FEDERATED CHANNEL LEARNING FOR INTELLIGENT REFLECTING SURFACES WITH FEWER PILOT SIGNALS

Ahmet M. Elbir[†]*, Sinem Coleri*[+] *and Kumar Vijay Mishra*[‡]

[†]Department of Electrical and Electronics Engineering, Duzce University, Duzce, Turkey
[+]Department of Electrical and Electronics Engineering, Koç University, Istanbul, Turkey
[‡]United States CCDC Army Research Laboratory, Adelphi, MD 20783 USA

ABSTRACT

Channel estimation is a critical task in intelligent reflecting surface (IRS)-assisted wireless systems due to the uncertainties imposed by environment dynamics and rapid changes in the IRS configuration. To deal with these uncertainties, deep learning (DL) approaches have been proposed. Previous works consider centralized learning (CL) approach for model training, which entails the collection of the whole training dataset from the users at the base station (BS), hence introducing huge transmission overhead for data collection. To address this challenge, this paper proposes a federated learning (FL) framework to jointly estimate both direct and cascaded channels in IRS-assisted wireless systems. We design a single convolutional neural network trained on the local datasets of the users without sending them to the BS. We show that the proposed FL-based channel estimation approach requires approximately 60% fewer pilot signals and it exhibits 12 times lower transmission overhead than CL, while maintaining satisfactory performance close to CL. In addition, it provides lower estimation error than the state-of-the-art DL-based schemes.

Index Terms— Channel estimation, federated learning, intelligent reflecting surfaces, machine learning, massive MIMO.

1. INTRODUCTION

Intelligent reflecting surfaces (IRSs) have been attracted a great research interest due to the lower hardware complexity, physical size and cost of conventional large arrays [1]. An IRS is an electromagnetic two-dimensional surface that is composed of large number of passive reconfigurable meta-material elements, which reflect the incoming signal by introducing a pre-determined phase shift. This phase shift is controlled via external signals by the base station (BS) through a backhaul control link. As a result, the incoming signal from the BS can be manipulated in real-time, thereby, reflecting the received signal toward the users. Hence, the usage of IRS enhances the signal energy received by distant users and expands the coverage of the BS [2].

The system configuration of the IRS-assisted systems, i.e., designing reflect beamformers, strongly relies on the knowledge of the channel information. In fact, the IRS-assisted systems include multiple communications links, i.e., a direct channel from BS to users and a cascaded channel from BS to users through IRS. This makes the IRS scenario even more challenging than the conventional massive MIMO (multiple-input multiple-output) systems. Furthermore, the wireless channel is dynamic and uncertain because of changing IRS configurations. Consequently, there exists an inherit uncertainty stemming from the IRS configuration and the channel dynamics. These characteristics of IRS make the system design very challenging [3, 4].

To address the aforementioned uncertainties and non-linearities imposed by channel equalization, hardware impairments, and sub-optimality of high-dimensional problems, model-free techniques have become common in wireless communications [3]. In this context, deep learning (DL) is particularly powerful in extracting the features from the raw data by constructing a model-free data mapping. DL-based channel estimation for IRS-assisted wireless systems is considered in [4, 5]. Specifically, [5] devise a twin convolutional neural network (CNN), which takes the least squares (LS) estimates of the direct and cascaded channels as input to estimate the channels. A deep denoising neural network (DDNN) approach is proposed in [4], where the input of the DDNN is a compressive channel estimate that is obtained via orthogonal matching pursuit (OMP) algorithm from the receive antenna array measurements. These DL techniques employ a centralized learning (CL) scheme, wherein the learning model is trained at a parameter server (PS) (possible at the BS) by collecting the training data from the users. The transmission of the training datasets from the users to the PS introduces huge overhead due to the large size of the datasets. This overhead can be reduced by decentralized techniques, such as federated learning (FL), by bringing the learning task to the edge level, e.g., the mobile users [6–8]. In FL, instead of transmitting the whole dataset to the PS, each user computes the local model updates on its local dataset and only transmits the model updates (gradients) to the PS. Due to its recent success on wireless sensor networks [**?**, 9], UAV (unmanned aerial vehicle) networks [10] and vehicular networks [11], FL has been regarded as a promising tool to exhibit a communication-efficient as well as privacy-preserving learning scheme since it does not involve raw data transmission. Recently, FL has been applied for beamformer design problems in massive MIMO [12] and IRS-assisted systems [13]. In [13], FL is applied for channel estimation in IRS-assisted systems, where the proposed approach requires pilot signals at least the number of BS antennas. The present work eliminates such requirement and provides satisfactory channel estimation performance close to CL.

