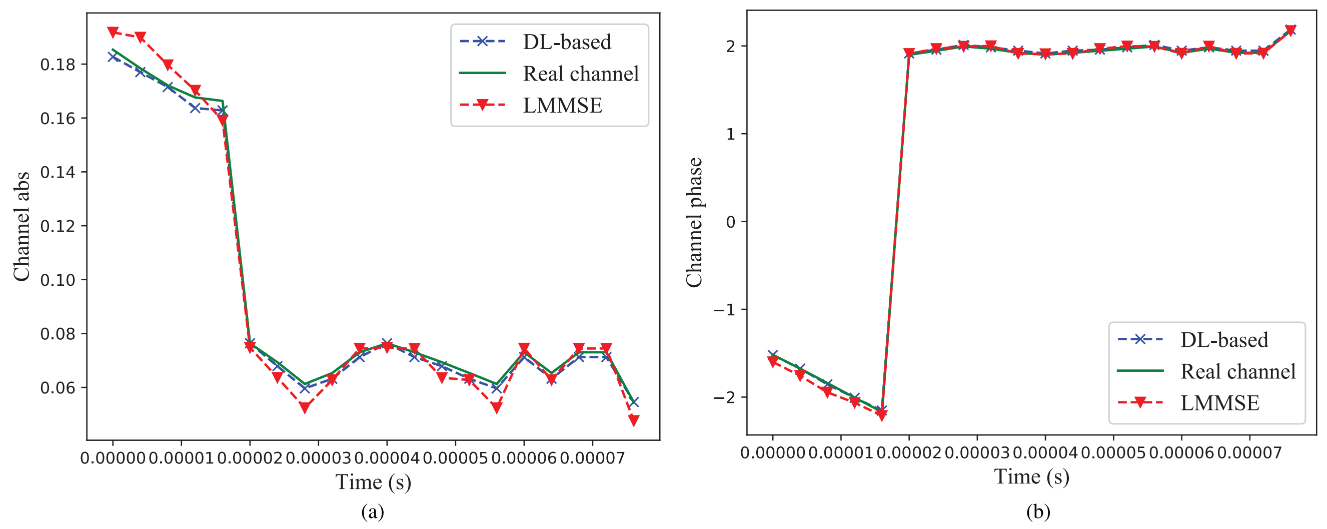

FIGURE 9. Performance of amplitude (a) and phase (b) tracking using the DL-based and LMMSE estimators. (a) Amplitude tracking. (b) Phase tracking.

