Deep Channel Learning For Large Intelligent Surfaces Aided mm-Wave Massive MIMO Systems

Ahmet M. Elbir, *Senior Member, IEEE*, Anastasios Papazafeiropoulos, *Senior Member, IEEE*, Pandelis Kourtessis, and Symeon Chatzinotas, *Senior Member, IEEE*

***Abstract*—This letter presents the first work introducing a deep learning (DL) framework for channel estimation in large intelligent surface (LIS) assisted massive MIMO (multiple-input multiple-output) systems. A twin convolutional neural network (CNN) architecture is designed and it is fed with the received pilot signals to estimate both direct and cascaded channels. In a multi-user scenario, each user has access to the CNN to estimate its own channel. The performance of the proposed DL approach is evaluated and compared with state-of-the-art DL-based techniques and its superior performance is demonstrated.**

***Index Terms*—Deep learning, channel estimation, large intelligent surfaces, massive MIMO.**

I. Introduction

The massive MIMO (multiple-input multiple-output) architecture has been suggested as a promising technology for fifth generation (5G) communications systems by providing high spectral efficiency exploiting high spatial multiplexing gains [1], [2]. However, its proposed marriage with millimeter wave (mm-Wave) transmission comes with the expensive cost of energy consumption and hardware complexity, even if hybrid beamforming is employed [1], [3]. Recently, large intelligent surface (LIS) (also known as reflective intelligent surface) technology has been proposed as a promising solution with low cost and hardware complexity [4]. An LIS is an electromagnetic 2-D surface that is composed of large number of passive reconfigurable reflecting elements which are fabricated from meta-materials [5].

LIS includes a programmable meta-surface which can be controlled via external signals such as backhaul control link from the base station (BS). Hence, real-time manipulation of the reflected phase and magnitude becomes possible. This property allows us to use LIS in wireless communications as a reflecting surface between the BS and the users to improve the received signal energy, expanding the coverage as well as reducing the interference [6]. While LIS can provide low-cost and simplistic architecture, it brings a difficulty of including two wireless channels between the BS and user, one being the direct channel and another one is the cascaded channel between the BS and the users through LIS [7]–[9].

This work was supported in part by the ERC project AGNOSTIC.

A. M. Elbir is with the EE department of Duzce University, Duzce, Turkey. E-mail: ahmetmelbir@gmail.com.

A. Papazafeiropoulos is with the CIS Research Group, University of Hertfordshire, Hatfield, U. K. and with SnT at the University of Luxembourg, Luxembourg. E-mail: tapapazaf@gmail.com.

P. Kourtessis is with the CIS Research Group, University of Hertfordshire, Hatfield, U. K. E-mail: p.kourtessis@herts.ac.uk

S. Chatzinotas is with the SnT at the University of Luxembourg, Luxembourg. Email:symeon.chatzinotas@uni.lu.

Regarding channel estimation in LIS, a transmission protocol has been proposed in [7] for orthogonal frequency division multiplexing, while [8] proposed a sparse matrix factorization approach. Moreover, a dual ascent-based estimation has been considered in [9]. One of the main challenges in LIS-assisted wireless networks is that the channel estimation complexity is high due to the large number of LIS elements. To lower the complexity, deep learning (DL) techniques can be of help [7]–[9]. By training a DL network with different channel characteristics, it can adapt to the changes in the environment such as the user motions and provide robust performance. Also, updating the channel information can be done less frequently, which lowers the complexity [10]. Notably, for LIS-assisted massive MIMO, DL has been applied for the reflected beamformer design [11] and signal detection [12]. Especially, in [12], the transmitted symbols are estimated without channel estimation. This approach is symbol-dependent, i.e., when the modulation type is changed, the deep network cannot identify the symbols and requires further training. To the best of our knowledge, this is the first work studying DL for channel estimation in an LIS scenario.

In this letter, we propose a DL approach for channel estimation in a LIS-assisted mm-Wave massive MIMO systems. In the proposed DL framework, we design a twin convolutional neural network (CNN) for the estimation of direct (BS-user) and cascaded (BS-LIS-user) channels and assume that each user has access to the deep network to estimate its own channel. The CNN is fed with the received pilot signals and it constructs a non-linear relationship between the received signals and the channel data.

The deep network is trained with several channel realizations to obtain a robust estimation performance. In the prediction stage, a test data, which is separately generated than the training data, is used to validate the performance. Finally, we show that the proposed DL framework achieves reasonable channel estimation accuracy and outperforms the existing DL-based techniques [10], [13]. Furthermore, the results show that the proposed DL approach has robust channel estimation performance, which is tolerant to the changes in the user locations up to 4 degrees.

II. System Model and Problem Formulation

We consider an LIS-aided mm-Wave massive MIMO system as shown in Fig. 1. We assume that the BS has M antennas to serve K single-antenna users with the assistance of LIS which is composed of L passive reflecting elements. In LIS-assisted

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2020.2993699, IEEE Wireless Communications Letters

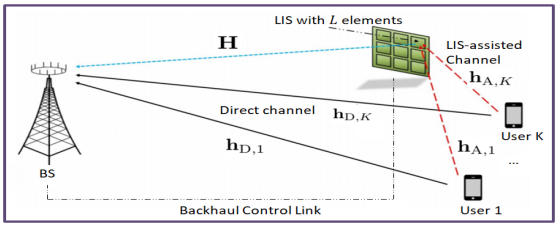

Fig. 1. An LIS-assisted mm-Wave massive MIMO scenario.

communication scheme, each LIS element introduces a phase shift onto the incoming signal from the BS. The phase of each LIS element can be adjusted through the PIN diodes which are controlled by the LIS-controller connected to the BS over the backhaul link [9], [14].

