Massive MIMO Channel Estimation With an Untrained Deep Neural Network

Eren Balevi, Akash Doshi, and Jeffrey G. Andrews, *Fellow, IEEE*

***Abstract*— This paper proposes a deep learning-based channel estimation method for multi-cell interference-limited massive MIMO systems, in which base stations equipped with a large number of antennas serve multiple single-antenna users. The proposed estimator employs a specially designed deep neural network (DNN) based on the deep image prior (DIP) network to first denoise the received signal, followed by conventional least-squares (LS) estimation. We analytically prove that our LS-type deep channel estimator can approach minimum mean square error (MMSE) estimator performance for high-dimensional signals, while avoiding complex channel inversions and knowledge of the channel covariance matrix. This analytical result, while asymptotic, is observed in simulations to be operational for just 64 antennas and 64 subcarriers per OFDM symbol. The proposed method also does not require any training and utilizes several orders of magnitude fewer parameters than conventional DNNs. The proposed deep channel estimator is also robust to pilot contamination and can even completely eliminate it under certain conditions.**

***Index Terms*— Deep learning, channel estimation, massive MIMO, OFDM, deep image prior.**

I. Introduction

In MULTI-ANTENNA systems, obtaining accurate channel state information (CSI) is a central activity both for precoding the spatial streams before transmission and for coherently combining the received signals from each antenna. This is particularly true for massive multi-input multi-output (MIMO) base stations, which are by definition equipped with a very large number of antennas that transmit to many users at the same time and on the same frequency band [1]. Channel estimation is nevertheless quite challenging for multicell massive MIMO cellular networks. This is fundamentally due to pilot contamination – which is the interference of pilot symbols utilized by the users in neighboring cells – and noise, but also because operations such as matrix inversion and singular value decomposition (SVD) are impractically complex for large channel matrices. A low complexity and scalable (in terms of the number of antennas) channel estimator is very desirable for massive MIMO and current solutions have non-trivial drawbacks. This paper leverages recent developments in deep learning to design a novel deep massive MIMO channel estimator that achieves these desirable properties.

A. Related Work and Motivation

The seminal paper on massive MIMO uses a least-squares (LS) estimator [1]. Despite its low complexity, the LS estimator achieves significantly less accurate channel estimation than minimum mean square error (MMSE) estimation [2], which has been used in subsequent massive MIMO studies [3]–[5]. Although the impact of channel estimation quality is profound in massive MIMO [4], employing an MMSE estimator is undesirable for two main reasons: (i) it requires an accurate estimate of the channel correlation matrix between the base station and each user, the estimation of which requires a very large number of samples in proportion to the number of antennas and has to be repeated frequently due to mobility; and (ii) the complexity of MMSE estimation is much higher than LS estimation. For both reasons, MMSE estimation scales very poorly in terms of the base station array size.

A key challenge for massive MIMO is pilot contamination, which is a fundamental limiting factor, since small scaling fading and noise vanish as the number of antennas grows large in [1]. There are many papers that attempt reliable channel estimation for massive MIMO under pilot contamination. The key idea is usually to exploit the differences among the channel covariance matrices of different users. Specifically, [6] partitions users into groups according to the similarity of their covariance matrices, and serves them accordingly. A similar idea was utilized in [7], which developed a covariance-aware pilot assignment algorithm with some coordination among base stations. A special pilot scheduling algorithm was developed for sparse massive MIMO channels in [8]. The sparsity of massive MIMO channels was also used for channel estimation with complex iterative approximate message passing and expected-maximization (EM) algorithms in [9]. Another method based on channel statistics was presented in [10]. In addition to these, there are blind channel estimators relying on channel second-order statistics to reduce the number of pilots [11], [12]. These works require estimating large covariance matrices or assume they are somehow available for free. Furthermore, their applicability is limited to NLOS zero-mean Gaussian channels, some of which further need sparsity, which

Manuscript received June 26, 2019; revised October 2, 2019; accepted December 22, 2019. Date of publication January 3, 2020; date of current version March 10, 2020. This work was supported in part by Intel. The associate editor coordinating the review of this article and approving it for publication was T. Q. Quek. *(Corresponding author: Eren Balevi.)*

The authors are with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: erenbalevi@utexas.edu; akashsdoshi@utexas.edu; jandrews@ece.utexas.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2019.2962474

exists only for low angle delay spread(which would violate the complex Gaussian channel assumption).

The introduction of techniques from deep learning points to a potential remedy, since these techniques have been recently used for other challenging communication theory problems without closed-form solutions [13]–[16]. In particular, there are several interesting deep learning-based channel estimators for OFDM systems [17], [18], MIMO [19], and massive MIMO [20]–[22]. However, these are based on supervised deep neural networks (DNNs), which are fairly complex and typically require a large number of parameters to be trained with large datasets [23]. Thus, they do not seem suitable for online channel estimation in wireless systems, where channels change quite rapidly. Promisingly, a recent special DNN design called a *deep image prior* [24] does not require training, and thus avoids the need for a training dataset. This design is very appealing for online DNN based channel estimation, in which training a DNN may also not be tolerable in terms of latency. The deep image prior is unique in that it can improve the SNR of the received signal when there is an inherent underlying structure or correlation in the received signal. Specifically, in this model the parameters of the DNN are periodically fitted for a single sample using gradient descent as opposed to conventional neural networks, where a large dataset of input and labels are used so that a NN can learn a set of weights that predicts the correct label for each input. We modify and optimize the deep image prior architecture for massive MIMO channel estimation so as to have a moderate number of parameters and avoid cumbersome training. One of the salient features of our deep channel estimator lies in not requiring any statistical knowledge about the channel except what can be directly obtained from the received signal. This not only eliminates the need to know or learn the channel statistics, but also makes the estimator applicable to any kind of channel including Gaussian or non-Gaussian, line-of-sight (LOS) or non-LOS (NLOS), and limited or rich scattering.

B. Contributions

The main contribution of this paper is to propose a novel low complexity massive MIMO channel estimation technique that is robust to pilot contamination. The novelty is the use of a DNN for denoising prior to a conventional LS-type operation, which is trivially simple. The proposed denoising is done via a specially designed DNN similar to the deep image prior proposed recently for image processing applications [24], [25]. We redesign the input and output layers of this architecture and optimize the number of parameters to ensure low complexity, eventually reducing the number of parameters from the order of millions to hundreds or a few thousand.

We mathematically prove that this proposed deep channel estimator approaches and ultimately achieves the MMSE performance as the product of the number of base station antennas, subcarriers and coherence time interval (in terms of OFDM symbols) becomes large. The simulation results appear to confirm this for moderate dimensionality, namely a $64 \times 64 \times 64$ signal block, i.e. the number of antennas, subcarriers, and OFDM symbols are all 64. Pilot contamination is reduced in the proposed estimator by learning some prior from the interference-free region in the OFDM grid and patching these priors into the pilot contaminated areas. Additionally, we do not assume that the base stations are perfectly synchronized, and the base stations allocate the pilots to users orthogonally over the time-frequency grid for one coherence time interval. Our results reveal that under some conditions (e.g., when 5% of the OFDM grid is contaminated by neighboring cells with 4 fold weaker interference power relative to the target signal in the low signal-to-noise ratio (SNR)) the deep channel estimator can completely remove the interference. Initial results for a simplified and unoptimized version of the proposed deep channel estimator were presented in [26] for SISO OFDM communication without any theoretical analysis or consideration of co-channel interference.

The paper is organized as follows. The system model and motivation are given in Section II. The deep channel estimator is explained in detail in Section III, an analysis of which appears in Section IV. The performance of the estimator is illustrated with simulations in Section V. The paper concludes with Section VI.