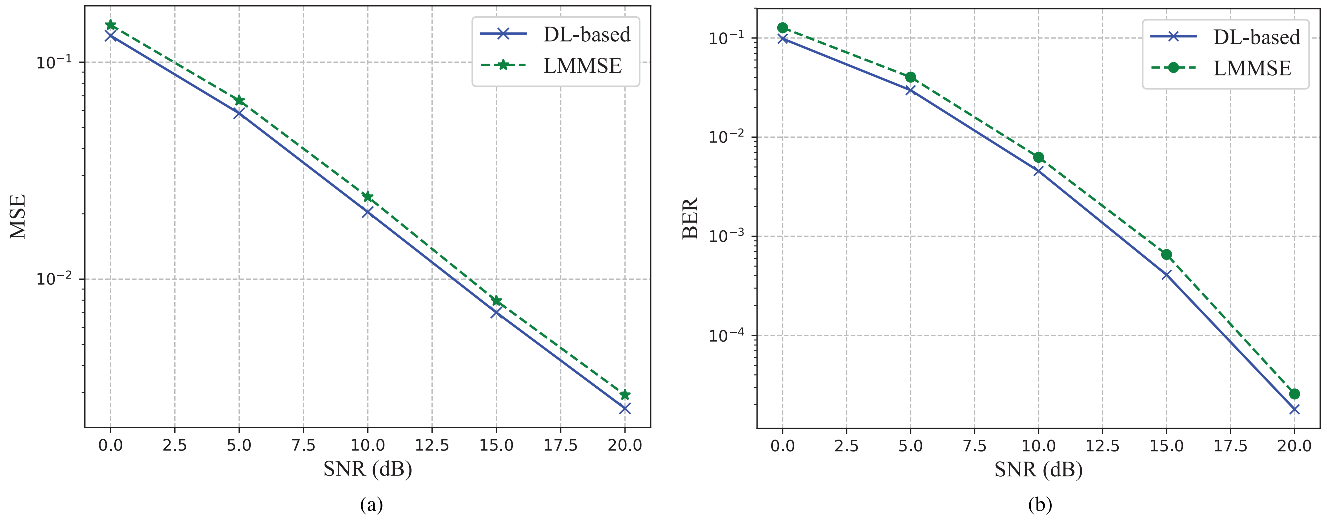

FIGURE 10. The MSE (a) and BER (b) performance of DL-based and LMMSE estimators versus SNR. (a) MSE performance versus SNR, $N_b = 12$. (b) BER performance versus SNR, $N_b = 12$.

than the LMMSE estimator and both estimators can track the phase of channels.

2) PERFORMANCE VERSUS SNR

Fig. 10 compares the MSE and BER performance of DL-based and LMMSE estimators versus SNR, where $N_b = 12$. As shown in Fig. 10a, the MSE performance of both estimators improves as SNR increases and the DL-based estimator achieves better performance than the LMMSE estimator. As shown in Fig. 10b, the BER performance of both estimators exhibits the similar results as the MSE performance. These results show the effectiveness of the DL-based estimator.

3) PERFORMANCE VERSUS N_B

Fig. 11 compares the BER performance of DL-based and LMMSE estimators versus N_b, where $Q = 1$, and SNRs are set to 10 dB and 20 dB, respectively. As shown in Fig. 11, the BER performance of both estimators improves as N_b increases and are close to saturation when N_b is more than 16. The DL-based estimator significantly outperforms the LMMSE estimator when N_b is less than 12 while the gap between the DL-based and LMMSE estimators decreases as N_b increases. This is because that when N_b is less than 8, the number of pilots is insufficient to estimate the channel for the LMMSE estimator, which results in severe performance degradation. While the DL-based estimator can compensate the performance degradation resulted from insufficient pilots with the aid of $\boldsymbol{u}_b(k)$, $\boldsymbol{r}(k)$, and $\hat{\boldsymbol{H}}(k-1)$, and therefore achieves better performance than the LMMSE estimator.

4) ROBUSTNESS ANALYSIS

In the simulations 1) - 3), the channels are generated by the Jakes model with the same statistics, i.e., the maximum Doppler frequency $f_{\max}$ is equal to $f_c v_{\max}/c = 260$ Hz. In this case, the channels in the testing stage have the same statistics with that in the pre-training and training stages for the DL-based estimator. The second-order statistics of

