Federated Unfolding Learning for CSI Feedback in Distributed Edge Networks

Chongyang Tan, *Graduate Student Member, IEEE*, Donghong Cai, *Member, IEEE*, Fang Fang, *Senior Member, IEEE*, Zhiguo Ding *Fellow, IEEE*, and Pingzhi Fan *Fellow, IEEE*

***Abstract*—In distributed edge networks employing frequency division duplex, the feedback of channel state information (CSI) from the edge devices to the edge server always consumes a lot of spectrum resources, resulting in a serious communication burden. In this paper, we first propose an end-to-end unfolding neural network framework inspired by the soft threshold iterative algorithm (U-ISTANet). The proposed U-ISTANet integrates the advantages of compression awareness and neural networks. Especially, the compression matrix and sparse transformation of channel matrix can be learned for accurate CSI compression and recovery. And a lightweight version of U-ISTANet, called U-ISTANet-L, is proposed to reduce the training parameters. To reduce the data transmission overhead in the centralized learning framework, we extend the proposed U-ISTANet-L to a federated U-ISTANet-L (FU-ISTANet-L), which can train a more generalizable model by increasing the number of edge devices to enlarge the data set in a distributed learning manner. The proposed FU-ISTANet-L reduces the transmission overhead and increases the training speed while achieving a performance close to that of centralized learning. Furthermore, we propose a personalized FU-ISTANet-L (P-FU-ISTANet-L) to solve the heterogeneous data training problem in different communication environments. Specifically, we first obtain a pre-trained model by federation unfolding learning, and then each edge device fine-tunes the model using only a small amount of train data to obtain a personalized model for local channel environment. Extensive experimental results are provided to show that the proposed networks achieve a significant performance over the benchmarking schemes in terms of the normalized mean square error.**

***Index Terms*—Distributed edge networks, CSI feedback, frequency division duplex, federal unfolding learning, personalized.**

I. Introduction

In distributed edge networks, the edge server is often equipped with massive antennas to achieve high-speed and high-reliability communications. By enlarging the number of antennas, the network can increase the transmission rate and interference robustness of the system and greatly improve the link capacity and spectral efficiency [1]–[4]. This advantage is further extended especially with the addition of techniques, such as precoding, beamforming, and power allocation. However, the efficacy of these techniques relies on the availability and accuracy of the downlink channel state information (CSI) [5], [6]. Therefore, it is imperative to ensure that the edge server has unrestricted access to the requisite downlink CSI data.

In the time-division duplex systems (TDD), the edge device can acquire the downlink CSI from the estimated uplink CSI due to the reciprocity of TDD channels. However, in the frequency division duplex (FDD) systems, there is no reciprocity between the uplink and downlink channels [7]. Traditional channel feedback reduces the feedback overhead by exploiting the spatial and temporal correlation of CSI or by using compressed sensing (CS) [8], but these methods often encounter limitations improving CSI recovery quality and they require the prior information.

Recently, a large number of CSI feedback methods based deep learning (DL) has been proposed [9]–[11], which outperforms traditional methods in terms of CSI recovery performance. In [12], the authors first proposed an auto-encoder structure termed CsiNet to reduce the dimensionality of CSI. Their study demonstrates the great advantage and potential of DL in CSI feedback. CsiNet outperforms traditional CS-based methods at various compression rates. The authors in [13] introduced the integration of a long short-term memory (LSTM) network into CsiNet, which is designed to capture the inter-time-slot correlations in time-varying channels. Further, the CsiNet+ was proposed by modifying CsiNet in [14], which further improves the feedback performance by exploiting the sparsity feature of CSI in the angular delay domain and combing with the idea of the refinement in ReconNet [15]. These networks inherit most of the architectural design of CsiNet, and achieve performance improvements but with some additional computational overhead. In order to improve CSI feedback performance while reducing the complexity of network computation, a channel reconstruction network (CRNet) based on a multi-resolution architecture was proposed in [16], and the feedback performance of CRNet is significantly better than that of CsiNet for the same computational complexity. For the same purpose to cope with the complex numbers in CSI, while remaining lightweight, a DL-based model called CLNet was presented in [17]. This model is distinguished

The work of Donghong Cai was supported by the Science and Technology Major Project of Tibetan Autonomous Region of China (No. XZ202201ZD0006G02), the National Natural Science Foundation of China (No. 62001190), and the Basic and Applied Basic Research Foundation of Guangdong Province (No. 2024A1515012398); the work of Pingzhi Fan was supported by NSFC project No.62020106001. The code is available at https://github.com/Donghong-Cai/FedUnfolding. (Corresponding author: Donghong Cai.)

C. Tan and D. Cai are with the College of Information Science and Technology, Jinan University, Guangzhou 510632, China (e-mail: cyatan@stu2021.jnu.edu.cn; dhcai@jnu.edu.cn).

Fang Fang is with the Department of Electrical and Computer Engineering, and also with the Department of Computer Science, Western University, London, ON N6A 3K7, Canada (e-mail: fang.fang@uwo.ca).

Z. Ding is with the Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi 127788, UAE. (e-mail: zhiguo.ding@ieee.org).

P. Fan is with the Key Lab of Information Coding and Transmission, Southwest Jiaotong University, Chengdu 610031, China (e-mail: p.fan@ieee.org).

by its integration of a spatial attention mechanism. It was specifically designed to enable CLNet to selectively focus on the most significant segments of the clustered signal. This feature enhances the model's ability to effectively process and interpret CSI data. In addition, due to the excellent performance of transformer [18] in the field of deep learning, it was successfully applied to CSI feedback in [19] and was shown to achieve superior performance.

However, the above approach of using DL for CSI feedback also has limitations. In more stringent channel conditions and under higher compression demands, these methods demonstrate merely satisfactory performance. In addition, a trained model can perform well in a specific training environment, but its performance significantly declines when applied to a new channel environment due to limited generalizability. This necessitates retraining the deep neural network (DNN) with new CSI from the novel environment. For FDD massive MIMO systems with high-dimensional downlink CSI, retraining a model requires a large amount of data, which will result in high training costs. In DL, migration learning and meta-learning are often investigated to solve these types of problems. In [20], the prediction of downlink channels was transformed into a deep transfer learning (DTL) problem, in which each learning task focuses on predicting the downlink channel based on the uplink CSI for a particular environment. Training of the classical DNN involves the use of data from all previous environments, followed by fine-tuning on a small set of data from the new environment. [21] used a DTL based on a fully convolutional network architecture and developed a model-agnostic meta-learning (MAML)-based algorithm to deal with the high training costs of CSI feedback networks. However, these works have mainly focused on the centralized learning scheme, which requires uploading all local data to the edge server before training. This approach results in significant communication cost and privacy leakage issues. To address this problem, federated learning was proposed [22], [23], which is a distributed learning approach and each edge server sends only local training model parameters instead of local dataset. Federated learning has been applied to wireless communication fields, such as channel estimation, power allocation and hybrid beamforming design [24]–[26].