The BS transmits K data symbols $s_k \in \mathbb{C}$ by using a baseband precoder $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_K] \in \mathbb{C}^{M\times K}$. Hence, the downlink $M \times 1$ transmitted signal becomes $\bar{\mathbf{s}} = \sum_{k=1}^{K} \sqrt{\gamma_k}\bar{\mathbf{f}}_k s_k$, where $\bar{\mathbf{f}}_k = \frac{\mathbf{f}_k}{||\mathbf{f}_k||_2}$ and γ_k denotes the allocated power at the k-th user. The transmitted signal is received from the k-user with two components, one of which is through the direct path from the BS and the another one is through the LIS. The received signal from the k-th user can be given by

$$y_k = \big(\mathbf{h}_{\mathrm{D},k}^{\mathsf{H}} + \mathbf{h}_{\mathrm{A},k}^{\mathsf{H}}\boldsymbol{\Psi}^{\mathsf{H}}\mathbf{H}^{\mathsf{H}}\big)\bar{\mathbf{s}} + n_k, \tag{1}$$

where $n_k \sim \mathcal{CN}(o, \sigma_n^2)$ and $\mathbf{h}_{\mathrm{D},k} \in \mathbb{C}^M$ denotes the direct channel between the BS and the k-th user. The vector $\mathbf{h}_{\mathrm{A},k} \in \mathbb{C}^L$ expresses the LIS-assisted channel between the LIS and the k-th user. $\boldsymbol{\Psi} \in \mathbb{C}^{L\times L}$ is a diagonal matrix, i.e., $\boldsymbol{\Psi} = \mathrm{diag}\{\beta_1 \exp(j\phi_1), \ldots, \beta_L \exp(j\phi_L)\}$. Here, $\beta_l \in \{0, 1\}$ represents the on/off state of the LIS elements. In practice, the LIS elements cannot be perfectly turned on/off, Hence, β_l can be modeled as $\beta_l = \begin{cases} 1-\epsilon_1 & \text{ON} \\ 0+\epsilon_0 & \text{OFF} \end{cases}$ for $\epsilon_1, \epsilon_0 \geq 0$ [15]. $\phi_l \in [0, 2\pi)$ is the phase shift of the reflective elements. Finally, the channel between the LIS and the BS is represented by $\mathbf{H} \in \mathbb{C}^{M\times L}$.

In mm-Wave transmission, the channel can be represented by the Saleh-Valenzuela (SV) model where a geometric channel model is adopted with limited scattering [7], [14]. Hence, we assume that the mm-Wave channels, i.e., $\mathbf{h}_{\mathrm{D},k}, \mathbf{h}_{\mathrm{A},k}$ and $\mathbf{H}$, include the contributions of N_{D}, N_{A} and N_{H} paths, respectively. Thus, we can represent the channels $\mathbf{h}_{\mathrm{D},k}$ and $\mathbf{h}_{\mathrm{A},k}$ as $\mathbf{h}_{\mathrm{D},k} = \sqrt{\frac{M}{N_{\mathrm{D}}}} \sum_{n_{\mathrm{D}}=1}^{N_{\mathrm{D}}} \alpha_{\mathrm{D},k}^{(n_{\mathrm{D}})} \mathbf{a}_{\mathrm{D}}(\theta_{\mathrm{D},k}^{(n_{\mathrm{D}})})$, and $\mathbf{h}_{\mathrm{A},k} = \sqrt{\frac{L}{N_{\mathrm{A}}}} \sum_{n_{\mathrm{A}}=1}^{N_{\mathrm{A}}} \alpha_{\mathrm{A},k}^{(n_{\mathrm{A}})} \mathbf{a}_{\mathrm{A}}(\theta_{\mathrm{A},k}^{(n_{\mathrm{A}})})$, where $\{\alpha_{\mathrm{D},k}^{(n_{\mathrm{D}})}, \alpha_{\mathrm{A},k}^{(n_{\mathrm{A}})}\}$ and $\{\theta_{\mathrm{D},k}^{(n_{\mathrm{D}})}, \theta_{\mathrm{A},k}^{(n_{\mathrm{A}})}\}$ are the complex channel gains and received path angles for the corresponding channels, respectively. $\mathbf{a}_{\mathrm{D}}(\theta)$ and $\mathbf{a}_{\mathrm{A}}(\theta)$ are $M \times 1$ and $L \times 1$ steering vectors of the path angles as $\mathbf{a}_{\mathrm{D}}(\theta) = \frac{1}{\sqrt{M}}[e^{j\omega_0}, \ldots, e^{j\omega_{M-1}}]^{\mathsf{T}}$, $\mathbf{a}_{\mathrm{A}}(\theta) = \frac{1}{\sqrt{L}}[e^{j\omega_0}, \ldots, e^{j\omega_{L-1}}]^{\mathsf{T}}$ where $\omega_n = n\frac{2\pi d}{\lambda}\pi \sin(\theta)$ and $d = \lambda/2$ is the array spacing for the wavelength λ. Further, the mm-Wave channel between the BS and the LIS is given by