II. System Model and Motivation

We consider a cellular network that has base stations with a large number of antennas and single antenna users. Specifically, base stations have M antennas and serve U users such that $U \ll M$. We assume that OFDM symbols with N_f subcarriers are transmitted in a time division duplex (TDD) frame structure. To estimate the reciprocal uplink and downlink channels, users in the same cell send orthogonal pilot sequences with length N_p. For the target base station the received signal in the frequency domain $\mathbf{Y} \in \mathbb{C}^{M \times N_f N_p}$ can be expressed as

$$\mathbf{Y} = \sum_{u \in S_u} \sqrt{\rho_u} \mathbf{H_u} \otimes \mathbf{x_u^H} + \sum_{v \in S_v} \sqrt{\rho_v} \mathbf{H_v} \otimes \mathbf{x_v^H} + \mathbf{Z}, \quad (1)$$

where S_u is the set of users connected to the target base station, ρ_u is the transmit power, $\mathbf{H_u} \in \mathbb{C}^{M \times N_f}$ is the channel between the target base station and its u^{th} user, $\mathbf{x_u} \in \mathbb{C}^{N_p \times 1}$ is the pilot sequence used for channel estimation such that $\mathbf{x_u^H x_u} = N_p$ and $\otimes$ denotes the Kronecker product. The notation is the same for the second term in the right-hand side (RHS) of (1), in which S_v is the set of users that are connected to the interferer cells and $\mathbf{H_v}$ is the channel between the target base station and the interferer users. The last term $\mathbf{Z} \in \mathbb{C}^{M \times N_f N_p}$ denotes the Gaussian noise matrix whose independent and identically distributed (i.i.d.) elements are zero-mean Gaussian random variables with variance σ^2.

Other users in other base stations can also reuse the same pilot sequences as the u^{th} user in the target cell. This is because pilots are limited by the time-frequency resources, and so it is not possible to allocate orthogonal pilots for all users in all cells. The resulting interference is known as pilot contamination. Notice that in (1) $\mathbf{x_u}$ and $\mathbf{x_v}$ can be either fully or partially synchronized depending on whether the system is synchronous or asynchronous massive MIMO, respectively. According to our system model the pilots are sent in all sub-carriers of the first OFDM symbol of a coherence block. The u^{th} user signal in the base station is obtained by

$$\mathbf{Y_u} = \mathbf{Y}(\mathbf{I_{N_f}} \otimes \mathbf{x_u}) \tag{2}$$

such that $\mathbf{Y_u} \in \mathbb{C}^{M \times N_f}$. Due to the mixed-product property of the Kronecker product

$$(\mathbf{H_u} \otimes \mathbf{x_u^H})(\mathbf{I_{N_f}} \otimes \mathbf{x_u}) = (\mathbf{H_u}\mathbf{I_{N_f}}) \otimes (\mathbf{x_u^H}\mathbf{x_u}) = N_p\mathbf{H_u}.$$

It is straightforward to express (2) as

$$\mathbf{Y_u} = \sqrt{\rho_u}N_p\mathbf{H_u} + \sum_{v \in S_v} \sqrt{\rho_v}\tilde{\mathbf{H}}_\mathbf{v} + \mathbf{Z_u}, \tag{3}$$

where $\tilde{\mathbf{H}}_\mathbf{v} = \mathbf{H_v} \otimes (\mathbf{x_v^H}\mathbf{x_u})$ and $\mathbf{Z_u} = \mathbf{Z}(\mathbf{I_{N_f}} \otimes \mathbf{x_u})$. To have more compact expressions, the matrices are defined as vectors by concatenating the columns, yielding

$$\underline{\mathbf{Y}}_\mathbf{u} = \text{vec}(\mathbf{Y_u}), \tag{4}$$

where $\underline{\mathbf{Y}}_\mathbf{u} \in \mathbb{C}^{MN_f}$. The same notation is utilized for $\underline{\mathbf{H}}_\mathbf{u}$, $\underline{\tilde{\mathbf{H}}}_\mathbf{v}$ and $\underline{\mathbf{Z}}_\mathbf{u}$. Substituting (3) with these yields

$$\underline{\mathbf{Y}}_\mathbf{u} = \sqrt{\rho_u}N_p\underline{\mathbf{H}}_\mathbf{u} + \sum_{v \in S_v} \sqrt{\rho_v}\underline{\tilde{\mathbf{H}}}_\mathbf{v} + \underline{\mathbf{Z}}_\mathbf{u}. \tag{5}$$

To estimate the channel between the u^{th} user and the target base station, (5) is multiplied with a linear matrix such that

$$\underline{\hat{\mathbf{H}}}_\mathbf{u} = \mathbf{A_u}\underline{\mathbf{Y}}_\mathbf{u}, \tag{6}$$

where

$$\mathbf{A_u} = \begin{cases} \dfrac{1}{\sqrt{\rho_u}N_p}\mathbf{I_{MN_f}} & \text{LS} \\ \sqrt{\rho_u}\mathbf{R}_{\underline{\mathbf{H}}_\mathbf{u}}(\Gamma_v + \rho_u N_p \mathbf{R}_{\underline{\mathbf{H}}_\mathbf{u}} + \sigma^2\mathbf{I_{MN_f}})^{-1} & \text{MMSE} \end{cases} \tag{7}$$

in which $\mathbf{R}_{\underline{\mathbf{H}}_\mathbf{u}}$ is the correlation matrix of the channel and

$$\Gamma_v = \sum_{v \in S_v} \rho_v \mathbf{R}_{\underline{\tilde{\mathbf{H}}}_\mathbf{v}}. \tag{8}$$

As is clear from (7), LS estimation has very low complexity, whereas MMSE estimation requires not only the autocorrelation matrices of all users that use the same pilot sequence but also a matrix inversion, the complexity of which scales as $(MN_f)^3$. Notice that there are lower complexity implementations of MMSE estimators [27], but these are still much higher complexity than LS estimation. Hence, the MMSE estimator is undesirable for a large number of antennas M and/or subcarriers N_f [9]. Despite the appeal of the LS estimator in terms of low complexity, it provides much less accurate estimation. This results in low average spectral efficiency, which is given by

$$\eta = \frac{N - N_p}{N}\mathbb{E}_{\mathbf{H_u}}[\log_2(1 + \text{SINR})], \tag{9}$$

where N is the coherence time interval and the expectation is with respect to the channel [28]. The average sum of the spectral efficiency based on (9) for LS and MMSE estimators is depicted for different combiners, namely for maximum ratio (MR), zero-forcing (ZF), and MMSE combiners in Fig. 1. There is a considerable decrease in the average sum spectral efficiency due to LS channel estimation, in particular for MMSE and ZF combiners. A channel estimation technique that exhibits MMSE estimator performance with LS estimator complexity is highly desirable.

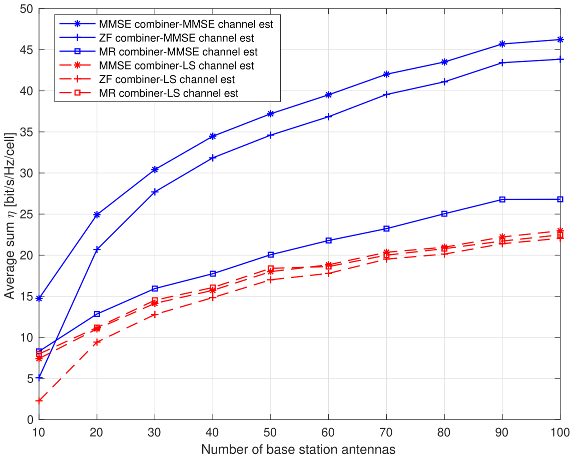


Fig. 1. Average sum spectral efficiency of LS and MMSE channel estimators for different combining techniques with increasing number of antennas.

We consider deep learning as a remedy, however the high dimensional signals are a challenge. This is because the higher the signal dimension is, the larger the number of necessary parameters in the DNN model, which needs to be trained with a dataset whose size is proportional to the number of parameters. To illustrate, a fully connected neural network for an M antenna OFDM system requires $U = MN_f$ input neurons. If there are l layers in this DNN, each of which has k_iU units for $i = 0, 1, \cdots, l-1$, this leads to $\sum_{i=0}^{l-1} k_ik_{i+1}U^2$ parameters, where $k_0 = k_l = 2$ due to the real and imaginary parts of the signal. This can easily yield millions of parameters, and thus requires a very large training dataset. To illustrate, if $M = 64$ and $N_f = 1024$, this yields approximately $5 * 10^{10}$ parameters for 6 layers when $k_i = 2$ for $i = 0, 1, \cdots 6$. Although convolutional neural networks can considerably decrease the number of parameters, a large training dataset is still necessary. This is obviously an impediment in using neural networks for real-time channel estimation, where only a very limited number of pilots (i.e. labels)[1] can be used.