In this paper, we propose an FL approach for channel estimation in IRS-assisted wireless systems. We design a single CNN to jointly estimate the direct and cascaded channels in the presence of insufficient pilot signals. During model training, each user computes the model updates by using the local datasets (i.e., input of received pilots and label of channel data) and transmits them to the BS where

S. C. acknowledges the support of Ford Otosan and the Scientific and Technological Research Council of Turkey EU CHIST-ERA grant 119E350.

the model updates are aggregated and sent back to the users. The proposed FL approach is advantageous since it provides decentralized learning, which significantly reduces the transmission overhead compared to the CL-based techniques while maintaining satisfactory channel estimation performance close to CL. Furthermore, the performance of the proposed method can be attributed to the fact that the designed learning model constructs a non-linear mapping from the insufficient number of received pilots to the channel matrix entries. Therefore, the proposed approach is also effective since it does not involve a matrix inversion operation, which may be required in model-based techniques in massive MIMO applications. To the best of our knowledge, this is the first FL-based method for channel estimation with fewer pilot signals.

2. SIGNAL MODEL

We consider the downlink channel estimation for IRS-assisted massive MIMO systems, where the BS has M antennas to serve K single-antenna users with the assistance of IRS, which is composed of L reflective elements. The incoming signal from the BS is reflected from the IRS, where each IRS element introduces a phase shift φ_l, for $l = 1, \ldots, L$. This phase shift can be adjusted through the PIN (positive-intrinsic-negative) diodes, which are controlled by the IRS-controller connected to the BS over the backhaul link [2,14]. As a result, IRS allows the users receive the signal transmitted from the BS when they are distant from the BS or there is a blockage among them. Assume that the BS transmits the k-th user the data symbol $s_k \in \mathbb{C}$ 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 $\overline{\mathbf{s}} = \sum_{k=1}^{K} \sqrt{\gamma_k}\bar{\mathbf{f}}_k s_k$, 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-th user with two components, one of which is through the direct path from the BS and another one is through the IRS. The received signal at the k-th user can be given by

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

where $n_k \in \mathbb{C}$ denotes the spatially and temporarily white Gaussian noise and $\boldsymbol{\Psi} = \mathrm{diag}\{\boldsymbol{\psi}\} \in \mathbb{C}^{L\times L}$ and $\boldsymbol{\psi} = [\psi_1, \ldots, \psi_L]^{\mathsf{T}} \in \mathbb{C}^L$ is the reflecting beamformer vector, whose l-th entry is $\psi_l = a_n e^{j\varphi_l}$, where $a_l \in \{0, 1\}$ denotes the on/off stage of the l-th element of the IRS and $\varphi_l \in [0, 2\pi]$ is the phase shift introduced by the IRS. In practice, the IRS elements cannot be perfectly turned on/off, hence, they can be modeled as $a_n = \begin{cases} 1-\epsilon_1 & \text{On} \\ 0+\epsilon_0 & \text{Off} \end{cases}$, for small $\epsilon_1, \epsilon_0 \geq 0$, which represents the insertion loss of the reflecting elements [5, 15]. $\mathbf{H} \in \mathbb{C}^{M\times L}$ is the channel between the BS and the IRS and it can be defined as $\mathbf{H} = \sqrt{\frac{ML}{N_{\mathrm{r}}}}\sum_{n=1}^{N_{\mathrm{r}}} \alpha_n \mathbf{a}_{\mathrm{BS}}(\phi_n^{\mathrm{BS}})\mathbf{a}_{\mathrm{IRS}}(\phi_n^{\mathrm{IRS}})^{\mathsf{H}}$, where N_{r} and α_l^{IRS} are the number of received paths and the complex gain respectively. $\mathbf{a}_{\mathrm{BS}}(\phi_n^{\mathrm{BS}}) \in \mathbb{C}^M$ and $\mathbf{a}_{\mathrm{IRS}}(\phi_n^{\mathrm{IRS}}) \in \mathbb{C}^L$ are the steering vectors corresponding to the BS and IRS with the angle-of-arrival/departure (AoA/AoD) angles $\phi_n^{\mathrm{BS}}, \phi_n^{\mathrm{IRS}}$, respectively. $\mathbf{h}_{\mathrm{BS},k} \in \mathbb{C}^M$ and $\mathbf{h}_{\mathrm{IRS},k} \in \mathbb{C}^L$ represent the direct channels for BS-user and IRS-user links, and they can be defined respectively as

$$\mathbf{h}_{\mathrm{BS},k} = \sqrt{\frac{M}{N_{\mathrm{r}}^{\mathrm{BS}}}}\sum_{n=1}^{N_{\mathrm{r}}^{\mathrm{BS}}} \alpha_{k,n}^{\mathrm{BS}}\mathbf{a}_{\mathrm{BS}}(\varphi_{k,n}^{\mathrm{BS}}), \tag{2}$$

and

$$\mathbf{h}_{\mathrm{IRS},k} = \sqrt{\frac{L}{N_{\mathrm{r}}^{\mathrm{IRS}}}}\sum_{l=1}^{N_{\mathrm{IRS}}} \alpha_{k,n}^{\mathrm{IRS}}\mathbf{a}_{\mathrm{IRS}}(\varphi_{k,n}^{\mathrm{IRS}}). \tag{3}$$