$$\mathbf{H} = \sqrt{\frac{ML}{N_{\mathrm{H}}}} \sum_{n_{\mathrm{H}}=1}^{N_{\mathrm{H}}} \alpha^{(n_{\mathrm{H}})} \mathbf{a}_{\mathrm{BS}}(\theta_{\mathrm{BS}}^{(n_{\mathrm{H}})}) \mathbf{a}_{\mathrm{LIS}}^{\mathsf{H}}(\theta_{\mathrm{LIS}}^{(n_{\mathrm{H}})}), \tag{2}$$

where $\alpha^{(n_{\mathrm{H}})} \in \mathbb{C}$ denotes the complex gain and $\{\theta_{\mathrm{BS}}^{(n_{\mathrm{H}})}, \theta_{\mathrm{LIS}}^{(n_{\mathrm{H}})}\}$ are the angle-of-departure (AOD) and angle-of-arrival (AOA) angles of the paths, respectively. $\mathbf{a}_{\mathrm{BS}}(\theta) \in \mathbb{C}^M$ and $\mathbf{a}_{\mathrm{LIS}}(\theta) \in \mathbb{C}^L$ are the steering vectors. Let $\mathbf{G}_k \in \mathbb{C}^{M\times L}$ be the cascaded channel matrix between the BS and the k-th user as $\mathbf{G}_k = \mathbf{H}\boldsymbol{\Gamma}_k$ where $\boldsymbol{\Gamma}_k = \mathrm{diag}\{\mathbf{h}_{\mathrm{A},k}\}$. Then, we can write $\mathbf{H}\boldsymbol{\Psi}\mathbf{h}_{\mathrm{A},k} = \mathbf{G}_k\boldsymbol{\psi}$, for which we have $\boldsymbol{\Psi} = \mathrm{diag}\{\boldsymbol{\psi}\}$.

In this work, our aim is to estimate the direct and cascaded channels $\{\mathbf{h}_{\mathrm{D},k}, \mathbf{G}_k\}$ in downlink transmission. In this case, we assume that each user feeds the received pilot signals to the deep network (henceforth, called ChannelNet) to estimate its own channel.

III. Channel Estimation Via Deep Learning

The proposed DL framework uses the received pilot signals as input to estimate the direct and cascaded channels.

A. Labeling

Consider the downlink scenario where the BS transmits the orthogonal pilot signals $\mathbf{x}_p \in \mathbb{C}^M$, one at a single coherence time τ, with $p = 1, \ldots, P$ and $P \geq M$. Hence, the total number of channel uses to estimate the direct channel is P The received signal at the k-th user can be given by

$$\mathbf{y}_k = \big(\mathbf{h}_{\mathrm{D},k}^{\mathsf{H}} + \boldsymbol{\psi}^{\mathsf{H}}\mathbf{G}_k^{\mathsf{H}}\big)\mathbf{X} + \mathbf{n}_k, \tag{3}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{C}^{M\times P}$ is the pilot signal matrix while $\mathbf{y}_k = [y_{k,1}, \ldots, y_{k,P}]$ and $\mathbf{n}_k = [n_{k,1}, \ldots, n_{k,P}]$ are $1 \times P$ row vectors and $\mathbf{n}_k \sim \mathcal{CN}(0, \sigma_n^2\mathbf{I}_P)$. We assume that the pilot training has two phases: direct channel estimation (i.e., $\mathbf{h}_{\mathrm{D},k}$) and the cascaded channel estimation (i.e., $\mathbf{G}_k$). In phase I, we assume that all of the LIS elements are turned off, i.e., $\beta_l = 0, \forall l$, by using the BS backhaul link[1]. We note here that by setting β_l as $\{1, 0\}$ does not affect the the direct and cascaded channels since they do not depend on the reflect beamformer $\boldsymbol{\Psi}$ as seen in (3). Then, the received baseband signal at the k-th user becomes

$$\mathbf{y}_{\mathrm{D}}^{(k)} = \mathbf{h}_{\mathrm{D},k}^{\mathsf{H}}\mathbf{X} + \mathbf{n}_{\mathrm{D},k}. \tag{4}$$

Here, the direct channel $\mathbf{h}_{\mathrm{D},k}$ is selected as the label of the deep network with the corresponding input data of $\mathbf{y}_{\mathrm{D}}^{(k)}$.