In this paper, we propose a new DNN based channel estimation method **that does not require training**. Our main idea is to denoise the received signal via this DNN and then use that denoised signal for LS channel estimation instead of the raw received signal. Since the proposed estimator does not require training, there is no complexity and latency increase due to training. This also prevents the inevitable performance loss for estimators that are trained for some channel realizations but then used in others. The details of the proposed method are elaborated next.

[1] There can be some unsupervised or semi-supervised learning models that make channel estimation with no labels or with very limited labels. However, there is not any generic known channel estimation model yet for this method, and this subject remains mostly open.

III. Deep Channel Estimator Model

Training overhead is the primary obstacle to making state-of-the-art DNNs practically implementable for high-dimensional channel estimation. In the context of image processing, a recent paper shows that training is not necessary for a special DNN design, which is known as Deep Image Prior (DIP) [24]. Specifically, this design eliminates the need to map labels to inputs as in supervised learning. Rather, we simply perform gradient descent on a function of the neural network weights/parameters with the objective of making the signal predicted by the neural network as close as possible to the received signal. The main idea behind this untrained DNN or DIP model is to fit the parameters of a neural network for each channel realization on the fly – without training them on large datasets beforehand – and use the structure of the DNN as a prior. This model was later optimized to further reduce the number of required parameters [25]. Both [24] and [25] observed very efficient denoising and inpainting performance thanks to the specifically designed DNN architecture, which has low impedance for natural images and high impedance against noise. That is, the model fits more to the structured signal and less to the unstructured noise.

For massive MIMO-OFDM channel estimation, denoising and inpainting are analogous to eliminating noise and pilot contamination, respectively, and adapting DIP model to the channel estimation problem is promising. This is because (i) in communication systems, there is a limited number of pilots (or labeled data), and thus the architectures based upon large training dataset are not feasible; (ii) in conventional DNNs, training and testing have to be done for the same channel realization to obtain better performance, which brings in heavy training overhead; (iii) noise and interference are the main impediments that hinders to make a reliable channel estimation for massive MIMO; (iv) in our case, the received signal is a high-dimensional vector due to massive MIMO-OFDM and its elements are spatially, temporally and/or frequency correlated, and all these correlations in the received signal create a useful structure; and (v) DIP model can be more robust to the changes, since the model parameters are fitted periodically and it is easier to capture different channel statistics with respect to traditional DNNs. Motivated by these factors, the specifically designed DNN architecture for the DIP model is leveraged to perform channel estimation. In particular, we modify the input and output layers of one variant of the DIP architecture [25], and use it as a baseline, which we term a *deep channel estimator*.

The proposed deep channel estimator is composed of two stages. In the first stage, a less noisy signal is generated from the received signal through a specially designed DNN architecture mentioned above, and some prior information is obtained to mitigate interference. In the second stage the generated or filtered signal is multiplied by the Hermitian of the pilot sequence for channel estimation. Effectively we are proposing an LS-type channel estimator with the only difference being that the signal generated by the DNN is used instead of the received signal. By doing that the low complexity nature of LS estimator is combined with the noise reduction capability of the DNN so as to have a near MMSE estimation performance. The price paid for the proposed deep channel estimator is the need for fitting the parameters of the DNN periodically for each OFDM grid, whose period is determined by the channel coherence time (or equivalently maximum Doppler spread). However, the complexity increase is quite reasonable thanks to the low number of parameters, as will be explained.

The received signal in (1) can be equivalently written in 3-dimensional form as

$$\mathbf{Y} = \{\{\{Y[m,q,n]\}_{m=1}^{M}\}_{q=1}^{N_f}\}_{n=1}^{N}, \tag{10}$$

where $Y[m,q,n]$ is the received signal of the target base station in the m^{th} antenna for the q^{th} subcarrier in the n^{th} OFDM symbol. Notice that (10) is expressed in terms of the length of the coherence time instead of the number of pilots, which contains N OFDM symbols. This is because the parameters of the DNN has to be fitted periodically with coherence time. The real and imaginary part of (10) is separated into 2 independent channels in our architecture, since tensors do not support complex operations. This tensor representation of $\mathbf{Y}$ is denoted as $\mathbf{Y_T}$. Specifically, $\mathbf{Y_T} \in \mathbb{R}^{M \times N_f \times N \times 2}$, where the dimensions are for the spatial, frequency, time, and complex domains. In our architecture, we stack the spatial and complex domains which leads to $\mathbf{Y_T} \in \mathbb{R}^{N_s \times N_f \times N}$, where $N_s = M \times 2$.

The working principle of the deep channel estimator is to generate $\mathbf{Y_T}$ by passing a randomly chosen input tensor Z_0, which can be considered as an input filled with uniform noise, through hidden layers, whose weights are also randomly initialized, and then optimizing the weights via gradient descent. Here, Z_0 is initially randomly chosen and then kept fixed. On the other hand, the DNN parameters are continuously updated. The overall DNN model that depicts the input, output and hidden layers for a 3-dimensional communication signal is given in Fig. 2. The key component in the aforementioned DNN model is the hidden layers, which are composed of four major components. These are: (i) a 1×1 convolution, (ii) an upsampler, (iii) a rectified linear unit (ReLU) activation function, and (iv) a batch normalization. A 1×1 convolution means that each element in the time-frequency grid is processed with the same parameters through the spatial domain, which changes the dimension. More precisely, an $N_s^{(i-1)} \times 1 \times 1$ data vectors in the i^{th} hidden layer is element-wise multiplied with an $N_s^{(i-1)} \times 1 \times 1$ kernel and summed. There are $N_s^{(i)}$ different kernels, which are shared for each slot in the time-frequency axes. Hence, the spatial dimension becomes $N_s^{(i)}$. This can be equivalently considered as each vector in the time-frequency slot being multiplied with the same (shared) $N_s^{(i)} \times N_s^{(i-1)}$ matrix. In what follows, upsampling is performed to exploit the couplings among neighboring elements in the time and frequency grid. More precisely, the time-frequency signal is upsampled with a factor of 2 via a bilinear transformation. Next, the ReLU activation function is used to make the model more expressive for nonlinearities. The last component of a hidden layer performs batch normalization for a batch size of 1 to avoid vanishing gradients. This structure of a hidden layer is portrayed in Fig. 3. All the hidden layers have the

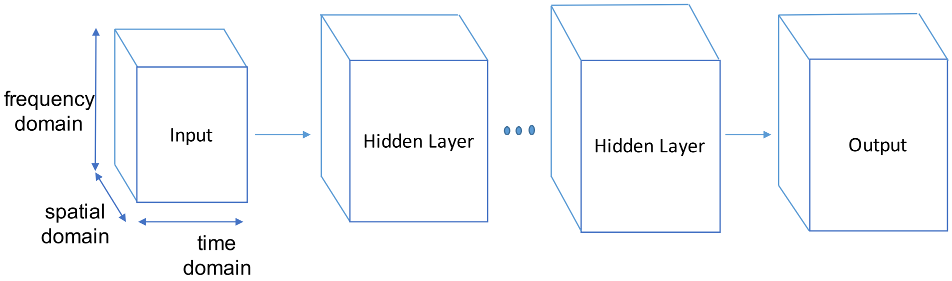


Fig. 2. The DNN architecture to denoise and inpaint the received signal before channel estimation for a 3-dimensional communication signal.

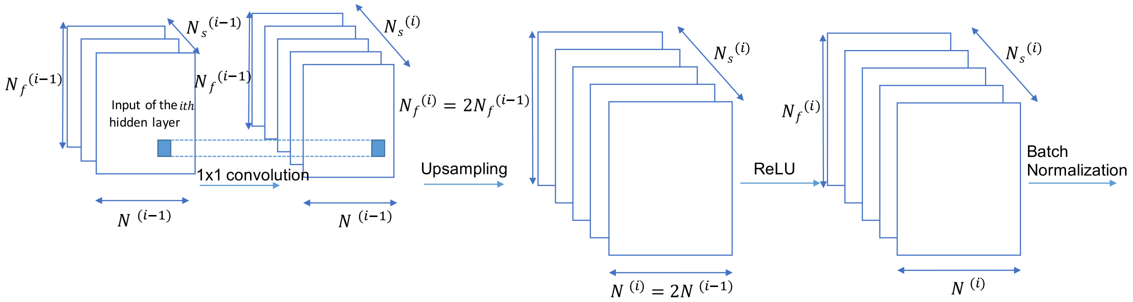


Fig. 3. The structure of the i^{th} hidden layer, whose input dimension is $N_s^{(i-1)} \times N_f^{(i-1)} \times N^{(i-1)}$ and output dimension is $N_s^{(i)} \times N_f^{(i)} \times N^{(i)}$. Note that $N_f^{(i)} = 2N_f^{(i-1)}$ and $N^{(i)} = 2N^{(i-1)}$. The spatial dimensions $N_s^{(i-1)}$ and $N_s^{(i)}$ are the hyperparameters that are used by the 1×1 convolution operations.

same structure except for the last hidden layer, which does not have an upsampler.