To cope with data heterogeneity in federated learning, many personalized federated learning methods have been proposed recently. For example, in [27], [28], each client jointly trains a global model as well as a local model, and then interpolates them to derive a personalized model. While, when local distributions and the average distribution are far apart, this approach often degenerates to every client learning only on its own local data. Then cluster FL is equally widely researched, which allows the grouping of clients into clusters so that clients belonging to the same cluster can share the same optimal model [29]. Essentially, clustered FL algorithms are trying to group together clients with similar distributions so that clients in the same cluster can leverage each other's data to perform federated learning more effectively. In fact, several clustered federated learning algorithms exist. [30] attempt to learn these distribution similarities indirectly when clients learn the cluster to which they should belong during training, but the initial clusters may act as noise affecting model convergence and performance. While [31] directly aims to efficiently identify distribution similarities among clients by analyzing the principal angles between the client data subspaces. However, these approaches are not advantageous when the data is highly heterogeneous, where the cluster can not always get great results. And retraining is still required when encountering new data.

Although the traditional CS algorithm may not outperform DL approaches in terms of feedback performance, incorporating the ideas from CS reconstruction can guide the design of neural networks to improve their accuracy and enhance their overall performance, such as, unfolding networks based on CS algorithms [32]. Recently, there have been a number of data-driven approaches designed based on CS theory. It has been widely studied in developing networks based on the iterative soft thresholding algorithm. [33] proposed learned iterative shrinkage thresholding algorithm (LISTA), which first expands the ISTA into a deep learning network, converting the matrices into a learnable feedforward neural network. This advancement has demonstrated efficacy in sparse signal recovery, yielding notable improvements in performance and efficiency. [34] proposed ALISTA based on LISTA and it reduces the training parameters in the network without any reduction in performance, and provides a theoretical analysis of the unfolding network. Inspired by the ISTA, [35] developed a fast and accurate CS reconstruction algorithm that utilizes a network to learn sparse transformations for natural images, ISTA-Net and ISTA-Net+, which combine the structural insights of conventional CS techniques with the rapid computational capabilities of neural networks. But these works do not systematically explore the performance of CSI feedback. Then, [36] from the same perspective, by incorporating model-based sparse recovery networks, TiLISTA-Joint is proposed, representing an innovative approach to improving CSI feedback. However, it requires too many training parameters.

In this paper, inspired by the success of ISTA-Net+, we design an ISTA-based DNN and propose an unfolding ISTA network (U-ISTANet) and its lightweight version, called U-ISTANet-L, which demonstrate enhanced capabilities in terms of compression and recovery of CSI compared to existing models. The proposed U-ISTANet not only learns the deeper sparse transform of CSI and the hyperparameters of ISTA, but also trains a compressed sensing matrix suitable for CSI and the input initial values of ISTA without manual design. Furthermore, we train the proposed U-ISTANet in the federated learning framework. Subsequently, a federated U-ISTANet-L (FU-ISTANet-L) is proposed for CSI feedback, which improves the generalization of the DNN by expanding the training dataset through distributed computation, and reducing the communication overhead associated with transmitting the dataset. In addition, to address the challenge of model retraining necessitated by significant alterations in the channel environment, we develop a personalized FU-ISTANet-L (P-FU-ISTANet-L) based on the concept of data heterogeneity. It enables DNNs to maintain high-quality feedback with minimal training, even when working with small batch datasets. Our contributions are mainly as follows.

- First, an end-to-end U-ISTANet and its lightweight version U-ISTANet-L are proposed for CSI feedback in distributed edge networks. The strict sparse transformation and inverse transformation of CSI are learnt from the perspective of sparse recovery. The model network structure is built according to the ISTA framework to perform CSI compression and recovery, and does not require any manually designed hyperparameters.
- Then the proposed U-ISTANet-L is extended to FU-ISTANet-L with federated learning framework, and we analyze the communication costs of using FU-ISTANet-L for CSI feedback in distributed systems. Compared to centralized learning, one can obtain a more generalizable model without sending huge datasets to be trained in the edge server. The communication cost of training only depends on the number of parameters of the neural network. Thus, the system can fully utilize the distributed data to learn the global CSI feedback model in the distributed learning framework without large-scale communication overhead.
- Finally, we design a P-FU-ISTANet-L for the CSI feedback of distributed networks, to combine a meta-learning approach where multiple edge devices in a federation environment learn an initial global model using a large amount of data. Moreover, adapting to a new channel environment, our approach necessitates only minimal data for the edge device to recalibrate the model. This streamlined adjustment process rapidly yields improved feedback performance.

The rest of this paper is organized as follows. Section II introduces the system model and the CSI feedback problem of distributed edge neworks. Section III presents the proposed end-to-end U-ISTANet, U-ISTANet-L and the FU-ISTANet for CSI feedback. The P-FU-ISTANet-L for data heterogeneity is then proposed in Section IV. Section V and Section VI present the experimental results and the conclusions, respectively.

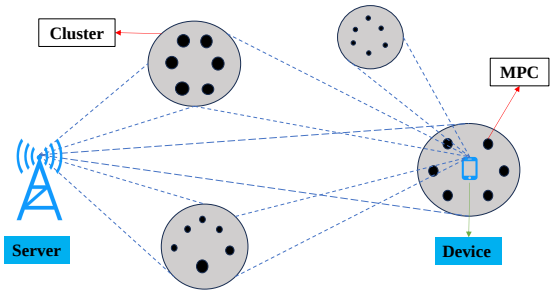
Notations: $\mathbf{A}, \mathbf{a}, a$ denote matrix, vector and scalar, respectively. $\mathrm{vec}(\mathbf{A})$ is the vector representation of matrix $\mathbf{A}$, and $\mathrm{invec}(\mathbf{a})$ is the matrix representation of vector $\mathbf{a}$, $(\mathbf{A})_{i,j}$ is element a_{ij} of matrix $\mathbf{A}$ in the i-th row and the j-th column, $\mathbf{a}(i)$ is the i-th element of vector $\mathbf{a}$ and $\otimes$ is Kronecker product.

II. System Model

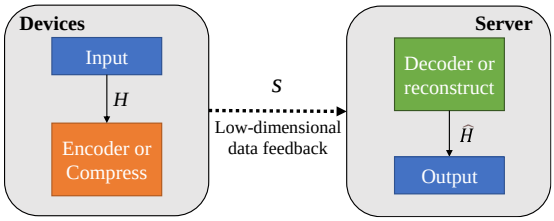
We consider the CSI feedback of a distributed edge communication network with N_c subcarriers, where an edge server equipped with a uniform linear antenna array with $N_{Es} \gg 1$ elements serves K distributed single-antenna edge devices, where the antenna spacing is $\kappa > 0$. It is assumed that, at the n_c-th subcarrier, the channel vector between the edge server and the k-th edge device can be written as

$$\tilde{\mathbf{h}}_k[n_c] = [\tilde{h}_k[n_c][1], \tilde{h}_k[n_c][2], \cdots, \tilde{h}_k[n_c][N_{ES}]]^H. \quad (1)$$

Note that the t delay time-domain channel $\bar{\mathbf{h}}_k[t]$ with COST 2100 channel model [37] containing N clusters and each cluster includes P multipath components (MPCs), where the delay



(a) Illustration of the channel model.



(b) The process of downlink CSI feedback.