Once $\mathbf{h}_{\mathrm{D},k}$, being the estimated channel, is obtained, in the second phase of the training stage, the cascaded channel $\mathbf{G}_k$ can be estimated. This can be achieved via two approaches. In the first approach, $P = M$ pilot signals are transmitted when each of the LIS elements is turned on one by one. In this case, the BS sends a request to LIS via the micro-controller device in the backhaul link to turn on a single LIS element at a time. For the l-th frame, the reflect beamforming vector becomes $\boldsymbol{\psi}^{(l)} = [0, \ldots, 0, \psi_l, 0, \ldots, 0]^{\mathsf{T}}$ where $\beta_{\bar{l}} = \{0 : \bar{l} = 1, \ldots, L, \bar{l} \neq l\}$ and the received signal from the cascaded channel at the k-th user becomes

$$\mathbf{y}_{\mathrm{C}}^{(k,l)} = \big(\mathbf{h}_{\mathrm{D},k}^{\mathsf{H}} + \mathbf{g}_{k,l}^{\mathsf{H}}\big)\mathbf{X} + \mathbf{n}_{k,l}, \tag{5}$$

[1]When the PIN diodes are turned off, the reflecting elements are almost transparent with the insertion loss nearly being zero such that the incoming signals pass through the reflecting elements [16].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2020.2993699, IEEE Wireless Communications Letters

where $\mathbf{y}_{\mathrm{C}}^{(k,l)} = [y_{\mathrm{C},1}^{(k,l)}, \ldots, y_{\mathrm{C},P}^{(k,l)}]$ and $\mathbf{n}_{k,l} = [n_{k,1}^{(l)}, \ldots, n_{k,P}^{(l)}]$ are $1 \times P$ row vectors. In (5), $\mathbf{g}_{k,l}$ represents the l-th column of $\mathbf{G}_k$ as $\mathbf{g}_{k,l} = \mathbf{G}_k \boldsymbol{\psi}^{(l)}$. Then the least-squares (LS) estimate of $\mathbf{g}_{k,l}$ becomes

$$\hat{\mathbf{g}}_{k,l} = \big(\mathbf{y}_{\mathrm{C}}^{(k,l)} \mathbf{X}^{\mathsf{H}} \big(\mathbf{X}\mathbf{X}^{\mathsf{H}}\big)^{-1}\big)^{\mathsf{H}} - \mathbf{h}_{\mathrm{D},k}. \tag{6}$$

By using $\hat{\mathbf{h}}_{\mathrm{D},k}$, (6) can be solved for $l = 1 \ldots, L$. Then, we can construct the estimated cascaded matrix as $\hat{\mathbf{G}}_k = [\hat{\mathbf{g}}_{k,1}, \ldots, \hat{\mathbf{g}}_{k,L}]$.

In the second approach, channel estimation is done when all LIS elements are turned on. In this case, the L columns of $\mathbf{G}_k$ are jointly estimated by using $\bar{\mathbf{X}} \in \mathbb{C}^{ML \times ML}$ pilot signal matrix. Let the reflect beamforming vector be an $L \times 1$ vector of all ones, i.e., $\bar{\boldsymbol{\psi}} = \mathbf{1}_L$, then we can write the $ML \times 1$ received signal as

$$\bar{\mathbf{y}}_{\mathrm{C}}^{(k)} = \big(\bar{\mathbf{h}}_{\mathrm{D},k}^{\mathsf{H}} + \bar{\mathbf{g}}_k^{\mathsf{H}}\big)\bar{\mathbf{X}} + \bar{\mathbf{n}}_k, \tag{7}$$

where $\bar{\mathbf{h}}_{\mathrm{D},k} = \mathbf{1}_L \otimes \mathbf{h}_{\mathrm{D},k}$ where $\otimes$ denotes the kronecker product and $\bar{\mathbf{g}}_k = [\mathbf{g}_{k,1}^{\mathsf{T}}, \ldots, \mathbf{g}_{k,L}^{\mathsf{T}}]^{\mathsf{T}}$. Then, the LS estimate of $\bar{\mathbf{g}}_k$ becomes

$$\hat{\bar{\mathbf{g}}}_k = \big(\bar{\mathbf{y}}_{\mathrm{C}}^{(k)} \bar{\mathbf{X}}^{\mathsf{H}} \big(\bar{\mathbf{X}}\bar{\mathbf{X}}^{\mathsf{H}}\big)^{-1}\big)^{\mathsf{H}} - \bar{\mathbf{h}}_{\mathrm{D},k}. \tag{8}$$

The estimated cascaded channel from (6) and (8) will yield the same results if perfectly orthogonal pilots are used. When the pilot signals become correlated/corrupted, then (6) provides better results since $\mathbf{X}$ involves less corruption than $\bar{\mathbf{X}}$ (Please see Fig. 4).

Algorithm 1 Training data generation for ChannelNet.