The mathematical representation of the aforementioned architecture is given next. Accordingly, the tensor $\mathbf{Y_T}$ is parameterized for the $l+1$ layer as

$$\hat{\mathbf{Y}}_{\mathbf{T}} = f_{\theta_l}(f_{\theta_{l-1}}(\cdots f_{\theta_0}(Z_0))), \tag{11}$$

where the input Z_0 has a dimension of $N_s^{(0)} \times N_f^{(0)} \times N^{(0)}$ in the spatial, frequency and time domain, respectively. These dimensions are determined according to the number of hidden layers and the output dimension, in which $N_s^{(l)} = N_s$, $N_f^{(l)} = N_f$, $N^{(l)} = N$. The layers from 0 to $l-1$ are counted as a hidden layer, and for $i = 0, 1, \cdots, l-2$

$$f_{\theta_i} = \text{BatchNorm}(\text{ReLU}(\text{Upsampler}(\theta_i \circledast Z_i)), \tag{12}$$

where Z_i is the input of the i^{th} hidden layer, θ_i are the parameters, and $\circledast$ represents the so-called "convolution" operator, which actually refers to cross-correlation in signal processing. More precisely, a 1×1 convolution is utilized as a cross-correlator, which means that the spatial vector for each element of the time-frequency grid is multiplied with the same shared parameter matrix to obtain the new spatial vector for the next hidden layer. The last hidden layer is

$$f_{\theta_{l-1}} = \text{BatchNorm}(\text{ReLU}(\theta_{l-1} \circledast Z_{l-1}), \tag{13}$$

and the output layer is

$$f_{\theta_l} = \theta_l \circledast Z_l. \tag{14}$$

All the parameters can be represented as

$$\Theta = (\theta_0, \theta_1, \cdots, \theta_l), \tag{15}$$

which are optimized according to the square of l_2-norm

$$\Theta^* = \arg\min_{\Theta} ||\mathbf{Y_T} - \hat{\mathbf{Y}}_{\mathbf{T}}||_2^2. \tag{16}$$

The output of the DNN for the optimized parameters is

$$\mathbf{Y}_{\mathbf{T}}^* = f_{\Theta^*}(Z_0), \tag{17}$$

where $\mathbf{Y}_{\mathbf{T}}^* = \{\{\{Y^*[m,q,n]\}_{m=1}^{M}\}_{q=1}^{N_f}\}_{n=1}^{N}$ for randomly taken Z_0. Notice that the spatial dimension of Z_0 is a hyperparameter and its other dimensions are determined by the output signal and the number of layers, since at each layer the time and frequency dimensions are doubled. Furthermore, it is worth emphasizing that in the architecture spatial correlations in the signal are captured by the 1×1 convolution so as to decrease the number of parameters, and frequency and temporal correlations are exploited by upsampling. After generating the denoised signal in (17) from a random input Z_0, an LS channel estimator is employed by multiplying (17) with the Hermitian transpose of the pilot sequence.

IV. Theoretical Analysis

The denoising capability of the proposed LS-type deep channel estimator determines how close it can approach the MMSE estimation performance. Next, we prove that our architecture can filter all the noise for asymptotically high-dimensional signals, e.g., for massive MIMO-OFDM, and can achieve the MMSE estimator performance.

Theorem 1: The proposed LS-type deep channel estimator achieves the MMSE estimator performance as the product of the number of base station antennas M, number of subcarriers N_f and coherence time interval N goes to infinity, assuming *there is no pilot contamination. That is,*

$$\lim_{MN_fN\to\infty} \frac{\epsilon_{\mathrm{dce}}}{\epsilon_{\mathrm{mmse}}} = \zeta, \tag{18}$$

where ϵ_{dce} and ϵ_{mmse} are the channel estimation errors for the proposed deep channel estimator and conventional MMSE estimator, respectively, and ζ is a scalar and depends on the number of the DNN parameters.

Proof: The proof is composed of three parts. In the first part, we generalize the noise suppression level of the architecture [25] for the deep channel estimator as

$$\min_{\Theta} \frac{||\mathbf{n} - \mathbf{n}_{\mathrm{fit}}(\Theta)||^2}{||\mathbf{n}||^2} \geq 1 - c\left(\frac{\left(\prod_{i=0}^{l-1} N_s^{(i)}\right)^{\frac{2}{l}} \log\left(\prod_{i=0}^{l-1} N_f^{(i)} N^{(i)}\right)}{N_s^{(l)} N_f^{(l)} N^{(l)}}\right), \tag{19}$$

where $\mathbf{n}$ is the noise vector in the received signal, and $\mathbf{n}_{\mathrm{fit}}(\Theta)$ shows the fitted amount of noise at the output of the deep channel estimator such that $||\mathbf{n}_{\mathrm{fit}}(\Theta)|| = 0$ means that all noise is canceled, and c is a numerical constant. Although (19) is satisfied with probability at least $1-2(N_f^{(0)}N^{(0)})^{-(\prod_{i=0}^{l} N_s^{(i)})^{\frac{2}{l+1}}}$ for $N_s^{(l)} = 1$ in [25], this probability goes to 1 for our case due to the high-dimensional massive MIMO-OFDM signal model, i.e.,

$$\lim_{M\to\infty} \left\{1 - 2(N_f^{(0)}N^{(0)})^{-(\prod_{i=0}^{l} N_s^{(i)})^{\frac{2}{l+1}}}\right\} = 1 \tag{20}$$

since $M = N_s^{(l)}/2$. Hence, with the optimum parameters Θ^*, (19) can be expressed in terms of the maximum noise suppression level

$$n_{\mathrm{supp}} = \inf\left\{\frac{||\mathbf{n} - \mathbf{n}_{\mathrm{fit}}(\Theta^*)||^2}{||\mathbf{n}||^2}\right\} = 1 - c\left(\frac{\left(\prod_{i=0}^{l-1} N_s^{(i)}\right)^{\frac{2}{l}} \log\left(\prod_{i=0}^{l-1} N_f^{(i)} N^{(i)}\right)}{N_s^{(l)} N_f^{(l)} N^{(l)}}\right) \tag{21}$$

with probability 1.

In the second part of the proof, we make use of deep learning theory regarding overparameterization. We observe that the denominator of the second term in the right-hand side of (21) scales with the dimension of the received signal, since $N_s^l = 2M$, $N_f^l = N_f$, $N^l = N$, whereas the dimension of the hidden layers does not. In particular,

$$\log\left(\prod_{i=0}^{l-1} N_f^{(i)} N^{(i)}\right) = \log\left(\frac{(N_f^{(l-1)} N^{(l-1)})^l}{\prod_{i=0}^{l-1} 4^i}\right) \tag{22}$$

due to (12) and (13), in which the denominator of the right-hand side of (22) is scaled by 4 due to the oversampling by 2 in the time and frequency axes. Since $N_f^{(l)} N^{(l)} \gg l$,

$$\log\left(\frac{(N_f^{(l-1)} N^{(l-1)})^l}{\prod_{i=0}^{l-1} 4^i}\right) \ll N_f^{(l)} N^{(l)}. \tag{23}$$

Now, we proceed to see how the spatial dimension of the hidden layers scales with the number of antennas. Accordingly, the objective function in (16) is written in terms of energy minimization [24]

$$\min_{\Theta} E(\mathbf{Y_T}, \hat{\mathbf{Y}}_\mathbf{T}). \tag{24}$$

It is standard to express (24) in terms of a function approximator $G(\cdot)$ and a regularizer $R(\cdot)$ as

$$\min_{\Theta} E(\mathbf{Y_T}, G(\Theta)) + R(G(\Theta)). \tag{25}$$

For (25), increasing the width of the last hidden layer while keeping the dimension of the other hidden layers fixed is sufficient to fit the received signal [29]. Using (25) by defining

$$R(G(\Theta)) = ||\Theta||_F^2 + \langle \mathbf{A}, \Theta^T\Theta \rangle, \tag{26}$$

where $\mathbf{A}$ is a random positive semidefinite matrix with arbitrarily small Frobenius norm, and writing the last layer of the deep channel estimator for a time-frequency slot as

$$(N_s^{(l)})^2 = (\theta_l \circledast N_s^{(l-1)})^2, \tag{27}$$

it is sufficient to increase the spatial dimension as [30]

$$N_s^{(l-1)} = \sqrt{2r}, \tag{28}$$

where r is the rank of the channel that shows the number of independent received samples. That is, it increases sublinearly with the increasing number of antennas.