Fig. 1: Illustration of system model.

of the p-th path in the n-th cluster for the k-th edge device is $\tau_{k,n,p}$ and the angle of departure (AoD) is $\phi_{k,n,p} \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Then, the array steering vector between the edge server and the k-th edge device is

$$\mathbf{c}_k(\phi_{k,n,p}) = [1, e^{j\frac{2\pi}{\lambda}\kappa \sin\phi_{k,n,p}}, ..., e^{j\frac{2\pi}{\lambda}\kappa(N_{ES}-1)\sin\phi_{k,n,p}}]^T, \quad (2)$$

where $\lambda = \frac{\iota}{f_c}$, and ι denotes the speed of light, and f_c is the carrier frequency.

The t delay time-domain CSI vector between the edge server and the k-th edge device is expressed as

$$\bar{\mathbf{h}}_k[t] = \sum_{n=1}^{N} \frac{V_n}{L} \sqrt{\frac{\chi_n}{L_n}} \left[\sum_{p=1}^{P} \alpha_{k,n,p} \mathbf{c}_k(\phi_{k,n,p}) \delta(t - \tau_{k,n} - \tau_{k,n,p}) \right], \quad (3)$$

where N is the number of clusters, and the complex path gain of the p-th path in the n-th cluster of the k-th edge device is denoted by $\alpha_{k,n,p}$, L is the overall pathloss, and the impact of entering or exiting the VR environment is expressed through the cluster visibility gain as shown in Fig. 1 (a), denoted by V_n, whereas the cluster shadow fading and attenuation are captured through χ_n and L_n, respectively, $\tau_{k,n}$ is the geometric delay of the k-th edge and $\delta(\cdot)$.

The time-domain channel in (3) can be transformed by N_c-point Discrete Fourier Transform (DFT) to obtain the frequency-domain CSI of the k-th edge device on the n_c-th subcarrier, which can be written as:

$$\tilde{\mathbf{h}}_k[n_c] = \sum_{t=0}^{T-1} \bar{\mathbf{h}}_k[t] e^{-j\frac{2\pi n t}{N_c}}, \quad (4)$$

where T is the cyclic prefix (CP) length. Subsequently, the channel between the edge server and the k-th edge device is

$$\tilde{\mathbf{H}}_k = [\tilde{\mathbf{h}}_k[1], \tilde{\mathbf{h}}_k[2], ..., \tilde{\mathbf{h}}_k[N_c]]^H \in \mathbb{C}^{N_c \times N_{ES}}. \quad (5)$$

The assumption is made, without loss of generality, that the edge devices possess perfect CSI. Subsequently, for precoding design purposes, it is necessary for the edge server to obtain CSI feedback from these devices. However, the direct feedback of full CSI consumes a large amount of communication resources and a more effective feedback method should be designed.

Note that the channel $\tilde{\mathbf{H}}_k$ is approximately sparse in the angular delay-domain, which can be calculated by

$$\mathbf{H}_k = \mathbf{F}_d \tilde{\mathbf{H}}_k \mathbf{F}_a^H, \quad (6)$$

where $\mathbf{F}_d \in N_c \times N_c$, $\mathbf{F}_a \in N_{ES} \times N_{ES}$ are DFT matrices. In the delay-domain, the temporal delay between multipath arrivals is finite, causing values to exist only in the first $N_s (N_s \leq N_c)$ rows of the matrix $\tilde{\mathbf{H}}_k \in \mathbb{C}^{N_s \times N_{ES}}$. As a result, retaining solely the first N_s rows of the matrix capitalizes on its sparsity, offering a viable compression method for CSI that lessens the communication burden of feedback. The detailed process of CSI feedback is described in Fig. 1 (b). At the edge device, the input CSI $\tilde{\mathbf{H}}_k$ is compressed by an encoder to obtain a vector $\hat{\mathbf{s}}_k$. The low-dimensional feature vectors are sent to the edge server thus reducing the communication overhead. At the edge server, the received $\hat{\mathbf{s}}_k$ is reconstructed by a decoder to obtain the estimated CSI $\hat{\mathbf{H}}_k$.

III. Federated Unfolding Learning for CSI Feedback

In this section, we propose a neural network model trained by the edge devices and edge server in federated learning framework. Specially, we first briefly describe the ISTA and unfold it into an end-to-end network model. Then the designed unfold ISTA network (U-ISTANet) and its lightweight version U-ISTANet-L are extended to the distributed networks with a federated learning framework.

A. The Proposed End-to-End U-ISTANet

The deep learning methods have been widely applied to the CSI feedback due to the excellent compression performance, which can be considered as an autoencoder containing an encoder and a decoder for compressing and recovering the CSI. More precisely, an encoder is deployed at the edge device to extract CSI features and a decoder is deployed at the edge server to recover the CSI. In particular, the encoder is defined as

$$\mathbf{s}_k = f_{en}(\tilde{\mathbf{H}}_k) \in \mathbb{R}^{N_{cr}}, \quad (7)$$

where $N_{cr} \ll N_s N_{ES}$. The edge device reduces the communication burden by converting the channel matrix into a low-dimensional vector $\mathbf{s}_k$ via an encoder and feeding it back to the edge server instead of directly to the CSI. From the perspective of compressed sensing, the encoder formulated in (7) can be expressed as

$$\mathbf{s}_k = \mathbf{\Phi} \mathbf{x}_k, \quad (8)$$

where $\mathbf{\Phi} \in \mathbb{C}^{N_{cr} \times N_c N_{ES}}$ is a linear transformation matrix, and $\mathbf{x}_k = \mathrm{vec}(\tilde{\mathbf{H}}_k) \in \mathbb{C}^{N_s N_{ES}}$.

Meanwhile, the following decoder is deployed at the edge server:

$$\hat{\mathbf{H}}_k = f_{de}(\mathbf{s}_k). \quad (9)$$

The edge server relies on the decoder to recover the full CSI from $\mathbf{s}_k$ which is fed back from the edge device. Therefore, the CSI feedback model can be expressed as

$$\hat{\mathbf{H}}_k = f_{de}(f_{en}(\tilde{\mathbf{H}}_k)). \quad (10)$$

It is important to point out that the decoder (9) can be designed based on a sparse signal recovery problem [4] by exploiting the special sparse structure of $\mathbf{H}_k$ in (6). Thus, the task of decoder is to reconstruct $\mathbf{x}_k$ from $\mathbf{s}_k$ and $\mathbf{\Phi}$ by solving the following problem:

$$\hat{\mathbf{x}}_k = \underset{\mathbf{x}_k}{\mathrm{argmin}} \frac{1}{2} \|\mathbf{s}_k - \mathbf{\Phi} \mathbf{x}_k\|_2^2 + \lambda \|\mathbf{x}_k\|_1, \quad (11)$$

which is a Lasso problem, and can be solved by ISTA. Especially, the iterations of ISTA for problem (11) are

$$\mathbf{r}_k^{(i)} = \mathbf{x}_k^{(i-1)} - \rho \mathbf{\Phi}^T (\mathbf{\Phi} \mathbf{x}_k^{(i-1)} - \mathbf{s}_k), \quad (12)$$

$$\mathbf{x}_k^{(i)} = \underset{\mathbf{x}_k}{\mathrm{argmin}} \frac{1}{2} \|\mathbf{x}_k - \mathbf{r}_k^{(i)}\|_2^2 + \lambda \|\mathbf{x}_k\|_1. \quad (13)$$

We can solve (13) by using the soft threshold function, i.e.,

$$\mathbf{x}_k^{(i)} = \mathrm{soft}(\mathbf{r}_k^{(i)}, \lambda), \quad (14)$$

where $\mathrm{soft}(\cdot)$ is the soft threshold function, which is defined as $\mathrm{soft}(\mathbf{r}_k^{(i)}, \lambda) = \mathrm{sign}(\mathbf{r}_k^{(i)}) \max(|\mathbf{r}_k^{(i)}| - \lambda, 0)$ with the sign function $\mathrm{sign}(\cdot)$.