Here, $N_{\mathrm{r}}^{\mathrm{BS}}$, $\alpha_{k,n}^{\mathrm{BS}}$ and $\mathbf{a}_{\mathrm{BS}}(\varphi_{k,n}^{\mathrm{BS}})$ ($N_{\mathrm{r}}^{\mathrm{IRS}}$, $\alpha_{k,n}^{\mathrm{IRS}}$, $\mathbf{a}_{\mathrm{IRS}}(\varphi_{k,n}^{\mathrm{IRS}})$) are the number of paths, complex gain and the steering vector with the AoA angle $\varphi_{k,n}^{\mathrm{BS}}$ ($\varphi_{k,n}^{\mathrm{IRS}}$) for the BS-user (IRS-user) communication link, respectively.

Let $\boldsymbol{\Lambda}_k = \mathrm{diag}\{\mathbf{h}_{\mathrm{IRS},k}\} \in \mathbb{C}^{L\times L}$, then the cascaded channel between the BS and the k-th user becomes $\mathbf{G}_k = \mathbf{H}\boldsymbol{\Lambda}_k \in \mathbb{C}^{M\times L}$ and (1) can be rewritten as

$$y_k = \big(\mathbf{h}_{\mathrm{BS},k}^{\mathsf{H}} + \boldsymbol{\psi}^{\mathsf{H}}\mathbf{G}_k^{\mathsf{H}}\big)\overline{\mathbf{s}} + n_k. \tag{4}$$

Our aim in this work is to estimate the direct and cascaded channels via FL. In the following, we first introduce how input and output data are designed for the proposed learning model, then discuss how FL-based training is performed.

3. CHANNEL ACQUISITION WITH FEWER PILOTS IN IRS-ASSISTED WIRELESS SYSTEMS

The proposed channel estimation scheme has two stages to separately collect the received data for direct and cascaded channels, $\mathbf{h}_{\mathrm{BS},k}$ and $\mathbf{G}_k$, respectively. In order to estimate the channels $\{\mathbf{h}_{\mathrm{BS},k}, \mathbf{G}_k\}$, $k = 1, \ldots, K$, conventional model-based [2, 14] and model-free [5, 13, 16] techniques require at least $\bar{M} \geq M$ orthogonal pilot signals, which can be represented as $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_M] \in \mathbb{C}^{M\times M}$ whose columns are orthogonal to each other, i.e., $||\mathbf{s}_{m_1}^{\mathsf{H}}\mathbf{s}_{m_2}||_2^2 = 0$ for $m_1 \neq m_2$, $m_1, m_2 = 1, \ldots, M$. In the presence of insufficient pilot signals, i.e., $\bar{M} < M$, using the incomplete pilot matrix $\bar{\mathbf{S}} \in \mathbb{C}^{M\times\bar{M}}$, the received signal at the k-th user becomes

$$\mathbf{y}_k = (\mathbf{h}_{\mathrm{BS},k}^{\mathsf{H}} + \boldsymbol{\psi}^{\mathsf{H}}\mathbf{G}_k^{\mathsf{H}})\overline{\mathbf{S}} + \mathbf{n}_k, \tag{5}$$

where $\mathbf{y}_k = [y_{1,k}, \ldots, y_{\bar{M},k}]$ and $\mathbf{n}_k = [n_{1,k}, \ldots, n_{\bar{M},k}]$ are $1\times\bar{M}$ row vectors. We intend to collect the received data with insufficient number of pilots, then construct a learning model which maps the received data to the channel data.

In the first stage, the received data with respect to $\mathbf{h}_{\mathrm{BS},k}$ can be collected when all of the IRS elements are turned off, i.e., $a_l = 0$, for $l = 1, \ldots, L$. Then, the $1 \times \bar{M}$ received signal at the k-th user is

$$\mathbf{y}_{\mathrm{D},k} = \mathbf{h}_{\mathrm{BS},k}^{\mathsf{H}}\overline{\mathbf{S}} + \mathbf{n}_{\mathrm{BS},k}. \tag{6}$$

The conventional model-based methods, such as least squares (LS) and minimum mean-squared-error (MMSE), fail to obtain $\mathbf{h}_{\mathrm{BS},k}$ from $\mathbf{y}_{\mathrm{BS},k}$ due to the lack of sufficient pilot data. Here, we leverage DL to estimate $\mathbf{h}_{\mathrm{BS},k}$ by selecting $\mathbf{y}_{\mathrm{B},k}$ as input to the CNN.