In the third part, we derive the channel estimation errors in view of the first two parts. The asymptotic channel estimation error for MMSE estimator can be expressed in terms of covariance matrix

$$\epsilon_{\mathrm{mmse}} = \mathrm{tr}(\mathbf{C}), \tag{29}$$

where

$$\mathbf{C} = \mathbf{R} - \mathbf{R}\left(\mathbf{R} + \frac{1}{\mathrm{SNR}}\mathbf{I}\right)^{-1}\mathbf{R}. \tag{30}$$

In terms of the eigenvalues of the correlation matrix $\mathbf{R}$, (29) can be written using (30) as

$$\epsilon_{\mathrm{mmse}} = \sum_{m=1}^{\mathrm{rank}(\mathbf{R})} \left(\lambda_m - \frac{\lambda_m^2}{\lambda_m + 1/\mathrm{SNR}}\right). \tag{31}$$

Since uncorrelated noise vanishes in massive MIMO, i.e., $M \to \infty$, and hence $MN_fN \to \infty$ and $\mathrm{SNR} \to \infty$

$$\lim_{MN_fN\to\infty} \epsilon_{\mathrm{mmse}} = 0. \tag{32}$$

For the LS estimator, we have

$$\epsilon_{\mathrm{ls}} = \frac{\mathrm{rank}(\mathbf{R})}{\mathrm{SNR}}. \tag{33}$$

For the proposed deep channel estimator, we replace (33) with an expression for $||\mathbf{n}_{\mathrm{fit}}(\Theta^*)||^2$ from (21), and utilizing (22) and (28), we have

$$\epsilon_{\mathrm{dce}} = \frac{\mathrm{rank(R)}}{\mathrm{SNR}} \times \left((1 + \mathrm{rank(R)c} \times \frac{l\log(N_fN) - \log(4^{0.5l(l-1)})}{MN_fN}\right)^{\frac{1}{2}} - 1\right)^2. \tag{34}$$

Hence,

$$\lim_{MN_fN\to\infty} \epsilon_{\rm dce} = 0, \tag{35}$$

and this completes the proof of (18), in which ζ is the ratio of (34) to (31) for $MN_fN \to \infty$. □

Notice that if the spatial dimensions of all hidden layers are increased equally instead of only increasing the last hidden layer spatial dimension, this would result in less increase than (28). This means that the spatial dimension increases at worst with the square root of the rank of the channel. As will be shown in Section V-B, our empirical results support this argument. To illustrate, $N_s^{(l-1)} = 8$ for single antenna, whereas $N_s^{(l-1)} = 16$ for 64 antennas. Another important point regarding this theorem is that the deep channel estimator can ultimately approach zero estimation error without increasing the transmission power, instead just by increasing the number of antennas and subcarriers.

Even if there is pilot contamination in the environment, the proposed estimator can inherently resist (and even eliminate) interference up to some point. However, this holds only if the interference is isolated to a limited region of the OFDM grid. We associate this behavior with the inpainting capability of the DIP architecture [24]. This implies that the DCE can approach single cell MMSE estimation performance even for the multicell case, if the pilot contamination is sufficiently localized in time and frequency. This feature can be attributed to learning prior information from some interference-free regions and then patching this prior information into the interference regions, similar to dictionary learning [31]. The comparison of various dictionary learning methods with our estimator as well as integrating our model into one of the dictionary learning methods for enhanced interference mitigation are left as future work.

V. Simulations

The proposed deep channel estimator is compared with the traditional LS and MMSE channel estimators given in (7) using the "LTE-Extended Pedestrian A Model (EPA)" and "Kronecker" channel model. The performance metric is the normalized mean square error (NMSE), which is defined as

$$\text{NMSE} = \mathbb{E}\left[\frac{||\underline{\mathbf{H}}_{\mathbf{u}} - \hat{\underline{\mathbf{H}}}_{\mathbf{u}}||_2^2}{||\underline{\mathbf{H}}_{\mathbf{u}}||_2^2}\right], \tag{36}$$

where $\underline{\mathbf{H}}_{\mathbf{u}}$ and $\hat{\underline{\mathbf{H}}}_{\mathbf{u}}$ are the column vectors that specify the actual and the estimated channel taps in the frequency domain over all antennas, respectively. In this section, we first state the experimental details, then provide the simulation results and discuss the complexity of the estimator.

A. Experimental Details

The deep channel estimator is implemented in Pytorch [32] with 6 hidden layers, i.e., $l = 6$ as described in Section III. Without any loss of generality, the spatial dimension of the hidden layers is taken as $N_s^{(i)} = k$ for $i = 0, 1, \cdots, l-1$. Then, the number of parameters (or equivalently the value of k) is optimized using two Nvidia GeForce GTX 2070 GPUs for acceleration.[2] The performance of our estimator is evaluated for two channel models, namely the LTE-EPA and Kronecker channel model, which is commonly used to model MIMO channels. However, we present most of our results only for the LTE-EPA model, because our empirical results show that there is not any significant performance difference between these two channel models. To generate a channel realization for the LTE-EPA model, we use the MATLAB® LTE Toolbox, and obtain an $M \times 64 \times 64$ (antennas × subcarriers × symbols) channel matrix assuming that coherence time interval is larger than or equal to 64 OFDM symbols. For the Kronecker channel model, we assume an exponential spatial correlation matrix at the base station with correlation coefficient $\rho = 0.5$ [33].

As is the case of multi-cell massive MIMO, users in the same cell are assigned orthogonal pilot signals, but these can be non-orthogonal to users in neighboring cells. Our estimator does not put any constraint on the pilot arrangement, since pilots are not used while fitting the parameters of the DNN. Specifically, pilots are only used to perform LS estimation after the received signal is filtered via the DNN. To be more practical in the sense of not requiring any tight synchronization among base stations, we prefer to simulate an asynchronous massive MIMO system. Hence, the pilots of the target cell can interfere with the data symbols of the interferer cells. In the system model we assume that the pilots are sent at the beginning of each coherence time interval for each base station.

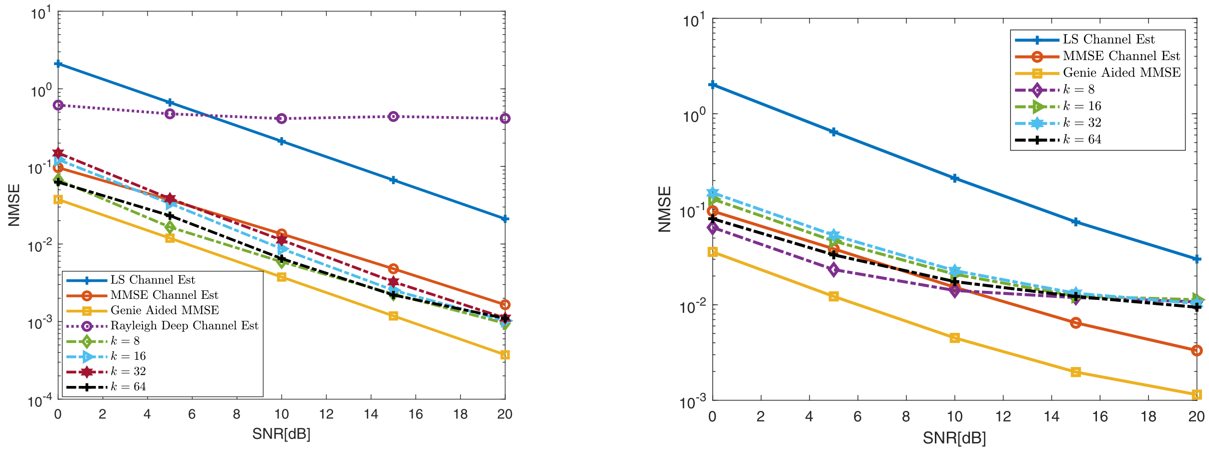
To simulate pilot contamination, we have assumed that a selected number of resource elements in the OFDM grid suffer from inter-cell interference. These interfering signals are chosen to have a multiple transmit signal of random QPSK symbols multiplied with complex channel matrices which are simply realizations generated by the LTE-EPA model. In the simulations, we also consider another pilot contamination scenario such that there is a contiguous interference both in the time and frequency domain over some number of resource elements in the OFDM grid. Throughout these simulations, SNR refers to the SNR per received antenna.