Input: K, U, V, $\mathbf{X}$, $\boldsymbol{\psi}$ SNR, $\mathrm{SNR}_{\mathbf{h}}$, $\mathrm{SNR}_{\mathbf{G}}$.
Output: Training datasets $\mathcal{D}_{\mathrm{DC}}$ and $\mathcal{D}_{\mathrm{CC}}$.
Initialize with $t = 1$ and the dataset length is $T = UVK$.
for $1 \le v \le V$ **do**
  Generate $\mathbf{h}_{\mathrm{D},k}^{(v)}$ and $\mathbf{G}_k^{(v)}$ from Section II , $\forall k$.
  for $1 \le u \le U$ **do**
  $[\mathbf{h}_{\mathrm{D},k}^{(u,v)}]_{i,j} \sim \mathcal{CN}([\mathbf{h}_{\mathrm{D},k}^{(v)}]_{i,j}, \sigma_{\mathbf{h}}^2)$, $\forall k$.
  $[\mathbf{G}_k^{(u,v)}]_{i,j} \sim \mathcal{CN}([\mathbf{G}_k^{(v)}]_{i,j}, \sigma_{\mathbf{G}}^2)$, $\forall k$.
  for $1 \le k \le K$ **do**
  Using $\mathbf{h}_{\mathrm{D},k}^{(u,v)}$ and $\mathbf{g}_{k,l}^{(u,v)}$, generate $\mathbf{y}_{\mathrm{D}}^{(k)^{(u,v)}}$ and $\mathbf{y}_{\mathrm{C}}^{(k,l)^{(u,v)}}$ from (4) an (5).
  Using $\mathbf{y}_{\mathrm{D}}^{(k)^{(u,v)}}$ and $\mathbf{y}_{\mathrm{C}}^{(k,l)^{(u,v)}}$; design $\mathbf{X}_{\mathrm{DC}}^{(t)}$ and $\mathbf{X}_{\mathrm{CC}}^{(t)}$.
  Using $\mathbf{h}_{\mathrm{D},k}^{(u,v)}$, $\mathbf{G}_k^{(u,v)}$; design the output $\mathbf{z}_{\mathrm{DC}}^{(t)}$, $\mathbf{z}_{\mathrm{CC}}^{(t)}$.
  $\mathcal{D}_{\mathrm{DC}}^{(t)} = (\mathbf{X}_{\mathrm{DC}}^{(t)}, \mathbf{z}_{\mathrm{DC}}^{(t)})$, $\mathcal{D}_{\mathrm{CC}}^{(t)} = (\mathbf{X}_{\mathrm{CC}}^{(t)}, \mathbf{z}_{\mathrm{CC}}^{(t)})$.
  $t \leftarrow t + 1$,
  end for k,
  end for u,
end for v,

B. Input Design: Received Pilots

The deep network accepts the received signals as input at the preamble stage. As a result, the input-output pairs become $\{\mathbf{y}_{\mathrm{D}}^{(k)}, \mathbf{h}_{\mathrm{D},k}\}$ and $\{\mathbf{y}_{\mathrm{C}}^{(k,l)}, \mathbf{g}_{k,l}\}$ for direct and cascaded channel estimation, respectively. In order to feed the deep network we use real, imaginary and the absolute value of each entry of the received signal. While the use of only real/imaginary components is still possible [10], it is shown in [17]–[20] that the use of "three-channel" data ameliorates the performance by enriching the features inherited in the input data. Let us define the input of the deep network as $\mathbf{X}_{\mathrm{DC}}$ and $\mathbf{X}_{\mathrm{CC}}$ for the direct and cascaded channel, respectively. Then, $\mathbf{X}_{\mathrm{DC}}$ is a $\tilde{M} \times \tilde{M} \times 3$ real-valued "three-channel" matrix, each of which is of size $\tilde{M} = \sqrt{M}$. In order to benefit from 2-D convolutional filters, we construct the matrix quantity $\mathbf{X}_{\mathrm{DC}}$ from the vector $\mathbf{y}_{\mathrm{D}}^{(k)}$ by partitioning $\mathbf{y}_{\mathrm{D},k}$ into $\sqrt{M}$ subvectors and put them into $\sqrt{M}$ columns[2]. In particular, for the first and the second "channels" of $\mathbf{X}_{\mathrm{DC}}$, we have $\mathrm{vec}\big\{[\mathbf{X}_{\mathrm{DC}}]_1\big\} = \mathrm{Re}\big\{\mathbf{y}_{\mathrm{D}}^{(k)}\big\}$ and $\mathrm{vec}\big\{[\mathbf{X}_{\mathrm{DC}}]_2\big\} = \mathrm{Im}\big\{\mathbf{y}_{\mathrm{D}}^{(k)}\big\}$. Finally, the third "channel" is denoted by the element-wise absolute value of $\mathbf{y}_{\mathrm{D}}^{(k)}$ as $\mathrm{vec}\big\{[\mathbf{X}_{\mathrm{DC}}]_3\big\} = \big|\mathbf{y}_{\mathrm{D}}^{(k)}\big|$. Similarly, we can define $\mathbf{X}_{\mathrm{CC}}$ as an $L \times M \times 3$ real-valued matrix and we have $\mathrm{vec}\big\{[\mathbf{X}_{\mathrm{CC}}]_1\big\} = \mathrm{Re}\big\{\tilde{\mathbf{y}}_k\big\}$, $\mathrm{vec}\big\{[\mathbf{X}_{\mathrm{CC}}]_2\big\} = \mathrm{Im}\big\{\tilde{\mathbf{y}}_k\big\}$ and $\mathrm{vec}\big\{[\mathbf{X}_{\mathrm{CC}}]_3\big\} = \big|\tilde{\mathbf{y}}_k\big|$ where $\tilde{\mathbf{y}}_k = [\mathbf{y}_{\mathrm{C}}^{(k,l)^{\mathsf{T}}}, \ldots, \mathbf{y}_{\mathrm{C}}^{(k,L)^{\mathsf{T}}}]^{\mathsf{T}}$ is an $LM \times 1$ vector composed of the received pilot signals. The input design for the second approach can also be done accordingly by using (7). The output of the deep network is the vectorized form of the channel matrices, i.e., $\mathbf{z}_{\mathrm{DC}} = \big[\mathrm{Re}\{\mathbf{h}_{\mathrm{D},k}\}^{\mathsf{T}}, \mathrm{Im}\{\mathbf{h}_{\mathrm{D},k}\}^{\mathsf{T}}\big]^{\mathsf{T}}$ and $\mathbf{z}_{\mathrm{CC}} = \big[\mathrm{Re}\{\mathrm{vec}\{\mathbf{G}_k\}\}^{\mathsf{T}}, \mathrm{Im}\{\mathrm{vec}\{\mathbf{G}_k\}\}^{\mathsf{T}}\big]^{\mathsf{T}}$ which are $2M \times 1$ and $2ML \times 1$ vectors, respectively. The training data can be obtained by generating the input-output pairs for several realizations, as described in Algorithm 1.