Remark 1. *Different channels have specific sparse fields, for example, the CSI $\mathbf{x}_k$ can be represented sparsely in angular-delay domain for the far-field channel, thus the sparse penalty term of* (11) *can be represented exactly to design the signal recovery algorithm with a proper transformation.*

To obtain higher reconstruction accuracy, it is assumed that there exists a non-liner transformation $\hat{f}(\cdot)$ and a liner transformation $\mathbf{W}$ such that $\mathbf{x}_k$ is strictly sparse. We use a learnable linear matrix instead of the linear transformation $\mathbf{W}$ and the sensing matrix $\mathbf{\Phi}$ of the encoder, and this matrix can be denoted by $\mathbf{W}_1$. Then the reconstruction problem (11) is further written as

$$\begin{aligned} \hat{\mathbf{x}}_k^{(i)} &= \underset{\mathbf{x}_k}{\mathrm{argmin}} \frac{1}{2} \|\mathbf{s}_k - \mathbf{W}_1 \mathbf{x}_k\|_2^2 + \lambda \|\hat{f}(\mathbf{W} \mathbf{x}_k)\|_1 \\ &= \underset{\mathbf{x}_k}{\mathrm{argmin}} \frac{1}{2} \|\mathbf{s}_k - \mathbf{W}_1 \mathbf{x}_k\|_2^2 + \lambda \|f(\mathbf{x}_k)\|_1, \end{aligned} \quad (15)$$

where $f(\mathbf{x}_k) \triangleq \hat{f}(\mathbf{W} \mathbf{x}_k)$. Similar to problem (11), we try to solve the problem (15) using ISTA. Equation (12) remains unchanged and (13) evolves as follows:

$$\mathbf{x}_k^{(i)} = \underset{\mathbf{x}_k}{\mathrm{argmin}} \frac{1}{2} \|\mathbf{x}_k - \mathbf{r}_k^{(i)}\|_2^2 + \lambda \|f(\mathbf{x}_k)\|_1. \quad (16)$$

the calculation of $\mathbf{r}_k^{(i)}$ can be expressed as

$$\mathbf{r}_k^{(i)} = \mathbf{x}_k^{(i-1)} - \rho^{(i)} \mathbf{W}_1^T (\mathbf{W}_1 \mathbf{x}_k^{(i-1)} - \mathbf{s}_k). \quad (17)$$

However, due to the complex non-orthogonal transform $f(\cdot)$, it is still challenging to solve the problem (16). Fortunately, we can solve problem (16) by using the DL approach. In [35], it is proven that $\|f(\mathbf{x}_k)-f(\mathbf{r}_k^{(i)})\|_2^2 \approx \alpha\|\mathbf{x}_k-\mathbf{r}_k^{(i)}\|_2^2$, where $\alpha > 0$ is a scalar parameter. Specifically, $f(\cdot) = \mathbf{B}\mathrm{ReLu}(\mathbf{A}\cdot)$, where $\mathbf{A}$ and $\mathbf{B}$ are liner projection matrices, and $\mathrm{ReLu}(\cdot)$ is the activation functions. This transformation represents a complete hidden layer in a neural network. Therefore, problem (16) can be expressed as

$$\mathbf{x}_k^{(i)} = \underset{\mathbf{x}_k}{\operatorname{argmin}} \frac{1}{2}\|f(\mathbf{x}_k)-f(\mathbf{r}_k^{(i)})\|_2^2 + \theta\|f(\mathbf{x}_k)\|_1, \quad (18)$$

where θ is also a constant, i.e., $\theta = \lambda\alpha$. Similarly, we can use the soft threshold function to obtain a closed-form solution for $f(\mathbf{x}_k^{(i)})$, i.e.,

$$f(\mathbf{x}_k^{(i)}) = \mathrm{soft}(f(\mathbf{r}_k^{(i)}), \theta). \quad (19)$$

Further, it is only necessary to find the inverter transformation $\tilde{f}(\cdot)$ for $f(\cdot)$ and $\mathbf{x}_k^{(i)}$ can be obtained, that is

$$\mathbf{x}_k^{(i)} = \tilde{f}(\mathrm{soft}(f(\mathbf{r}_k^{(i)}), \theta)). \quad (20)$$

Based on the iterations in (12) and (20), we can design the neural network to unfold ISTA for channel feedback from a sparse recovery perspective.

Note that the iterations for solving problem (15) can be summarized as

$$\begin{cases} \mathbf{r}_k^{(i)} = \mathbf{x}_k^{(i-1)} - \rho^{(i)}\mathbf{W}_1^T(\mathbf{W}_1\mathbf{x}_k^{(i-1)} - \mathbf{s}_k), \\ f(\mathbf{x}_k^{(i)}) = \mathrm{soft}(f(\mathbf{r}_k^{(i)}), \theta^{(i)}), \\ \mathbf{x}_k^{(i)} = \tilde{f}(f(\mathbf{x}_k^{(i)})). \end{cases} \quad (21)$$

From (21), we can see that each iteration is parameterized by its own step size $\rho^{(i)}$ and threshold $\theta^{(i)}$, which are set as learnable parameters of unfold ISTA to avoid the manual setting of hyperparameters in ISTA. Our proposed end-to-end network model is shown in Fig. 2. In fact, this network can be viewed as an autoencoder with the difference that its encoder is a simple linear projection. In contrast, the decoder, which is the reconfiguration network, is composed of several structurally identical phases corresponding to the iterations of ISTA.

Specifically, we take the CSI vector $\mathbf{x}_k$ as the input of the network, and after a linear projection of the encoder, the CSI vector is compressed into a feature vector $\mathbf{s}_k$. This completes the CSI compression, then the vector $\mathbf{s}_k$ is fed back to the decoder as its partial input. As can be seen in (21), in addition to the feature vector $\mathbf{s}_k$ and the projection matrix $\mathbf{\Phi}$ in the encoder, there is also an initial vector $\mathbf{x}_k^{(0)}$ as the input of the decoder.