B. Results

The performance of the proposed channel estimator is first observed for traditional single antenna communication such that single antenna users transmit/receive OFDM symbols to/from a single antenna base station, i.e., $M = 1$. This enables us to quantify how the number of parameters scales with the number of antennas. We then proceed to the case of single cell massive MIMO. In other words, we consider the hypothetical scenario where all users in the cell are assumed to have orthogonal pilots and there is no pilot contamination at the base station. We finally demonstrate the robustness of our estimator in a multi-cell massive MIMO system, where pilot contamination occurs at the base station.

1) Single Antenna OFDM Communication: We first highlight our results in the case of one antenna on the base

[2] Note that the dimension of the time and frequency axes of the hidden layers are not tunable, since these are determined by the size of received signal matrix and the number of hidden layers.



(a) Without interference

(b) With co-channel interference (SIR = 6 dB)

Fig. 4. NMSE of the proposed estimator for different k and $M = 1$ with respect to SNR in comparison to LS and MMSE estimators.

station in the presence and absence of co-channel interference. For this case, the received signal or the output of the deep channel estimator is chosen to be a $2 \times 64 \times 64$ matrix, where 2 represents the real and imaginary part, the first 64 represents the number of subcarriers, and the next 64 is the number of OFDM symbols. More precisely, our architecture performs the operation outlined in (12) for $l - 1 = 5$ times, then the last hidden layer calculates the expression in (13), and finally the output layer brings the output signal to the required channel matrix size, i.e., $2 \times 64 \times 64$.

To find the optimum number of parameters in the absence of co-channel interference, we simply add AWGN to the desired signal, and adjust its variance so as to have an SNR between 0 and 20 dB range. In order to optimize the number of channels per layer or the value of k, we take a single channel realization disturbed by the least noise (i.e the highest SNR in our range) and observe the convergence of its NMSE with the number of epochs by performing Adam optimization [34] with a learning rate of 0.01. We find the number of epochs at which the lowest NMSE is achieved for a given k, and proceed to denoise the received signal for the aforementioned range of SNRs for the calculated number of epochs. This approach is often referred to as *early stopping*. The number of epochs is tabulated in Table I as a function of k and the total number of weights in the architecture.

As depicted in Fig. 4(a), the NMSE is lowest for $k = 8$, gets progressively higher for $k = 16$ and $k = 32$, and once again decreases for $k = 64$, almost equal to $k = 8$ at an SNR of 0 dB. However, there is very little to tell apart the different architectures at SNR of 20 dB. This is beacuase larger noise levels require smaller values of k. If the noise is significantly stronger, then we can either choose smaller k or use early stopping. In this plot, the MMSE curve is obtained using the channel correlation matrix that is computed via Monte Carlo simulation as outlined in [2], whereas the “Genie Aided MMSE” assumes that the channel correlation matrix is available for free, which is impractical. Promisingly, our channel estimator for $k = 8$ clearly outperforms LS and MMSE estimators and approaches the “Genie Aided MMSE” performance without having any (statistical) information other than the received signal. To have a better understanding of why the deep channel estimator works so well, we observe its performance for an unrealistic case, in which each subcarrier in the OFDM grid has an i.i.d Rayleigh fading channel. In this case, our estimator does not perform well, which implies that its success can be attributed to exploiting correlations.

TABLE I
HYPERPARAMETER TUNING FOR $M = 1$

k	Epochs	Total Weight Count
8	2000	496
16	1300	1760
32	900	6592
64	250	25472

To find the optimum number of parameters in the presence of co-channel interference, 10% of the resource elements of the OFDM grid expressed in (10) are corrupted by injecting interference that is 6 dB weaker than the desired signal, i.e., SIR = 6 dB. As shown in Fig. 4(b), clearly $k = 8$ outperforms $k = 64$ for SNRs less than 10 dB, after which their performance is similar. Hence, it is reasonable to take $k = 8$ in the architecture for the case where $M = 1$. We observed that with the addition of co-channel interference, stopping earlier than was ascertained in the interference-free case could be beneficial, however we did not change the number of epochs for which the training was performed. This is because in a practical scenario, where we do not have access to the noiseless received signal, we cannot ascertain when to stop. It has to be determined beforehand. Even without dynamically adapting early stopping, the deep channel estimator with $k = 8$ beats MMSE estimator up to 10 dB, which also means that it has better interference mitigation capability.

2) Single-Cell Massive MIMO: The deep channel estimator is mainly intended for multiple antennas in this paper. Thus, M is set to 64 and a $128 \times 64 \times 64$ matrix is obtained by concatenating the real and imaginary part of the signal with the antennas in the spatial axis. Here, the spatial domain is used to stack up the real and imaginary domain, because this axis is

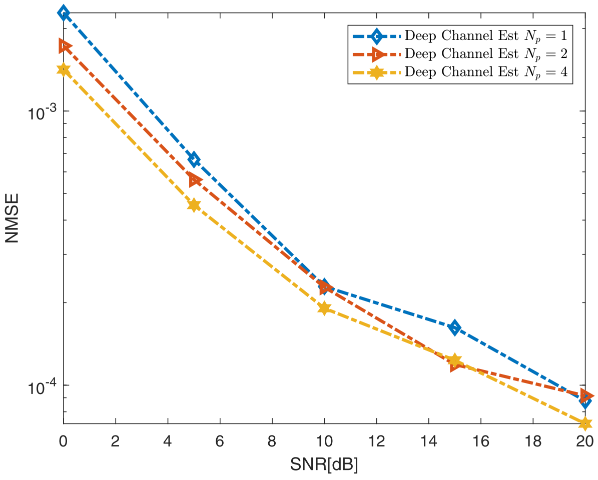


Fig. 5. NMSE of the proposed estimator for different amount of pilots N_p.

TABLE II

HYPERPARAMETER TUNING FOR $M = 64$

k	Epochs	Total Weight Count
8	4000	1504
16	1970	3776
32	1800	10624
64	1000	33536

more appropriate for uncorrelated samples in the architecture. In this case, we first observe the impact of an increased number of pilots by varying $N_p = 1, 2, 4$ under the assumption of block type pilot arrangement. Accordingly, the pilots are transmitted periodically over all the subcarriers assuming that the coherence bandwidth is equal to one subcarrier without any loss of generality. The results are shown in Fig. 5. Clearly an increase from $N_p = 1$ to $N_p = 4$ benefits the NMSE, but no benefit is obtained beyond that. This is because the LTE-EPA model which has very high temporal correlation, and consequently needs very few pilots in the time domain to represent the channel accurately. For the rest of the experiments,[3] we adopt $N_p = 1$. Notice that although there is a single OFDM pilot symbol, orthogonal pilot sequences can still be formed using the frequency domain thanks to the OFDM.

Following the same procedure that was adopted for single antenna communication, the optimum number of parameters is determined first, which is tabulated in Table II in terms of k, the number of epochs and the total number of weights. In what follows, the NMSE as a function of SNR is plotted for different values of k. As depicted in Fig. 6, the results perfectly reconcile with the single antenna case. That is, at larger noise levels (or lower SNR), smaller values of k perform much better. However at higher SNR, due to early stopping, all the architectures tend to the same NMSE, with the higher k ones performing slightly better. Accordingly, $k = 16$ appears to have the best performance.