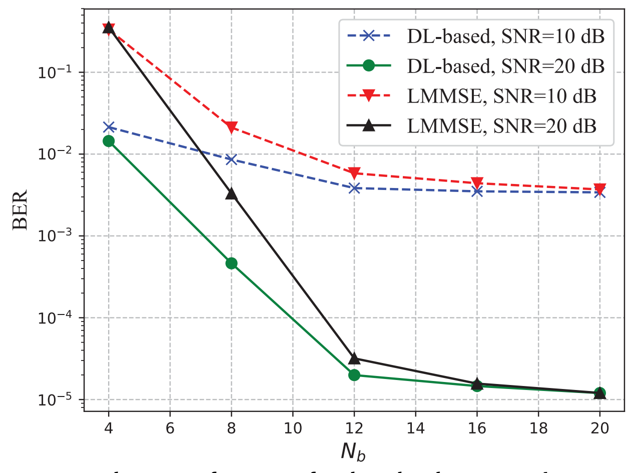

FIGURE 11. **The BER performance of DL-based and LMMSE estimators versus N_b.**

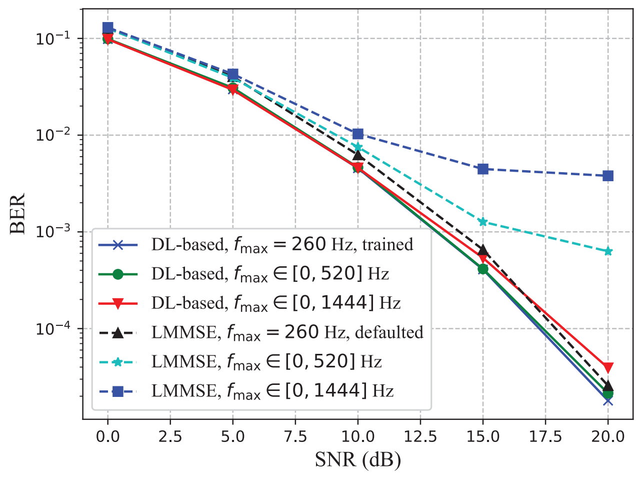

FIGURE 12. **The BER performance of DL-based estimation and LMMSE algorithms with random Doppler frequency.**

the channels are also accurate for the LMMSE estimator. However, in real-world wireless environments, the maximum Doppler frequency is often time varying and hard to track. Hence, statistics mismatches often occur between the testing and training stages for the DL-based estimator. The statistics errors are also inevitable for the LMMSE estimator.

In this example, the impacts of varying maximum Doppler frequency on the performance of both DL-based and LMMSE estimators are analyzed. The maximum Doppler frequency $f_{\max}$ is set to be 260 Hz in the training stage while $f_{\max}$ in the testing stage is set to be a random variable within two different ranges, i.e., $f_{\max} \in [0, 520]$ Hz and $f_{\max} \in [0, 1444]$ Hz, respectively. The LMMSE estimator uses the default channel statistics with $f_{\max}$ fixed as 260 Hz while the true $f_{\max}$ in testing varies randomly within the same ranges as those of DL-based estimator. The number of pilots N_b is set to be 12.

As shown in Fig. 12, the BER performance of the LMMSE estimator degrades when $f_{\max}$ varies randomly and the degradation becomes severer as SNR increases. In particular, when SNR is equal to 20 dB, the performance losses resulted from varying $f_{\max}$ reach 5 dB and 9 dB, respectively, for $f_{\max} \in [0, 520]$ Hz and $f_{\max} \in [0, 1444]$ Hz. While for the DL-based estimator, the performance losses resulted from varying $f_{\max}$ are less than 0.5 dB and 1.5 dB, respectively. These results show that the variations on statistics of channel models degrade the performance of the LMMSE estimator but have no significant influence on the performance of the DL-based estimator. These results also validate the excellent generalization ability of DL-based estimator with respect to the maximum Doppler frequency.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the DL-based estimation algorithm for doubly selective channels. We also designed the pre-training approach for the DNN to acquire a desired initialization, which can further improve the performance of the DL-based estimator. Extensive numerical experiments have been conducted to evaluate the performance of the proposed DL-based estimator, which showed that the DL-based estimator outperforms the BEM-LMMSE estimator in terms of both efficiency and robustness, and also demonstrated the great potential of DL on channel estimation and tracking. Since the proposed DL-based estimator is data-driven and does not rely on the knowledge of channel statistics, it could be a promising candidate when the channel models are unknown or difficult to model analytically, such as in high mobility vehicular communications, chemical communications, underwater communications, etc. ptimizing strategies.

REFERENCES

[1] B. R. Hamilton, X. Ma, J. E. Kleider, and R. J. Baxley, "OFDM pilot design for channel estimation with null edge subcarriers," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3145–3150, Oct. 2011.

[2] J. Fang, X. Li, H. Li, and F. Gao, "Low-rank covariance-assisted downlink training and channel estimation for FDD massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1935–1947, Mar. 2017.

[3] X. Ma, G. B. Giannakis, and S. Ohno, "Optimal training for block transmissions over doubly selective wireless fading channels," *IEEE Trans. Signal Process.*, vol. 51, no. 5, pp. 1351–1366, May 2003.

[4] Y. Liu, Z. Tan, and X. Chen, "Modeling the channel time variation using high-order-motion model," *IEEE Commun. Lett.*, vol. 15, no. 3, pp. 275–277, Mar. 2011.