Next, we consider the cascaded channel estimation. We assume that each IRS element is turned on one by one while all the other elements are turned off. This is done by the BS requesting the IRS via a micro-controller device in the backhaul link so that a single IRS element is turned on at a time. Then, the reflecting beamformer vector at the l-th frame becomes $\boldsymbol{\psi}^{(l)} = [0, \ldots, 0, \psi_l, 0, \ldots, 0]^{\mathsf{T}}$, where $a_l = \{0 : \tilde{l} = 1, \ldots L, \tilde{l} \neq l\}$ and the received signal becomes

$$\mathbf{y}_{\mathrm{C},k}^{(l)} = (\mathbf{h}_{\mathrm{BS},k}^{\mathsf{H}} + \mathbf{g}_k^{(l)^{\mathsf{H}}})\overline{\mathbf{S}} + \mathbf{n}_k, \tag{7}$$

where $\mathbf{g}_k^{(l)} \in \mathbb{C}^M$ is the l-th column of $\mathbf{G}_k$, i.e., $\mathbf{g}_k^{(l)} = \mathbf{G}_k\boldsymbol{\psi}^{(l)}$, where $\psi_l = 1$. Using the estimate of $\mathbf{h}_{\mathrm{BS},k}$ from (6), (7) can be solved for $\mathbf{g}_k^{(l)}$, $l = 1, \ldots, L$, and the cascaded channel $\mathbf{G}_k$ can be estimated. Then, the received data for $l = 1 \ldots, L$ can be collected as

$$\mathbf{Y}_{\mathrm{C},k} = \left[\mathbf{y}_{\mathrm{C},k}^{(1)^{\mathsf{T}}}, \ldots, \mathbf{y}_{\mathrm{C},k}^{(L)^{\mathsf{T}}}\right]^{\mathsf{T}} \in \mathbb{C}^{L\times\bar{M}}. \tag{8}$$

In order to train the CNN for IRS-assisted massive MIMO scenario, we select the input-output data pair as $\{\mathbf{y}_{\mathrm{D},k}, \mathbf{h}_{\mathrm{BS},k}\}$ and $\{\mathbf{Y}_{\mathrm{C},k}, \mathbf{G}_k\}$ for direct and cascaded channels, respectively. To jointly learn both channels, a single input is constructed to train a single CNN as $\mathbf{\Upsilon}_k = \begin{bmatrix} \mathbf{y}_{\mathrm{D},k} \\ \mathbf{Y}_{\mathrm{C},k} \end{bmatrix} \in \mathbb{C}^{(L+1)\times \bar{M}}$. We design the input of the CNN as three *channel* of the input data, i.e., $[\mathcal{X}_k]_1 = \mathrm{Re}\{\mathbf{\Upsilon}_k\}$ and $[\mathcal{X}_k]_2 = \mathrm{Im}\{\mathbf{\Upsilon}_k\}$, $[\mathcal{X}_k]_3 = \angle\{\mathbf{\Upsilon}_k\}$, respectively. We can define the output data as $\mathbf{\Sigma}_k = [\mathbf{h}_{\mathrm{BS},k}, \mathbf{G}_k] \in \mathbb{C}^{\bar{M}\times(L+1)}$, hence, the output label can be given by a $2M(L+1)\times 1$ real-valued vector as

$$\mathcal{Y}_k = \left[\mathrm{vec}\{\mathrm{Re}\{\mathbf{\Sigma}_k\}\}^{\mathsf{T}}, \mathrm{vec}\{\mathrm{Im}\{\mathbf{\Sigma}_k\}\}^{\mathsf{T}}\right]^{\mathsf{T}}. \tag{9}$$

Consequently, the sizes of $\mathcal{X}_k$ and $\mathcal{Y}_k$ are $(L+1)\times\bar{M}\times 3$ and $2M(L+1)\times 1$, respectively. Once the CNN is trained, the received signals $\mathbf{y}_{\mathrm{D},k}$ and $\mathbf{Y}_{\mathrm{C},k}$ can be collected and fed to the CNN to estimate the channels. We discuss the FL-based model training in the following section.

4. FL-BASED MODEL TRAINING

Let $\mathcal{D}_k$ be the local dataset of the k-user, in which the i-th element is given by $\mathcal{D}_i = (\mathcal{X}_k^{(i)}, \mathcal{Y}_k^{(i)})$, where $\mathcal{X}_k^{(i)}$ and $\mathcal{Y}_k^{(i)}$ denote the input and output for $i = 1, \ldots, \mathsf{D}_k$ and $\mathsf{D}_k = |\mathcal{D}_k|$ ($\mathsf{D} = \sum_k \mathsf{D}_k$) is the size of the local dataset. We begin by introducing the training concept in conventional CL-based training, then develop FL-based model training.