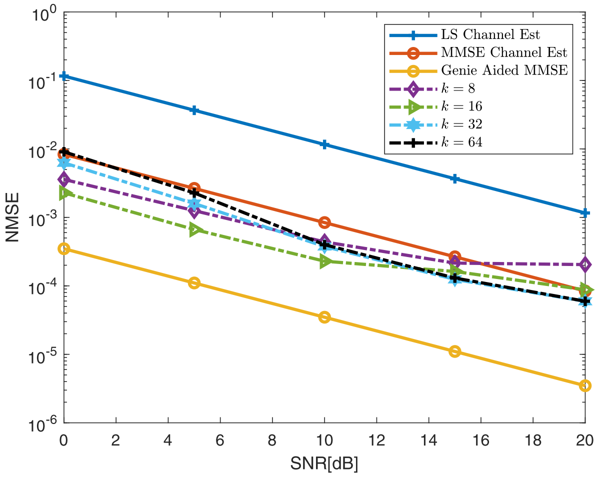


Fig. 6. NMSE of the proposed estimator for different k and $M = 64$ with respect to SNR in comparison to LS and MMSE estimators.

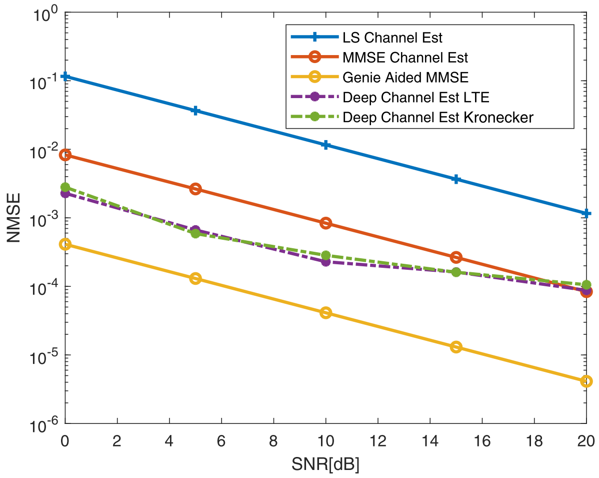
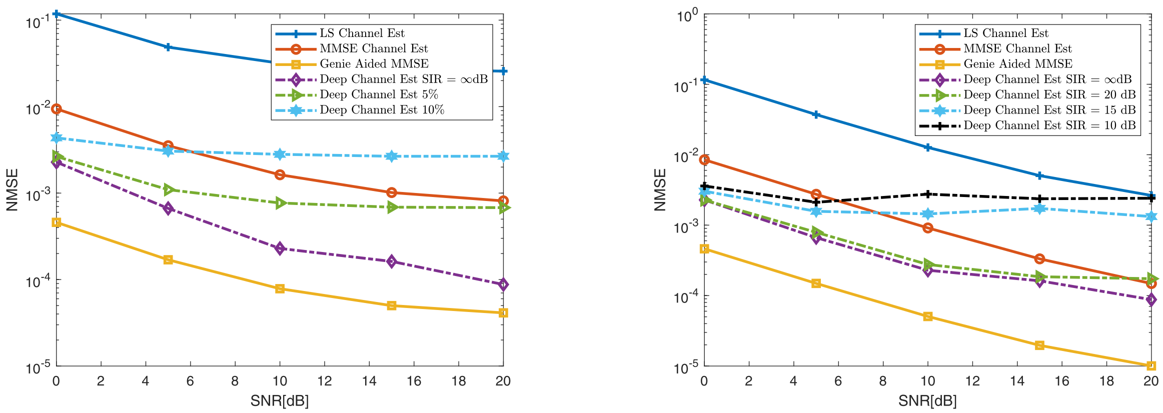


Fig. 7. NMSE of the proposed estimator for different channel models.

We repeat this experiment for the Kronecker channel model by taking the exponential spatial correlation matrix at the base station with correlation coefficient $\rho = 0.5$. However, as shown in Fig. 7, the performance of the deep channel estimator is almost unaffected by the change in spatial correlation of the channel matrix. This is because oversampling is not utilized in this domain. Hence, for brevity the results are only presented for the LTE-EPA model in the rest of the paper.

3) Multi-Cell Massive MIMO: To assess the robustness of the deep channel estimator against pilot contamination, we first search for the optimum value of k, and then exhibit the results. Since base stations allocate random pilots that are spread over the OFDM grid in one coherence time interval, the optimum value of k is searched after contaminating 5% of the time-frequency grid randomly (but across all antennas) with interference at an SIR of 6 dB. In particular, we checked whether $k = 16$ is the optimal architecture as in the case of single cell massive MIMO. We found $k = 16$ to outperform all the other architectures, hence the architecture is optimized with $k = 16$ for the rest of the multi cell massive MIMO experiments.

[3] A random pilot arrangement is used instead of a block type pilot arrangement in the case of multi-cell massive MIMO, which refers to that pilot tones are allocated to the subcarriers that belong to different OFDM symbols.



(a) Random pilot contaminations at SIR = 6 dB. Reference curves are in the presence of 10% interference at SIR = 6 dB.

(b) Contiguous pilot contamination. Reference curves are at SIR = 10 dB.

Fig. 8. NMSE of the proposed estimator for $k = 16$ and $M = 64$ with respect to SNR in comparison to LS and MMSE estimators.

This experiment is extended by also contaminating 10% of the OFDM grid for $k = 16$. The results for both 5% and 10% contamination are presented in Fig. 8(a). In this case, the deep channel estimator outperforms MMSE estimator up to an SNR of 7 dB even in the presence of up to 10% pilot contamination. The flattening out of the NMSE curve with increased interference is due to not patching the signal in the areas corrupted by interference beyond a certain limit. In image processing, this corresponds to an upper bound on the size of patches that can be recovered by region inpainting.

To further quantify the pilot contamination performance of our estimator, we verify its robustness for a different power allocation method. Accordingly, pilots are not only randomly but also contiguously distributed over the resource elements. To be more precise, 2 blocks of 8×8 squares (corresponding to $\sim\ 3\%$ of the overall time-frequency grid) are chosen randomly, in which interference at SIRs of 10, 15 and 20dB is injected. Although the deep channel estimator in this case can tolerate lower powers of interference than the previous case, its performance, as illustrated in Fig. 8(b), is still better than LS estimator for all SNRs and MMSE estimator up to an SNR of 6 dB for the SIRs that are greater than 10 dB.

C. Complexity

Regarding the complexity of the deep channel estimator, it is important to note the trade-off between the number of parameters and the number of epochs required. As seen in Table I, a larger number of epochs is required to attain the minimum NMSE for a lower number of parameters. For instance, in the case of a single antenna the number of parameters for $k = 8$ are 496, in which the NMSE are the least, but it requires 2000 epochs to attain this NMSE. On the other hand, the $k = 64$ architecture has 25,472 parameters, and attains a slightly higher NMSE than $k = 8$, but requires a mere 250 epochs to attain this NMSE. This result has in fact been proven for the case of supervised learning of a single hidden layer neural network in [35], where they show that as the degree of overparameterization of the NN increases, it takes fewer epochs to converge to one of the many global minima in its objective function's landscape. As a result, if the deep channel estimator was to be deployed in a latency critical application and subject to online training, where a slightly higher NMSE could be tolerated, one should use the model with a higher k value. For the case of 64 antennas, the optimal architecture surfaces for $k = 16$, which has only 3776 parameters but requires 1970 epochs to attain its lowest NMSE. For training around the same number of epochs such as 2000, the single antenna architecture has 496 parameters, while the massive MIMO architecture has 3776 parameters. This comparison is quite important, and specifies the sub-linear increase in computational complexity with the number of antennas.

For a more quantitative analysis, consider the number of multiply-accumulate operations (MACs) for the DNN. This can be easily seen to be directly proportional to the number of channels, k, which for a multi-user setting is simply the number of receive antennas M. Therefore the complexity of the deep channel estimator is $O(M)$. On the other hand, for the MMSE estimator, as explained in [28], the precomputation of $\mathbf{A_u}$ in (7) requires $(4M^3 - M)/3$ complex multiplications and M complex divisions per UE. This precomputation has to be performed only when the channel statistics have changed substantially (e.g. due to UE mobility). In our case, since we assume a pedestrian channel model LTE-EPA, it has to be performed only once. Every coherence block, however, to multiply $\mathbf{A_u}$ with $\underline{\mathbf{Y}}_\mathbf{u}$ in (6), it requires $O(M^2)$ computations for the matrix multiplication. Therefore, an MMSE estimator requires $O(M^n)$ computations where $2 \leq n < 3$. This clearly indicates the immense complexity reduction brought about by the deep channel estimator for large M.