In the ISTA, the initial vector $\mathbf{x}_k^{(0)}$ is usually set as a random vector or a zero vector. However, it can not be used for a data-driven approach. To make the initial values closer to the original CSI $\mathbf{x}_k$, an initial mapping matrix $\mathbf{Q}_k^{(0)}$ can be found for all the existing data pair $(\mathbf{x}_k, \mathbf{s}_k)$ of device k, i.e., the known CSI and its compressed vector. Using the least squares to calculate the input $\mathbf{x}_k^{(0)}$ as an initial vector. Specifically, the mapping $\mathbf{Q}_k^{(0)}$ is given by

$$\mathbf{Q}_k^{(0)} = \underset{\mathbf{Q}_k}{\operatorname{argmin}} \|\mathbf{Q}_k\mathbf{S}_k - \mathbf{X}_k\|_F^2 = \mathbf{X}_k\mathbf{S}_k^T(\mathbf{X}_k\mathbf{S}_k^T)^{-1}, \quad (22)$$

where $\mathbf{X}_k \in \mathbb{C}^{N_cN_{ES}\times|\mathcal{D}|}, \mathbf{S} \in \mathbb{C}^{N_s\times|\mathcal{D}|}$ represent all the data pair in the dataset $\mathcal{D}$ with size $|\mathcal{D}|$. Hence, given any input $\mathbf{s}_k \in \mathcal{D}$, its initialization $\mathbf{x}_k^{(0)}$ is computed as

$$\mathbf{x}_k^{(0)} = \mathbf{Q}_k^{(0)}\mathbf{s}_k. \quad (23)$$

In one basic phase of the network, i.e. one iteration corresponding to ISTA, $\mathbf{r}_k^{(i)}$ is calculated according to (21). We unfold it further, then $\mathbf{r}_k^{(i)}$ can be expressed as

$$\mathbf{r}_k^{(i)} = \mathbf{x}_k^{(i-1)} - \rho^{(i)}(\mathbf{W}_1^T\mathbf{W}_1\mathbf{x}_k^{(i-1)} - \mathbf{W}_1^T\mathbf{s}_k). \quad (24)$$

In order to improve the generalization ability of the model, we parameterize the Hessian matrix and $\mathbf{x}_k^{(0)}$ in (23). Thus we get

$$\mathbf{s}_k = \mathbf{W}_1\mathbf{x}_k \quad (25a)$$

$$\mathbf{r}_k^{(i)} = \mathbf{x}_k^{(i-1)} - \rho^{(i)}(\mathbf{W}_2\mathbf{x}_k^{(i-1)} - \mathbf{W}_1^T\mathbf{s}_k), \quad (25b)$$

$$\mathbf{x}_k^{(0)} = \mathbf{W}_3\mathbf{s}_k, \quad (25c)$$

where $\mathbf{W}_2, \mathbf{W}_3$ are learnable linear matrices. Then, the convolution operation is used to represent the sparse transform $f(\cdot)$ and its inverse transformation $\tilde{f}(\cdot)$. As shown in Fig. 2, all the convolution layers in the model are set to N_f filters, and each convolution kernel size is 3×3. We set $N_f = 32$ in our experiments.

Remark 2. *In the proposed U-ISTANet, $\mathbf{W}_2$ is set as new learnable linear matrix to improve the learning ability of the neural network. In the proposed U-ISTA-L, we let $\mathbf{W}_2 = \mathbf{W}_1^T\mathbf{W}_1$ to reduce the network parameters while maintaining the learning ability of the network.*

Let N_p denote the total number of the phases. Given that the basic unit contains multiple layers of neural networks that are ultimately connected by N_p units, this deep network structure requires the inclusion of residual connections. Furthermore, we introduce additional layers, $\mathcal{H}(\cdot)$ and $\tilde{\mathcal{H}}(\cdot)$, during the incorporation of residual connections. These layers extract residual features and facilitate recovery at the beginning and end of the basic block, respectively. The unit of the decoder can be represented as

$$\begin{aligned} \mathbf{x}_k^{(i)} &= \mathbf{r}_k^{(i)} + (\tilde{\mathcal{H}}(\tilde{f}(\mathrm{soft}(f(\mathcal{H}(\mathbf{r}_k^{(i)}))), \theta^{(i)}))) \\ &\triangleq \mathbf{r}_k^{(i)} + \tilde{\mathcal{F}}(\mathrm{soft}(\mathcal{F}(\mathbf{r}_k^{(i)}), \theta^{(i)})). \end{aligned} \quad (26)$$

Then the learnable parameters of our proposed network, including learnable matrices $\mathbf{W}_{1,2,3}$, step size $\rho^{(i)}$, threshold $\theta^{(i)}$, sparse transform $\mathcal{F}(\cdot)$ and inverse transform $\tilde{\mathcal{F}}(\cdot)$, can be expressed as $\mathbf{\Theta} = \{\mathbf{W}_{1,2,3}, \rho^{(i)}, \theta^{(i)}, \mathcal{F}(\cdot), \tilde{\mathcal{F}}(\cdot)\}_{i=1}^{N_p}$. To distinguish between local and global models, we define $\mathbf{\Theta}_k$ as the learned model parameter $\mathbf{\Theta}$ based the data $\mathbf{x}_k \in \mathcal{D}_k$ of device k; while $\mathbf{\Theta}_G$ denotes the global model parameter. In this model, we customize the loss function used for training. Given the dataset $\mathbf{x}_k \in \mathcal{D}_k$, our ultimate goal is to reconstruct the original CSI $\mathbf{x}_k$ at the edge server, i.e., to make the output $\hat{\mathbf{x}}$ of the network close to the input $\mathbf{x}_k$. This can be done using MSE as the loss function. We introduce the sparse transform and its inverse transform into the network and train