In CL-based model training for channel estimation [5, 16, 17], the training of the global CNN is performed by collecting the local datasets $\mathcal{D}_k, k = 1, \ldots, K$ from the users. Once the BS has collected the whole dataset $\mathcal{D} = \bigcup_k \mathcal{D}_k$, the training is performed by solving the following problem

$$\underset{\boldsymbol{\theta}}{\mathrm{minimize}}\, \mathcal{F}(\boldsymbol{\theta}) = \frac{1}{\mathsf{D}} \sum_{i=1}^{\mathsf{D}} \mathcal{L}(f(\mathcal{X}^{(i)}|\boldsymbol{\theta}), \mathcal{Y}^{(i)}), \tag{10}$$

where $\boldsymbol{\theta} \in \mathbb{R}^P$ denotes the learnable parameters and $\mathcal{L}(\cdot)$ is the loss function defined by the learning model as

$$\mathcal{L}(f(\mathcal{X}^{(i)}|\boldsymbol{\theta}), \mathcal{Y}^{(i)}) = \|f(\mathcal{X}^{(i)}|\boldsymbol{\theta}) - \mathcal{Y}^{(i)}\|_{\mathcal{F}}^2, \tag{11}$$

which is the MSE between the label data $\mathcal{Y}^{(i)}$ and the prediction of the CNN, $f(\mathcal{X}^{(i)}|\boldsymbol{\theta})$ for the whole dataset, i.e., $i = 1, \ldots, \mathsf{D}$. The minimization of the empirical loss $\mathcal{F}(\boldsymbol{\theta})$ is achieved via gradient descent (GD) by updating the model parameters $\boldsymbol{\theta}_t$ at iteration t as $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \mathbf{g}(\boldsymbol{\theta}_t)$, where η_t is the learning rate and

$$\mathbf{g}(\boldsymbol{\theta}_t) = \nabla_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{\theta}_t) = \frac{1}{\mathsf{D}} \sum_{i=1}^{\mathsf{D}} \nabla_{\boldsymbol{\theta}} \mathcal{L}(f(\mathcal{X}^{(i)}|\boldsymbol{\theta}_t), \mathcal{Y}^{(i)}), \tag{12}$$

denotes the *full* or *batch* gradient vector in $\mathbb{R}^P$ [18, 19]. For large datasets, it is computationally inefficient to implement GD, which motivates the use of stochastic GD (SGD), where $\mathcal{D}$ is partitioned into M_B mini-batches as $\mathcal{D} = \bigcup_{m\in\mathcal{M}_B} \mathcal{D}_m$, for $\mathcal{M}_B = \{1, \ldots, M_B\}$. Then, $\boldsymbol{\theta}_t$ is updated by $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \mathbf{g}_{\mathcal{M}_B}(\boldsymbol{\theta}_t)$, where $\mathbf{g}_{\mathcal{M}_B}(\boldsymbol{\theta}_t) = \frac{1}{M_B}\sum_{m=1}^{M_B} \mathbf{g}_m(\boldsymbol{\theta}_t)$ includes the contribution of gradients computed over $\{\mathcal{D}_m\}_{m\in\mathcal{M}_B}$ as $\mathbf{g}_m(\boldsymbol{\theta}_t) = \frac{1}{\mathsf{D}_m}\sum_{i=1}^{\mathsf{D}_m} \nabla_{\boldsymbol{\theta}} \mathcal{L}(f(\mathcal{X}_m^{(i)}|\boldsymbol{\theta}_t), \mathcal{Y}_m^{(i)})$, where $\mathsf{D}_m = |\mathcal{D}_m|$ is the mini-batch size and $\mathbf{g}_{\mathcal{M}_B}(\boldsymbol{\theta}_t)$ satisfies $\mathbb{E}\{\mathbf{g}_{\mathcal{M}_B}(\boldsymbol{\theta}_t)\} = \nabla_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{\theta}_t)$. Therefore, SDG provides the minimization of the empirical loss by partitioning the dataset into M_B mini-batches and accelerates the learning process, which is known as mini-batch learning.

In FL, the training dataset $\mathcal{D}$ is partitioned into small portions, i.e., $\mathcal{D}_k$, $k = 1, \ldots, K$, however they are available at the users and not transmitted to the BS. Hence, given initial model $\boldsymbol{\theta}_{t-1}$, the k-th user aims to solve the following local problem at iteration t, i.e.,

$$\underset{\boldsymbol{\theta}}{\mathrm{minimize}}\, \frac{1}{\mathsf{D}_k} \sum_{i=1}^{\mathsf{D}_k} \mathcal{L}(f(\mathcal{X}_k^{(i)}|\boldsymbol{\theta}_{t-1}), \mathcal{Y}_k^{(i)}), \tag{13}$$