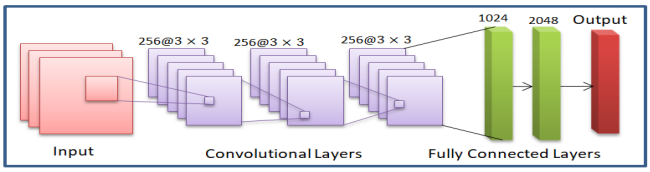

Fig. 2. Proposed deep neural network architecture.

C. Network Architectures and Training

ChannelNet composed of two identical CNNs, each of which is composed of 9 layers as illustrated in Fig. 2. The first layer is the input layer which accepts the received pilot signals. Since the same network is used for both direct and cascaded channels, the size of the input differs as described in Section III-B. The $\{2, 3, 4\}$-th layers are convolutional layers (CLs) with 256 filters of size 3×3. The fifth and the seventh layers are fully connected layers (FCLs) with 1024 and 2048 units respectively. There are dropout layers with a 50% probability after each FCL and the last layer is the regression layer. The network parameters are fixed after a hyperparameter tuning process that yields the best performance for the considered scenario [17]–[19]. The proposed deep network is realized and trained in MATLAB on a PC with a single GPU and a 768-core processor. We used the stochastic gradient descent algorithm with momentum 0.9 and updated the network parameters with learning rate 0.0002 and

[2] $\sqrt{M}$ is assumed to be an integer value. If not, a rectangular $\mathbf{X}_{\mathrm{DC}}$ can always be constructed without affecting the network training.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2020.2993699, IEEE Wireless Communications Letters

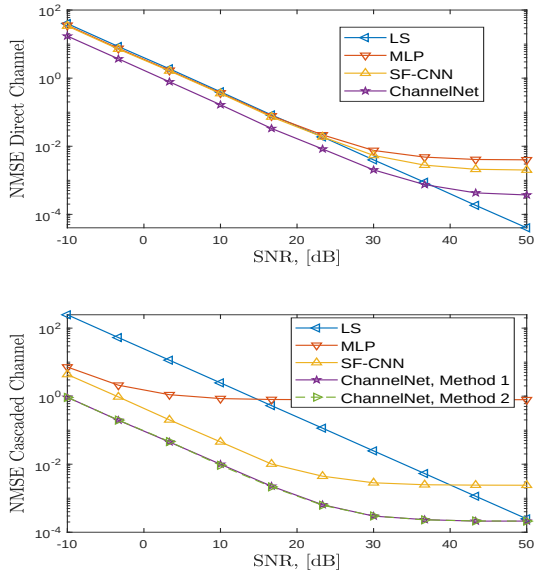

Fig. 3. Channel estimation NMSE with respect to SNR.

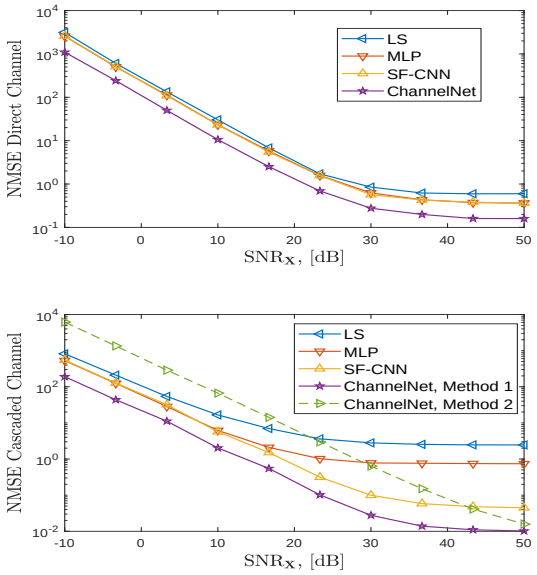

Fig. 4. Channel estimation NMSE with respect to SNR$_{\mathbf{X}}$.

mini-batch size of 128 samples. Then, we applied a stopping criterion in training which ceases training when the validation accuracy does not improve in three consecutive epochs.