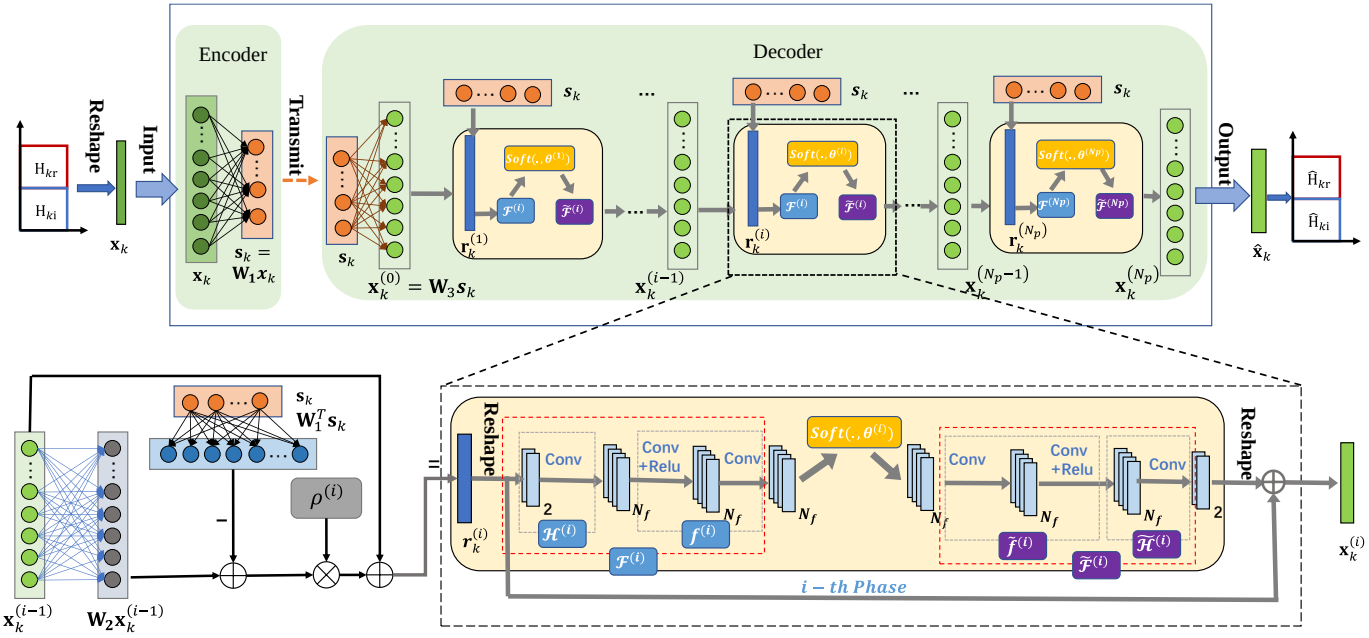


Fig. 2: Illustration of our proposed end-to-end U-ISTANet or U-ISTANet-L.

them specifically for $\mathbf{x}_k^{(i)} = \tilde{f}(f(\mathbf{x}_k^{(i)}))$ to hold. Therefore, we design the end-to-end loss function as follows:

$$\mathcal{L}(\mathbf{\Theta}_k) = \left(\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|_2^2 + \gamma \sum_{i=1}^{N_p} \|\tilde{f}(f(\mathcal{H}(\mathbf{r}_k^{(i)}))) - \mathcal{H}(\mathbf{r}_k^{(i)})\|_2^2 \right) \tag{27}$$

Remark 3. *The term* $\gamma \sum_{i=1}^{N_p} \|\tilde{f}(f(\mathcal{H}(\mathbf{r}_k^{(i)}))) - \mathcal{H}(\mathbf{r}_k^{(i)})\|_2^2$ *ensures that* $\tilde{f}(\cdot)$ *is trained as an inverse transformation of* $f(\cdot)$*, where* γ *is the regularization parameter, and* γ *is set as 0.01. On this basis, the term* $\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|_2^2$ *aims to minimize the difference between the original CSI* $\mathbf{x}_k$ *and its reconstruction* $\hat{\mathbf{x}}_k$*, thus ensuring that the DNN is updated in the desired direction.*

B. The Proposed FU-ISTANet-L

For distributed edge networks, the volume of local data of each edge device may be limited. This can hinder the ability of a single device to independently train a model with superior performance. In order to train a generalizable model, we use a federated learning framework, which can reduce the communication overhead caused by transmitting datasets for centralized learning and also obtain the performance gain from a larger dataset. The amount of communication does not increase with the size of the data set, it only depends on the number of parameters of the neural network model. In this way, the system can fully utilize the data owned by the devices without additional consideration of the communication burden when the number of parameters in the model is small.

Each edge device trains its local autoencoder model $\hat{\mathbf{x}}_k = f_{de}(f_{en}(\mathbf{x}_k)) \triangleq \mathcal{M}(\tilde{\mathbf{H}}_k, \mathbf{\Theta}_{k,t})$ using the local dataset $\mathcal{D}_k$, and minimizes the loss function to acquire the optimal parameters

$$\begin{aligned} \mathbf{\Theta}_{k,t} = &\underset{\mathbf{\Theta}_{k,t}}{\operatorname{argmin}} \sum_{\mathbf{x}_k} \mathcal{L}(\mathbf{\Theta}_{k,t}) \\ &\text{s.t. } \mathbf{x}_k \subseteq \mathcal{D}_k, \end{aligned} \tag{28}$$

where $\mathbf{\Theta}_{k,t}$ represents the model parameters trained by the k-th edge device during the t-th communication round. After each local model update, all edge devices upload their respective local model parameters to the edge server. The edge server uses aggregation algorithms, such as *Fedavg*, to aggregate a global model with stronger generalization ability, resulting in

$$\mathbf{\Theta}_{G,t} = \underset{\mathbf{\Theta}_{G,t}}{\operatorname{argmin}} \frac{1}{\sum_{k=1}^{K} |\mathcal{D}_k|} \sum_{k=1}^{K} \sum_{\mathbf{x}_k \in \mathcal{D}_k} \mathcal{L}(\mathbf{\Theta}_{k,t}), \tag{29}$$

where $\mathbf{\Theta}_{G,t}$ denotes the aggregated global model parameters during the t-th communication round. Then, the edge server broadcasts $\mathbf{\Theta}_{G,t}$ to the edge devices to start a new round of updates.

In fact, with the same size data set, the training performance of federated learning is no better than that of centralized learning, but federated learning has its own unique advantages. The specific process of the proposed FU-ISTANet-L is shown in Fig. 3. Each edge device collects a large amount of CSI between itself and the edge server in dataset $\mathcal{D}_k$ for training. First, the edge server initializes the model parameter for training and then broadcasts it to all edge devices to enable the training of federated learning. Furthermore, each edge device will individually train the network model given by the edge server with its own dataset. The local model parameter updating and global model parameter updating based on (28) and (32) are respectively presented as follows:

- **Local model parameter updating**. Actually, the optimization task of each edge device is described in (28),

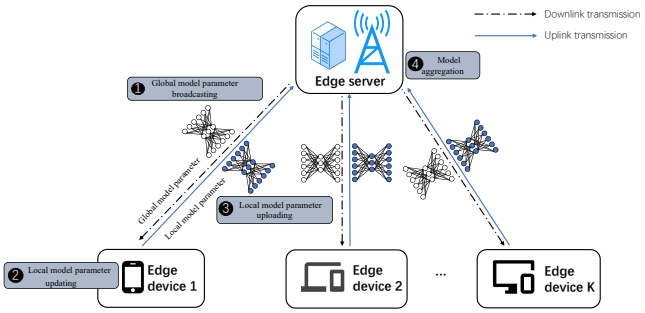


Fig. 3: The training process of the proposed FU-ISTANet.

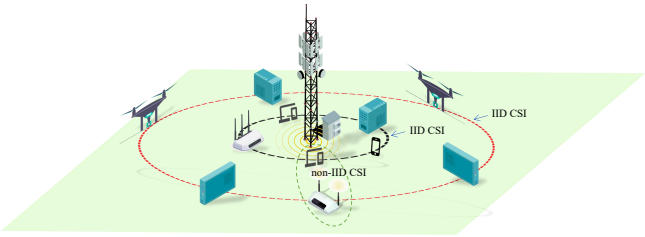


Fig. 4: The illustration of CSI feedback in heterogeneous environments.