with the use of the local gradient $\mathbf{g}_k(\boldsymbol{\theta}_t)$. The k-th user transmits $\mathbf{g}_k(\boldsymbol{\theta}_t)$ to the BS, which receives $\tilde{\mathbf{g}}_k(\boldsymbol{\theta}_t) = \mathbf{g}_k(\boldsymbol{\theta}_t) + \mathbf{e}_{k,t}$, where $\mathbf{e}_{k,t} \in \mathbb{R}^P$ denotes the noise term added onto $\mathbf{g}_k(\boldsymbol{\theta}_t)$ at the t-th iteration. Without loss of generality, we assume that $\mathbf{e}_{k,t}$ obeys normal distribution, i.e., $\mathbf{e}_{k,t} \sim \mathcal{N}(0, \sigma_{\boldsymbol{\theta}}^2 \mathbf{I}_P)$, for $\mathbf{I}_P$ being $P \times P$ identity matrix with variance $\sigma_{\boldsymbol{\theta}}^2$ and the signal-to-noise-ratio (SNR) in gradient transmission is given by $\mathrm{SNR}_{\boldsymbol{\theta}} = 20\log_{10}\frac{||\mathbf{g}_k(\boldsymbol{\theta}_t)||_2^2}{\sigma_{\boldsymbol{\theta}}^2}$. Once the gradient data from all users are collected, the BS finally incorporates $\tilde{\mathbf{g}}_k(\boldsymbol{\theta}_t)$ for $k = 1, \ldots, K$ to update $\boldsymbol{\theta}_t$ as $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \frac{1}{K}\sum_{k=1}^{K} \tilde{\mathbf{g}}_k(\boldsymbol{\theta}_t)$, where $\tilde{\mathbf{g}}_k(\boldsymbol{\theta}_t) = \frac{1}{\mathsf{D}_k}\sum_{i=1}^{\mathsf{D}_k} \nabla_{\boldsymbol{\theta}} \mathcal{L}(f(\mathcal{X}_k^{(i)}|\boldsymbol{\theta}_t), \mathcal{Y}_k^{(i)}))$. After model aggregation, the BS returns the updated model parameters $\boldsymbol{\theta}_{t+1}$ to the users, and the received model parameter at the k-th user can be written as $\tilde{\boldsymbol{\theta}}_{t+1} = \boldsymbol{\theta}_{t+1} + \mathbf{e}_{k,t+1}$, which will be used for the computation of the gradients in the next iteration.

4.1. Neural Network Architecture

We design a single CNN trained on the local datasets of the users via FL. The proposed network architecture is comprised of 10 layers. The first layer is the input layer, which accepts the input data of size $(L+1)\times\bar{M}\times 3$. The $\{2, 4, 6\}$-th layers are the convolutional layers with $N_{\mathrm{SF}} = 128$ filters, each of which employs a 3×3 kernel for 2-D spatial feature extraction. The $\{3, 5, 7\}$-th layers are the normalization layers. The eighth layer is a fully connected layer with $N_{\mathrm{FCL}} = 1024$ units, whose main purpose is to provide feature mapping. The ninth layer is a dropout layer with $\kappa = 1/2$ probability. The dropout layer applies an $N_{\mathrm{FCL}} \times 1$ mask on the weights of the fully connected layer, whose elements are uniform randomly selected from $\{0, 1\}$. As a result, at each iteration of FL training, randomly selected different set of weights in the fully connected layer is updated. Thus, the use of dropout layer reduces the size of $\boldsymbol{\theta}_t$ and $\mathbf{g}_k(\boldsymbol{\theta}_t)$, thereby, reducing model transmission overhead. Finally, the last layer is output regression layer, yielding the output channel estimate of size $2M(L+1)\times 1$. The CNN architecture is similar to the one in [13] with the main difference being that the current model accepts $(L+1)\times\bar{M}\times 3$ input with fewer pilots instead of $(L+1)\times M\times 3$ as in [13].

4.2. Transmission Overhead

Transmission overhead can be defined as the size of the transmitted data during model training [7, 9, 13, 20]. Let $\mathcal{T}_{\mathrm{FL}}$ and $\mathcal{T}_{\mathrm{CL}}$ denote the transmission overhead of FL and CL, respectively. Then, we can define $\mathcal{T}_{\mathrm{CL}}$ as $\mathcal{T}_{\mathrm{CL}} = \big(3\bar{M}(L+1) + 2M(L+1)\big)\mathsf{D}$, which includes the number of symbols in the uplink transmission of the training dataset from the users to the BS. In contrast, the transmission overhead of FL includes the transmission of $\mathbf{g}_k(\boldsymbol{\theta}_t)$ and $\boldsymbol{\theta}_t$ in uplink and downlink communication for $t = 1, \ldots, T$, respectively. Compared to [13], the first term on the right hand side of $\mathcal{T}_{\mathrm{CL}}$ is computed with $\bar{M}$ instead of M. Finally, $\mathcal{T}_{\mathrm{FL}}$ is given by

$$\mathcal{T}_{\mathrm{FL}} = 2PTK. \tag{14}$$

We can see that the dominant terms are D and P, which are the number of training data pairs and the number of CNN parameters, respectively. While D can be adjusted according to the amount of available data at the users, P is usually unchanged during model training. Here, P is computed as

$$P = \underbrace{N_{\mathrm{CL}}(CN_{\mathrm{SF}}W_xW_y)}_{\text{Convolutional Layers}} + \underbrace{\kappa N_{\mathrm{SF}}W_xW_yN_{\mathrm{FCL}}}_{\text{Fully Connected Layers}}, \qquad (15)$$

where $N_{\mathrm{CL}} = 3$ is the number of convolutional layers and $C = 3$ is the number of spatial "channels". $W_x = W_y = 3$ are the 2-D kernel sizes. As a result, we have $P = 600,192$.