IV. Numerical Simulations

In this part, we evaluate the performance of the proposed ChannelNet framework with comparison to the state-of-the-art DL-based approaches such as MLP [13] and SF-CNN [10]. Throughout the simulations, we select $P = M = 64$, $L = 100$, and $K = 8$. The physical environment is modeled with $N_{\mathrm{D}} = N_{\mathrm{A}} = N_{\mathrm{H}} = 10$ paths, where the direction of users are uniform randomly drawn from the interval $[-\pi, \pi]$ and we select $\epsilon_0 = \epsilon_1 = 0$ unless stated otherwise.

To train the network, we generate different channel scenarios for $U = 100$, $V = 500$ and $K = 8$. During training, three signal-to-noise ratio (SNR) levels are used to improve the robustness, i.e., SNR= $\{10, 20, 30\}$ dB. In addition, synthetic noise is added to the labels with SNR$_{\mathbf{h}}$ =SNR$_{\mathbf{G}}$ = $\{20, 30\}$ dB, where SNR$_{\mathbf{h}} = 20\log_{10}(\frac{|[\mathbf{h}]_i|^2}{\sigma^2_{\mathbf{h}}})$ and SNR$_{\mathbf{G}} = 20\log_{10}(\frac{|[\mathbf{G}]_{i,j}|^2}{\sigma^2_{\mathbf{G}}})$, respectively. Hence, the total data length is $T = 240000$. During training 70% and 30% of the whole generated data are used for training and validation respectively. The training stage takes about 40 minutes, whereas the online deployment of the proposed DL network only needs 0.004 seconds. Once the training is completed, a new received pilot data other than the training data is generated and used in the prediction stage, where $J = 100$ Monte Carlo experiments are conducted to assess the normalized mean-square-error (NMSE) performance i.e., the NMSE for $\mathbf{G}_k$ is defined as $\frac{1}{J}\sum_{j=1}^{J}||\mathbf{G}_k - \hat{\mathbf{G}}_k^{(j)}||_{\mathcal{F}}/||\mathbf{G}_k||_{\mathcal{F}}$.

In Fig. 3, we present the channel estimation for direct and cascaded channels with respect to SNR. We can see that the DL-based approaches have better NMSE than the LS [7] due to their better mapping architectures from the received pilots to channel data. Among the DL-based techniques, ChannelNet has superior performance as compared to the others. The effectiveness of ChannelNet is due to the joint use of CL and FCLs, whereas MLP and SF-CNN have FCL-only and CL-only structures, respectively. While FCLs are powerful in constructing non-linear mapping between input and the output, CLs play very important role in DL networks when generating new features to enrich the mapping performance. We also observe that the performance of the DL-based approaches saturates at high SNR (i.e., $>$ 20 dB) because of the biased nature of the neural networks which do not provide unlimited accuracy. This problem can be mitigated by increasing the number of units in various network layers. Unfortunately, it may lead to network memorizing the training data and perform poorly when the test data are different than the ones in training. To balance this trade-off, we have used noisy data-sets during training so that the network attains reasonable tolerance to corrupted/imperfect inputs.

In Fig. 4, the effect of corrupted pilot data is examined and the performance of the algorithms is obtained with respect to SNR on the pilot data, i.e., SNR$_{\mathbf{X}} = 20\log_{10}(\frac{|[\mathbf{X}]_{i,j}|^2}{\sigma^2_{\mathbf{X}}})$ when SNR= 10 dB. We observe that all of the algorithms require at least SNR$_{\mathbf{X}} \geq$ 20 dB to provide a reasonable NMSE performance and the proposed DL approach has the superior performance among all.We see that cascaded channel estimation with the first method is more robust to pilot corruption than the second method due the use of fewer pilot signals.

The adaptation of DL-based techniques when the channel data change is an important performance measure. In Fig. 5, the NMSE is presented when the AOAs of the users in the test stage are different for training. In the test stage, we introduce an angle mismatch into the AOA of all users with standard deviation σ_θ and present the results in Fig. 5. We can see that ChannelNet outperforms the other algorithms and it provides satisfactory performance up to 4° angular mismatch in the test data.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2020.2993699, IEEE Wireless Communications Letters

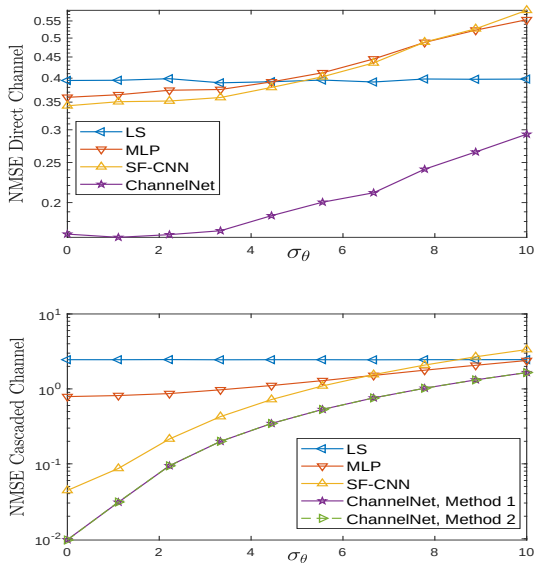

Fig. 5. Channel estimation NMSE with respect to σ_θ.