VI. Conclusion

In this paper we proposed a novel deep channel estimator comprised of a DNN followed by a simple LS-type estimator. This deep channel estimator exhibits superior performance compared to LS and MMSE estimators that have no inherent way of dealing with pilot contamination (or co-channel interference). Promisingly, our low-complexity estimator outperforms the more complex MMSE estimator, in which the channel correlation matrices are estimated from the samples, and even approaches the "Genie-Aided MMSE" where the channel statistics are perfectly known for free. The deep channel estimator exploits correlations in the time-frequency grid very efficiently. The strong performance is also explained by a supporting mathematical analysis. The salient features of the proposed estimator are as follows. The number of parameters scale at a rate less than the square root of number of antennas, which yields hundreds or thousands of weights as opposed to millions of parameters in conventional DNNs. Furthermore, the proposed estimator is appropriate for any environment or channel type, since it only needs the received signal and some pilots.

It would be interesting as future work to study the deep channel estimator for high mobility channels. Similarly, observing the performance of the deep channel estimator for mmWave channels seems intriguing. Furthermore, enhancing its interference mitigation capability at high SNRs can also be a good future research direction. In particular, some other dictionary learning algorithms can be adapted to our model. Additionally, it would be interesting to observe how our estimator performs when the eigenspace of the covariance matrices of interfering users does not fully overlap with the target user.

REFERENCES

[1] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[2] J.-J. van de Beek, O. Edfors, M. Sandell, S. Wilson, and P. Borjesson, "On channel estimation in OFDM systems," in *Proc. IEEE VTC*, vol. 2, Mar. 1995, pp. 815–819.

[3] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "The multicell multiuser MIMO uplink with very large antenna arrays and a finite–dimensional channel," *IEEE Trans. Commun.*, vol. 61, no. 6, pp. 2350–2361, Jun. 2013.

[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] A. Adhikary, A. Ashikhmin, and T. L. Marzetta, "Uplink interference reduction in large scale antenna systems," *IEEE Trans. Commun.*, vol. 5, no. 65, pp. 2194–2206, May 2017.

[6] A. Adhikary, J. Nam, J.-Y. Ahn, and G. Caire, "Joint spatial division and multiplexing-the large-scale array regime," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6441–6463, Oct. 2013.

[7] H. Yin, D. Gesbert, M. Filippou, and Y. Liu, "A coordinated approach to channel estimation in large–scale multiple–antenna systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 264–273, Feb. 2013.

[8] L. You, X. Gao, A. L. Swindlehurst, and W. Zhong, "Channel acquisition for massive MIMO–OFDM with adjustable phase shift pilots," *IEEE Trans. Signal Process.*, vol. 64, no. 6, pp. 1461–1476, Mar. 2016.

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

[10] H. Yin, L. Cottatellucci, D. Gesbert, R. R. Müller, and G. He, "Robust pilot decontamination based on joint angle and power domain discrimination," *IEEE Trans. Signal Process.*, vol. 64, no. 11, pp. 2990–3003, Jun. 2016.

[11] H. Q. Ngo and E. G. Larsson, "EVD-based channel estimation in multicell multiuser MIMO systems with very large antenna arrays," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 3249–3252.

[12] R. R. Müller, L. Cottatellucci, and M. Vehkapera, "Blind pilot decontamination," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 773–786, Oct. 2014.

[13] 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.

[14] S. Dorner, S. Cammerer, J. Hoydis, and S. T. Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

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

[16] E. Balevi and J. G. Andrews, "Online antenna tuning in heterogeneous cellular networks with deep reinforcement learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1113–1124, Dec. 2019.

[17] 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.

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

[19] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for joint MIMO channel estimation and signal detection," Feb. 2019, *arXiv:1907.09439*. [Online]. Available: https://arxiv.org/abs/1907.09439

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

[21] 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.

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

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org.

[24] V. Lempitsky, A. Vedaldi, and D. Ulyanov, "Deep image prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9446–9454.

[25] R. Heckel and P. Hand, "Deep decoder: Concise image representations from untrained non-convolutional networks," Feb. 2019, *arXiv:1810.03982*. [Online]. Available: https://arxiv.org/abs/1810.03982

[26] E. Balevi and J. G. Andrews, "Deep learning-based channel estimation for high-dimensional signals," Apr. 2019, *arXiv:1904.09346*. [Online]. Available: https://arxiv.org/abs/1904.09346

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

[28] E. Björnson *et al.*, *Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency* (Foundations and Trends in Signal Processing). Boston, MA, USA: Now, 2017.

[29] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. NIPS*, May 2010.

[30] S. S. Du and J. D. Lee, "On the power of over-parametrization in neural networks with quadratic activation," Jun. 2018, *arXiv:1803.01206*. [Online]. Available: https://arxiv.org/abs/1803.01206

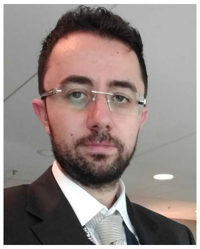
[31] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Proc. NIPS*, 2009, pp. 1033–1040.

[32] A. Paszke, S. Gross, S. Chintala, and G. Chanan, "Pytorch: Tensors and dynamic neural networks in Python with strong GPU acceleration," *PyTorch, Tensors Dyn. Neural Netw. Python Strong GPU Acceleration*, vol. 6, 2017. [Online]. Available: https://github.com/pytorch/pytorch

[33] S. Loyka, "Channel capacity of MIMO architecture using the exponential correlation matrix," *IEEE Commun. Lett.*, vol. 5, no. 9, pp. 369–371, Sep. 2001.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[35] M. Li, M. Soltanolkotabi, and S. Oymak, "Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks," Mar. 2019, *arXiv:1903.11680*. [Online]. Available: https://arxiv.org/abs/1903.11680



Eren Balevi received the B.S., M.S., and Ph.D. degrees in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, in 2008, 2010, and 2016, respectively. He is currently a Post-Doctoral Research Scholar with the Department of Electrical and Computer Engineering, The University of Texas at Austin. His current research interests lie mainly in the intersection between machine learning and communication theory. He is also interested with the general areas of 5G and beyond wireless systems, fog/edge networking, and molecular communications.

Akash Doshi received the B.Tech. degree (Hons.) in electrical engineering from IIT Bombay in 2018, with a minor in computer science. He is currently pursuing the M.S. and Ph.D. degrees with the Department of Computer and Electrical Engineering, The University of Texas at Austin. His research interests lie mainly in the intersection between wireless networks and machine learning.

Jeffrey G. Andrews (Fellow, IEEE) received the B.S. degree (Hons.) in engineering from the Harvey Mudd College and the M.S. and Ph.D. degrees in electrical engineering from Stanford University.

He has developed code division multiple access systems at Qualcomm from 1995 to 97 and has consulted for entities, including Apple, Samsung, Verizon, AT&T, the WiMAX Forum, Intel, Microsoft, Clearwire, Sprint, and NASA. He is currently the Cullen Trust Endowed Professor (#1) of ECE, The University of Texas at Austin. He is also a coauthor of the books *Fundamentals of WiMAX* (Prentice-Hall, 2007) and the *Fundamentals of LTE* (Prentice-Hall, 2010). He is a member of the Technical Advisory Board of Artemis Networks and GenXComm. He is also an Elected Member of the Board of Governors of the IEEE Information Theory Society. He is an ISI Highly Cited Researcher and has been a co-recipient of paper awards, including the 2016 IEEE Communications Society and Information Theory Society Joint Paper Award, the 2014 IEEE Stephen O. Rice Prize, the 2014 and 2018 IEEE Leonard G. Abraham Prize, the 2011 and 2016 IEEE Heinrich Hertz Prize, and the 2010 IEEE ComSoc Best Tutorial Paper Award. He received the 2015 Terman Award and the NSF CAREER Award. He is also the Chair of the IEEE Communications Society Emerging Technologies Committee. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2014 to 2016.