5. NUMERICAL SIMULATIONS

The goal of the simulations is to compare the performance of the proposed FL-based channel estimation approach with the state-of-the-art model-based channel estimation technique SF-CNN [16] together with the MMSE and LS estimation in terms of normalized MSE (NMSE) [21], defined by $\mathrm{NMSE} = \frac{1}{J_T K}\sum_{i=1}^{J_T}\sum_{k=1}^{K}\frac{\|\mathbf{\Sigma}_k - \hat{\mathbf{\Sigma}}_{k,i}\|_{\mathcal{F}}^2}{\|\mathbf{\Sigma}_k\|_{\mathcal{F}}^2}$, for $J_T = 100$ number of Monte Carlo trials. We also present the validation RMSE of the training process, for which the validation dataset includes 20% of the whole dataset $\mathcal{D}$.

The local dataset of each user includes $N = 100$ different channel realizations for $K = 8$ users. The number of antennas at the BS and the IRS are $M = 64$ and $L = 64$, respectively. Also, we select $N_{\mathrm{r}} = 5$ and the number of pilots is $\bar{M} = 32$, unless stated otherwise. The location of each user is selected as $\phi_{k,n} \in \Phi_k$ and $\varphi_{k,n} \in \bar{\Psi}_k$, where Φ_k and $\bar{\Psi}_k$ are the equally-divided subregions of the angular domain Θ, i.e., $\Theta = \bigcup_k \Phi_k = \bigcup_k \bar{\Psi}_k$, respectively. The pilot data are generated as $\mathbf{S} = \mathbf{I}_M$ and $\overline{\mathbf{S}}$ is composed of the $\bar{M}$ columns of $\mathbf{S}$. During training, $N = 200$ channel realizations are conducted and we have added AWGN on the input data for three SNR levels, i.e., SNR$= \{20, 25, 30\}$ dB, for $G = 160$ realizations in order to provide robust performance against noisy input [5]. As a result, the number of input-output pairs in the whole training dataset is $\mathsf{D} = 3KNG = 3 \cdot 8 \cdot 200 \cdot 160 = 768,000$. The proposed CNN model is realized and trained in MATLAB on a PC with a 2304-core GPU. For CL, we use the SGD algorithm with momentum of 0.9 and the mini-batch size $M_B = 128$, and update the network parameters with learning rate 0.001. For FL, we train the CNN for $T = 100$ iterations/rounds. Once the training is completed, the labels of the validation data (i.e., 20% of the whole dataset) are used in prediction stage. During the prediction stage, each user estimates its own channel by feeding the CNN with $\mathbf{\Upsilon}_k$ and obtains $\hat{\mathbf{h}}_{\mathrm{BS},k}$ and $\hat{\mathbf{G}}_k$ at the output.

Let us first compare the transmission overhead of the proposed FL approach with CL. Using the analysis in Section 4.2 and the above system settings, we get $\mathcal{T}_{\mathrm{CL}} = 1.1 \times 10^{10}$ whereas $\mathcal{T}_{\mathrm{FL}} = 9.6 \times 10^{8}$, which is approximately 12 times lower than $\mathcal{T}_{\mathrm{CL}}$. This demonstrates the effectiveness of the FL over CL in terms of transmission overhead. The complexity of the proposed CNN approach is also lower than that of SF-CNN, for which the detailed complexity analysis is presented in [13]. In the sequel, we present the NMSE performance of the proposed approach.

Fig. 1a shows the channel estimation NMSE with respect to SNR. Notice that CL provides better performance than that of FL for the proposed CNN model since it has access to the whole dataset at once. Nevertheless, FL has satisfactory channel estimation performance despite decentralized training and outperforms SF-CNN with CL. Specifically, the proposed CNN with FL and CL have similar NMSE for SNR$\leq$ 25 dB and the performance of FL maxes out in high SNR regime. This is because the learning model loses precision due to FL training and cannot perform better. SF-CNN also exhibits similar behavior but performs worse than the proposed method. This is because SF-CNN has convolutional-only layers. In contrast, the proposed CNN includes both convolutional and fully connected layers, exhibiting better feature extraction and data mapping performance.