[5] A. Jain, R. Upadhyay, P. D. Vyavahare, and L. D. Arya, "Stochastic modeling and performance evaluation of fading channel for wireless network design," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. Workshops (AINAW)*, vol. 2, Niagara Falls, ON, Canada, May 2007, pp. 893–898.

[6] D. K. Borah and B. T. Hart, "Frequency-selective fading channel estimation with a polynomial time-varying channel model," *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 862–873, Jun. 1999.

[7] M. Martone, "Wavelet-based separating kernels for array processing of cellular DS/CDMA signals in fast fading," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 979–995, Jun. 2000.

[8] W. Cheng, H. Zhang, L. Liang, H. Jing, and Z. Li, "Orbital-angular-momentum embedded massive MIMO: Achieving multiplicative spectrum-efficiency for mmWave communications," *IEEE Access*, vol. 6, pp. 2732–2745, Dec. 2018.

[9] J. K. Tugnait, S. He, and H. Kim, "Doubly selective channel estimation using exponential basis models and subblock tracking," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1275–1289, Mar. 2010.

[10] J. T. Dias, R. C. de Lamare, and Y. V. Zakharov, "BEM-based channel estimation for 5G multicarrier systems," in *Proc. 22nd Int. ITG Workshop Smart Antennas (WSA)*, Bochum, Germany, Mar. 2018, pp. 1–5.

[11] M. B. Sutar and V. S. Patil, "LS and MMSE estimation with different fading channels for OFDM system," in *Proc. Int. Conf. Electron., Commun., Aerosp. Technol. (ICECA)*, vol. 1, Coimbatore, India, Apr. 2017, pp. 740–745.

[12] J. K. Tugnait and S. He, "Recursive least-squares doubly-selective channel estimation using exponential basis models and subblock-wise tracking," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Las Vegas, NV, USA, Mar./Apr. 2008, pp. 2861–2864.

[13] Z. Aida and B. Ridha, "LMMSE channel estimation for block—Pilot insertion in OFDM systems under time varying conditions," in *Proc. 11th Medit. Microw. Symp. (MMS)*, Hammamet, Tunisia, Sep. 2011, pp. 223–228.

[14] K. Zhang, L. Xue, X. Liu, and X. Chen, "Performance comparison of LS, LMMSE channel estimation method in frequency and time domain for OQAM/OFDM systems," in *Proc. IEEE 17th Int. Conf. Commun. Technol. (ICCT)*, Chengdu, China, Oct. 2017, pp. 224–228.

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

[16] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[17] V. Vanhoucke, A. Senior, and M. Z. Mao, "Improving the speed of neural networks on CPUs," in *Proc. Deep Learn. Unsupervised Feature Learn. Workshop*, vol. 1, 2011, p. 4.

[18] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Commun.*, vol. 14, no. 11, pp. 92–111, 2017.

[19] H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu. (2018). "Model-driven deep learning for physical layer communications." [Online]. Available: https://arxiv.org/abs/1809.06059

[20] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang. (2018). "Deep learning in physical layer communications." [Online]. Available: https://arxiv.org/abs/1807.11713

[21] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–5.

[22] H. He, C.-K. Wen, S. Jin, and G. Y. Li. (2018). "A model-driven deep learning network for MIMO detection." [Online]. Available: https://arxiv.org/abs/1809.09336

[23] 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.

[24] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sapporo, Japan, Jul. 2017, pp. 1–6.

[25] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[26] M. Liu, T. Song, and G. Gui, "Deep cognitive perspective: Resource allocation for NOMA based heterogeneous iot with imperfect SIC," *IEEE Internet Things J.*, to be published.

[27] H. Huang, W. Xia, J. Xiong, J. Yang, G. Zheng, and X. Zhu, "Unsupervised learning-based fast beamforming design for downlink MIMO," *IEEE Access*, vol. 7, pp. 7599–7605, Dec. 2018.

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

[29] T. Wang, C. Wen, S. Jin, and G. Y. Li, "Deep learning-based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wireless Commun. Lett.*, to be published.