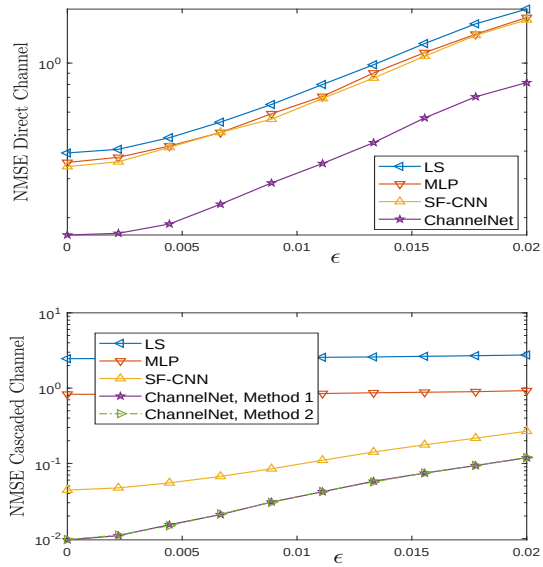

Fig. 6. Channel estimation NMSE with respect to ϵ.

In Fig.6, we present the performance with respect to non-ideal switching of LIS elements for $\epsilon = \epsilon_0 = \epsilon_1$. As it is seen, $\epsilon \leq 5 \times 10^{-3}$ provides satisfactory NMSE performance.

V. Summary

We proposed a DL-based channel estimation technique for LIS-assisted massive MIMO systems. In the proposed scheme, each user has an identical deep network which is fed by the received pilot signals to effectively estimate the direct and the cascaded channels. We have conducted several experiments to investigate the performance of the algorithms and observed that the proposed approach outperforms the other algorithms. We have shown that the proposed approach does not need to be re-trained when the user locations change up to 4 degrees. We have also investigated the non-ideal switching scenario for the LIS elements and shown that the proposed method can provide reasonable performance up to 0.5% amplitude error in switching.

References

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[2] C. Pan, H. Ren, K. Wang, M. Elkashlan, A. Nallanathan, J. Wang, and L. Hanzo, "Intelligent reflecting surface enhanced mimo broadcasting for simultaneous wireless information and power transfer," *arXiv preprint arXiv:1908.04863*, 2019.

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

[4] Q. Wu and R. Zhang, "Towards Smart and Reconfigurable Environment: Intelligent Reflecting Surface Aided Wireless Network," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 106–112, January 2020.

[5] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable Intelligent Surfaces for Energy Efficiency in Wireless Communication," *IEEE Trans. Wireless Commun.*, vol. 18, pp. 4157–4170, 2018.

[6] W. Tang, J. Y. Dai, M. Z. Chen, K.-K. Wong, X. Li, X. Zhao, S. Jin, Q. Cheng, and T. J. Cui, "MIMO Transmission through Reconfigurable Intelligent Surface: System Design, Analysis, and Implementation," *arXiv preprint arXiv:1912.09955*, 2019.

[7] B. Zheng and R. Zhang, "Intelligent Reflecting Surface-Enhanced OFDM: Channel Estimation and Reflection Optimization," *IEEE Wireless Commun. Lett.*, pp. 1–1, 2019.

[8] Z. He and X. Yuan, "Cascaded channel estimation for large intelligent metasurface assisted massive mimo," *IEEE Wireless Commun. Lett.*, vol. 9, no. 2, pp. 210–214, Feb 2020.

[9] J. Lin, G. Wang, R. Fan, T. A. Tsiftsis, and C. Tellambura, "Channel estimation for wireless communication systems assisted by large intelligent surfaces," *arXiv preprint arXiv:1911.02158*, 2019.

[10] P. Dong, H. Zhang, G. Y. Li, I. S. Gaspar, and N. NaderiAlizadeh, "Deep CNN-Based Channel Estimation for mmWave Massive MIMO Systems," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 989–1000, Sep. 2019.

[11] A. Taha, M. Alrabeiah, and A. Alkhateeb, "Enabling large intelligent surfaces with compressive sensing and deep learning," *arXiv preprint arXiv:1904.10136*, 2019.

[12] S. Khan and S. Y. Shin, "Deep-learning-aided detection for reconfigurable intelligent surfaces," *arXiv preprint arXiv:1910.09136*, 2019.

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

[14] Z. Wang, L. Liu, and S. Cui, "Channel estimation for intelligent reflecting surface assisted multiuser communications," *arXiv preprint arXiv:1911.03084*, 2019.

[15] D. Mishra and H. Johansson, "Channel estimation and low-complexity beamforming design for passive intelligent surface assisted miso wireless energy transfer," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 4659–4663.

[16] Kihun Chang, Sang il Kwak, and Young Joong Yoon, "Equivalent circuit modeling of active frequency selective surfaces," in *2008 IEEE Radio and Wireless Symposium*, Jan 2008, pp. 663–666.

[17] A. M. Elbir, K. V. Mishra, and Y. C. Eldar, "Cognitive radar antenna selection via deep learning," *IET Radar, Sonar & Navigation*, vol. 13, pp. 871–880, 2019.

[18] A. M. Elbir, "CNN-based precoder and combiner design in mmWave MIMO systems," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1240–1243, 2019.

[19] A. M. Elbir and K. V. Mishra, "Joint antenna selection and hybrid beamformer design using unquantized and quantized deep learning networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1677–1688, March 2020.

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