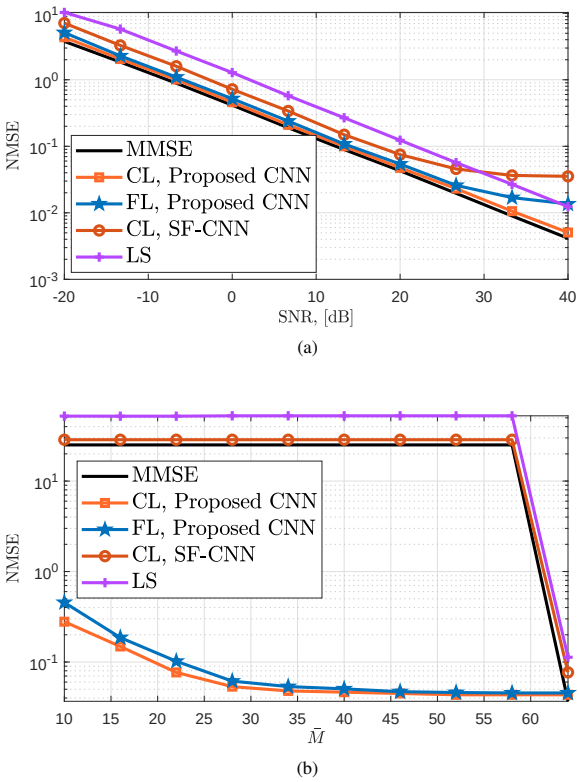


Fig. 1. Channel estimation NMSE with respect to (a) SNR when $\bar{M} = 32$ and (b) $\bar{M}$ when SNR $= 20$ dB.

Fig 1b shows the channel estimation NMSE with respect to the number of pilot signals $\bar{M}$ for SNR$=$ 20 dB. We can see that the proposed approach provides significantly better performance in the presence of insufficient pilot signals, i.e., $\bar{M} < M$, whereas the other algorithms perform poorly since they rely on the complete channel data, demanding $\bar{M} \geq M$.

6. CONCLUSIONS

We propose an FL framework for channel estimation in IRS-assisted massive MIMO systems. Via simulations, we demonstrate that FL-based channel estimation approach requires approximately 60% fewer less pilot signals as compared to the model-based techniques. Moreover, FL-based training enjoys approximately 12 times lower transmission overhead than CL-based training while providing a channel estimation performance close to CL. We further observe that at least 15 dB SNR on the model parameters is required for reliable channel estimation performance, i.e, NMSE ≤ 0.01. As a future work, we plan to study the FL-based models for joint channel estimation and IRS reflect beamformer design.

7. REFERENCES

[1] S. Gong, X. Lu, D. T. Hoang, D. Niyato, L. Shu, D. I. Kim, and Y. Liang, "Towards Smart Wireless Communications via Intelligent Reflecting Surfaces: A Contemporary Survey," *IEEE Commun. Surveys Tuts.*, pp. 1–1, 2020.

[2] 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.

[3] A. M. Elbir and K. V. Mishra, "A Survey of Deep Learning Architectures for Intelligent Reflecting Surfaces," *arXiv preprint arXiv:2009.02540*, 2020.

[4] S. Liu, Z. Gao, J. Zhang, M. D. Renzo, and M. Alouini, "Deep Denoising Neural Network Assisted Compressive Channel Estimation for mmWave Intelligent Reflecting Surfaces," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9223–9228, 2020.

[5] A. M. Elbir, A. Papazafeiropoulos, P. Kourtessis, and S. Chatzinotas, "Deep Channel Learning for Large Intelligent Surfaces Aided mm-Wave Massive MIMO Systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 9, pp. 1447–1451, 2020.

[6] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *arXiv*, Feb 2016. [Online]. Available: https://arxiv.org/abs/1602.05629v3

[8] A. M. Elbir, A. K. Papazafeiropoulos, and S. Chatzinotas, "Federated Learning for Physical Layer Design," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 81–87, Dec 2021.

[9] M. Mohammadi Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, 2020.

[10] T. Zeng, O. Semiari, M. Mozaffari, M. Chen, W. Saad, and M. Bennis, "Federated Learning in the Sky: Joint Power Allocation and Scheduling with UAV Swarms," *arXiv preprint arXiv:2002.08196*, 2020.

[11] A. M. Elbir and S. Coleri, "Federated Learning for Vehicular Networks," *arXiv preprint arXiv:2006.01412*, 2020.

[12] A. M. Elbir and S. Coleri, "Federated Learning for Hybrid Beamforming in mm-Wave Massive MIMO," *IEEE Commun. Lett.*, pp. 1–1, 2020.

[13] A. M. Elbir and S. Coleri, "Federated Learning for Channel Estimation in Conventional and RIS-Assisted Massive MIMO," *IEEE Trans. Wireless Commun.*, p. 1, Nov 2021.

[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] 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.

[17] A. M. Elbir, K. V. Mishra, M. R. B. Shankar, and B. Ottersten, "A Family of Deep Learning Architectures for Channel Estimation and Hybrid Beamforming in Multi-Carrier mm-Wave Massive MIMO," *IEEE Trans. Cognit. Commun. Networking*, p. 1, Dec 2021.

[18] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, 2017, pp. 1709–1720.

[19] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the Convergence of FedAvg on Non-IID Data," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HJxNAnVtDS

[20] M. M. Amiri and D. Gündüz, "Federated Learning Over Wireless Fading Channels," vol. 19, no. 5, pp. 3546–3557, 2020.

[21] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, 2019.