and the training optimizer for neural networks is generally based on the gradient descent (GD) algorithm, thus the training process can be represented as

$$\mathbf{\Theta}_{k,t}(e) = \mathbf{\Theta}_{k,t}(e-1) - \eta \nabla \mathcal{L}(\mathbf{\Theta}_{k,t}(e-1)), \quad (30)$$

where $\mathbf{\Theta}_{k,t}(e)$ is the local model parameters for the e-th epoch of the k-th edge device, η is step size, and ∇ is the gradient symbol. When each edge device completes E epochs training, it uploads the magnitude of parameter alterations following updates across e epochs g_k to the edge server for aggregation, we still call it gradient, where

$$g_k = \mathbf{\Theta}_{k,t}(0) - \mathbf{\Theta}_{k,t}(e). \quad (31)$$

- **Global model parameter updating**. Different aggregation algorithms lead to different training results and communication burdens. In general, federated learning can use the simplest Fedavg algorithm, which directly weights the model gradient from all edge devices equally in the aggregation phase. Since the edge server has the last update of the global model, the aggregation can be expressed as

$$\mathbf{\Theta}_{G,t} = \mathbf{\Theta}_{G,t-1} - \frac{1}{\sum_{k=1}^{K} |\mathcal{D}_k|} \sum_{k=1}^{K} |\mathcal{D}_k| g_k. \quad (32)$$

Therefore, the iterations of our proposed FU-ISTANet-L are shown in **Algorithm 1**

Algorithm 1 Our Proposed FU-ISTANet-L

Input: Vector of CSI $\mathbf{x}$ to be feedback.
Output: Estimated CSI vectors $\hat{\mathbf{x}}$.

Training For distributed edge networks:
Initialization: The edge server initializes the parameters of the global model:
$\mathbf{\Theta}_{G,0} = \{\mathbf{W}_{1,2,3}, \rho^{(i)}, \theta^{(i)}, \mathcal{F}(\cdot), \tilde{\mathcal{F}}(\cdot)\}_{i=1}^{N_p}$
Start to train:
for $t = 1, 2, \ldots$ **do**
  Edge server broadcasts the global model parameters to K edge devices:
  $\mathbf{\Theta}_{k,t}(0) = \mathbf{\Theta}_{G,t-1}, \ k = 1, 2, \cdots, K$
  for each epoch $e = 1, 2, ...,$ edge devices parallel **do**
    $\mathbf{\Theta}_{k,t}(e) = \mathbf{\Theta}_{k,t}(e-1) - \eta \nabla \mathcal{L}(\mathbf{\Theta}_{k,t}(e-1))$
  EndFor
  $g_k = \mathbf{\Theta}_{k,t}(0) - \mathbf{\Theta}_{k,t}(e)$
  Local model gradient g_k of all edge devices are uploaded and the global model parameter is updated by
  $\mathbf{\Theta}_{G,t} = \mathbf{\Theta}_{G,t-1} - \frac{1}{\sum_{k=1}^{K} |\mathcal{D}_k|} \sum_{k=1}^{K} |\mathcal{D}_k| g_k.$
EndFor
CSI Feedback: Edge devices load the encoder and edge server loads the decoder for CSI feedback.
Encoder of edge device: $\mathbf{s}_k = \mathbf{W}_1 \mathbf{x}_k, k = 1, 2, \cdots, K.$
Decoder of edge server:
$\mathbf{x}_k^{(0)} = \mathbf{W}_3 \mathbf{s}_k$
for $i = 1, 2, ..., N_p$ **do**
  $\mathbf{r}_k^{(i)} = \mathbf{x}_k^{(i-1)} - \rho^{(i)}(\mathbf{W}_2 \mathbf{x}_k^{(i-1)} - \mathbf{W}_1^T \mathbf{s}_k)$
  Feature extraction: $\mathbf{r}_{k,1}^{(i)} = \mathcal{H}(\mathbf{r}_k^{(i)})$
  Sparse transformation: $\mathbf{r}_{k,2}^{(i)} = f(\mathbf{r}_{k,1}^{(i)})$
  Soft threshold calculation: $\boldsymbol{\mu}_{k,1}^{(i)} = \text{soft}(\mathbf{r}_{k,2}^{(i)}, \theta^{(i)})$
  Sparse inverse transformation: $\boldsymbol{\mu}_{k,2}^{(i)} = \tilde{f}(\boldsymbol{\mu}_{k,1}^{(i)})$
  Feature Recovery: $\boldsymbol{\mu}_k^{(i)} = \tilde{\mathcal{H}}(\boldsymbol{\mu}_{k,2}^{(i)})$
  $\mathbf{x}_k^{(i)} = \mathbf{r}_k^{(i)} + \boldsymbol{\mu}_k^{(i)}$
EndFor
return $\mathbf{x}_k^{(N_p)}$

C. Computational Complexity and communication overhead analysis

In this part, we analyze the complexity of the proposed network and the communication overhead for FL.

1) Computational Complexity Analysis: Note that the the main computational complexity of our proposed U-ISTANet and U-ISTANet-L derived from matrix operations and CNNs. Matrix operations are included in the computation of the encoder and the computation of $\mathbf{r}_k^{(i)}$, i.e.,

$$\begin{aligned} \mathbf{s}_k &= \mathbf{W}_1 \mathbf{x}_k, \\ \mathbf{r}_k^{(i)} &= \mathbf{x}_k^{(i-1)} - \rho^{(i)}(\mathbf{W}_2 \mathbf{x}_k^{(i-1)} - \mathbf{W}_1^T \mathbf{s}_k), \\ \mathbf{x}_k^{(0)} &= \mathbf{W}_3 \mathbf{s}_k. \end{aligned} \quad (33)$$

The computational complexity of (33) can be written as

$$\mathcal{C}_M = \mathcal{O}(N_{cr} N_s N_{ES} + (N_s N_{ES})^2). \quad (34)$$

Since $N_{cr} < N_s N_{ES}$ and each phase in the network needs to compute $\mathbf{r}_k^{(i)}$, the complexity in (34) can be further expressed as

$$\mathcal{C}_M = \mathcal{O}(N_p(N_s N_{ES})^2). \tag{35}$$

Generally, the complexity of a single convolutional layer can be given by

$$\mathcal{C}_l = \mathcal{O}(C_{in} C_{out} W H K_1 K_2), \tag{36}$$

where C_{in}, C_{out} represent the number of input and output channels, W and H are the width and height of the input feature map, and K_1, K_2 denote the size of convolution kernels. In our proposed U-ISTANet and U-ISTANet-L, each phase contains 6 convolutional layers, 4 of which have input and output channels of 32, i.e., $C_{in} = C_{out} = 32$ and the remaining two are $C_{in} = 2, C_{out} = 32$ and $C_{in} = 32, C_{out} = 2$, respectively. In addition, the input feature map size is $W \times H = N_s \times N_{ES}$ and the convolution kernel size is $K_1 \times K_2 = 3 \times 3$. Thus, the complexity of the CNNs in the network can be expressed as

$$\mathcal{C}_{CNN} = \mathcal{O}(4 \cdot 9 \cdot 32^2 \cdot N_p N_s N_{ES}). \tag{37}$$

Hence the total time complexity of U-ISTANet and U-ISTANet-L is

$$\begin{aligned} \mathcal{C}_{total} &= \mathcal{C}_M + \mathcal{C}_{CNN} \\ &= \mathcal{O}(N_p(N_s N_{ES})^2 + 4 \cdot 9 \cdot 32^2 \cdot N_p N_s N_{ES}). \end{aligned} \tag{38}$$