[30] Y. Li, X. Cheng, and G. Gui, "Co-robust-ADMM-net: Joint ADMM framework and DNN for robust sparse composite regularization," *IEEE Access*, vol. 6, pp. 47943–47952, Aug. 2018.

[31] K. Karra, S. Kuzdeba, and J. Petersen, "Modulation recognition using hierarchical deep neural networks," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw. (DySPAN)*, Maryland, MD, USA, Mar. 2017, pp. 1–3.

[32] Y. Wu, X. Li, and J. Fang, "A deep learning approach for modulation recognition via exploiting temporal correlations," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Maryland, MD, USA, Jun. 2018, pp. 1–5.

[33] T. Gruber, S. Cammerer, J. Hoydis, and S. T. Brink, "On deep learning-based channel decoding," in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Maryland, MD, USA, Mar. 2017, pp. 1–6.

[34] F. Liang, C. Shen, and F. Wu, "An iterative BP-CNN architecture for channel decoding," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 144–159, Feb. 2018.

[35] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 119–131, Feb. 2018.

[36] S. Cammerer, T. Gruber, J. Hoydis, and S. ten Brink, "Scaling deep learning-based decoding of polar codes via partitioning," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–6.

[37] 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.

[38] X. Ma, H. Ye, and Y. Li, "Learning assisted estimation for time-varying channels," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Lisbon, Portugal, Aug. 2018, pp. 1–5.

[39] H. Ji, G. Zaharia, and J. Hélard, "Performance of DSTM MIMO systems in continuously changing rayleigh channel," in *Proc. Int. Symp. Signals, Circuits Syst. (ISSCS)*, Iasi, Romania, Jul. 2015, pp. 1–4.

[40] X. Ma and G. B. Giannakis, "Maximum-diversity transmissions over doubly selective wireless channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 7, pp. 1832–1840, Jul. 2003.

[41] L. Song and J. K. Tugnait, "On designing time-multiplexed pilots for doubly-selective channel estimation using discrete prolate spheroidal basis expansion models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, vol. 3, Honolulu, HI, USA, Apr. 2007, pp. III-433–III-436.

[42] K. Zhong, X. Lei, and S. Li, "Wiener filter for basis expansion model based channel estimation," *IEEE Commun. Lett.*, vol. 15, no. 8, pp. 813–815, Aug. 2011.

[43] S. Ohno and G. B. Giannakis, "Capacity maximizing MMSE-optimal pilots for wireless OFDM over frequency-selective block Rayleigh-fading channels," *IEEE Trans. Inf. Theory*, vol. 50, no. 9, pp. 2138–2145, Sep. 2004.

[44] S.-I. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, nos. 4–5, pp. 185–196, 1993.

[45] S. De, A. Mukherjee, and E. Ullah. (2018). "Convergence guarantees for RMSProp and ADAM in non-convex optimization and their comparison to Nesterov acceleration on autoencoders." [Online]. Available: https://arxiv.org/abs/1807.06766

[46] D. P. Kingma and J. Ba. (2014). "ADAM: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1026–1034.

[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Art. Intell. Statist. (ICAIS)*, San Diego, CA, USA, 2010, pp. 249–256.

[49] S. Liang, R. Sun, Y. Li, and R. Srikant. (2018). "Understanding the loss surface of neural networks for binary classification." [Online]. Available: https://arxiv.org/abs/1803.00909

[50] L. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Atlanta, GA, USA, 2013, pp. 1139–1147.

[51] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, 1998.

[52] J. Pennington, S. Schoenholz, and S. Ganguli, "Resurrecting the sigmoid in deep learning through dynamical isometry: Theory and practice," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, California, CA, USA, 2017, pp. 4785–4795.

[53] D. Duvenaud, O. Rippel, R. Adams, and Z. Ghahramani, "Avoiding pathologies in very deep networks," in *Proc. Artif. Intell. Statist. (AISTATS)*, Reykjavik, Iceland, 2014, pp. 202–210.

Authors' photographs and biographies not available at the time of publication.