2) Communication Overhead Analysis: Next, we analyze the communication overhead for FU-ISTANet-L. Generally, the communication overhead can be expressed as

$$\mathcal{T}_{FL} = PTK, \tag{39}$$

where P is the number of trainable parameters, T denotes the number of communication rounds and K is the number of edge devices participating in FL. Actually, $P \gg TK$, it can be seen that the transmission overhead depends primarily on the number of trainable parameters of the network model in FL. In our proposed U-ISTANet-L, the parameters mainly come from the trainable linear matrix $\mathbf{W}_{1,3} \triangleq \{\mathbf{W}_1, \mathbf{W}_3\}$ and CNN, where the matrix is the main part of the total overhead. For a trainable matrix, the number of parameters is related to the input CSI dimension $2N_s N_{ES}$ and the compression dimension N_{cr}, it can be written as

$$\mathcal{T}_{\mathbf{W}_{1,3}} = 4 N_{cr} N_s N_{ES}. \tag{40}$$

While the number of parameters for CNN is only related to the setup of the network structure of our proposed U-ISTANet-L, where the complexity is

$$\begin{aligned} \mathcal{T}_{CNN} &= \sum \{C_{in} C_{out} K_1 K_2\}_i \\ &= 5 \cdot (4 \cdot (32 \cdot 32 \cdot 3 \cdot 3) + 2 \cdot (2 \cdot 32 \cdot 3 \cdot 3)) \\ &= 190080. \end{aligned} \tag{41}$$

Therefore, the total communication overhead for our proposed F-U-ISTANet-L is

$$\mathcal{T}_{FL} = (4 N_{cr} N_s N_{ES} + \mathcal{T}_{CNN}) TK. \tag{42}$$

However, the communication overhead for centralized learning can be expressed as

$$\mathcal{T}_{CL} = 2 N_s N_{ES} |\mathcal{D}| K, \tag{43}$$

where $|\mathcal{D}|$ is the total number of feedback CSIs from all edge devices.

It can be seen that the communication overhead of CL increases with the size of the dataset, whereas communication overhead in federated learning remains almost constant across training scenarios. Let $R = \frac{\mathcal{T}_{CL}}{\mathcal{T}_{FL}} = \frac{|\mathcal{D}|}{(2N_{cr} + \frac{\mathcal{T}_{CNN}}{2N_s N_{ES}})T}$, larger R means more efficient communication for FU-ISTANet-L. As the dimension of CSI becomes larger, more data is usually needed for training a network, i.e., $|\mathcal{D}|$ needs to be increased with it. Conversely, it can be seen that the denominator component decreases as the dimension of CSI increases, i.e., R will be larger. Therefore, FL is more advantageous for CSI feedback with high dimensionality (larger $N_s N_{ES}$), large dataset training (larger $|\mathcal{D}|$), and high compression requirements (lower N_{cr}).

IV. Personalized Federated Unfolding Learning for CSI Feedback

In the case that the decentralized data is non-IID, as shown in Fig. 4, for edge devices on the same circle i.e. on the same black circle or red circle, their local CSI data is IID. However, the CSI data between the edge devices on the red and black circles have different statistical characteristics due to the distance, which means that their CSI is non-IID. This highly personalized and heterogenous feature lead the global model may not generalize well to the local data of each edge device, and the federated learning can not converge well and the accuracy of the results is significantly reduced. In this section, we propose P-FU-ISTANet-L to address this issue.

The main idea is to combine meta-learning with our proposed FU-ISTANet-L, and the goal is to learn the initialization parameters on a collection of tasks $\mathcal{T}$. To this end, a small number of samples can then be used to train a model with excellent performance.

We train the initialization parameters of global model $\mathbf{\Theta}_G$ on task set $\mathcal{T}$. Each edge device in the distributed edge network has a specific task $T \subseteq \mathcal{T}$, and the local data set is divided into support set $\mathcal{D}_T^S$ and query set $\mathcal{D}_T^Q$. Instead of focusing on the performance of $\mathbf{\Theta}_G$, we focus on the performance of the local model parameters $\mathbf{\Theta}_k$ updated from $\mathbf{\Theta}_G$. Therefore, one episode of training process is divided into two phases: internal update and external update. For the internal update, $\mathbf{\Theta}_k$ is trained and updated based on the support set $\mathcal{D}_T^S$ from $\mathbf{\Theta}_G$. Then $\mathbf{\Theta}_k$ is assessed on the query set $\mathcal{D}_T^Q$ and $\mathbf{\Theta}_G$ are updated by calculating the loss in the external update.

The specific steps of the proposed P-FU-ISTANet-L are described in **Algorithm 2**. In our scheme, each edge device carries out local training for internal updates and the edge server receives all the losses sent by the edge devices for external updates. The detailed training between the edge devices and the edge server is shown as follows:

Algorithm 2 Our Proposed P-FU-ISTANet-L

Require: Edge devices prepare the dataset $\mathcal{D}_k$.
Output: Personalized network model parameters $\mathbf{\Theta}_k$.

Initialization: Edge server initializes $\mathbf{\Theta}_{G,0}$ and the learning rate α, β.
for each episode $t = 1, 2, ...$ **do**
  Edge server selects a set U_t of m edge devices and broadcasts $\mathbf{\Theta}_{G,t-1}$ to them
  for each edge device $k \in U_t$ in parallel **do**
    Sample the dataset $\mathcal{D}_k^Q, \mathcal{D}_k^S$ for MAML from $\mathcal{D}_k$.
    $\mathbf{\Theta}_{k,t}(0) = \mathbf{\Theta}_{G,t-1}$
    for each epoch $e = 1, 2, ...$ **do**
      $\mathcal{L}_{\mathcal{D}_k^S}(\mathbf{\Theta}_{k,t}(e-1)) = \frac{1}{|\mathcal{D}_k^S|}\sum_{\mathcal{D}_k^S}\mathcal{L}(\mathbf{\Theta}_{k,t}(e-1))$
      $\mathbf{\Theta}_{k,t}^{'}(e) = \mathbf{\Theta}_{k,t}(e-1) - \alpha\nabla\mathcal{L}_{\mathcal{D}_k^S}(\mathbf{\Theta}_{k,t}(e-1))$
      $\mathcal{L}_{\mathcal{D}_k^Q}(\mathbf{\Theta}_{k,t}^{'}(e)) = \frac{1}{|\mathcal{D}_k^Q|}\sum_{\mathcal{D}_k^Q}\mathcal{L}(\mathbf{\Theta}_{k,t}^{'}(e)$
      $\mathbf{\Theta}_{k,t}(e) = \mathbf{\Theta}_{k,t}(e-1) - \beta\nabla_{\mathbf{\Theta}_{k,t}(e-1)}\mathcal{L}_{\mathcal{D}_k^S}(\mathbf{\Theta}_{k,t}^{'}(e-1))$
    EndFor
    $g_k = \mathbf{\Theta}_{k,t}(0) - \mathbf{\Theta}_{k,t}(e)$
  EndFor
  $\mathbf{\Theta}_{G,t} = \mathbf{\Theta}_{G,t-1} - \frac{1}{\sum_{k\in U_t}|\mathcal{D}_k|}\sum_{k\in U_t}|\mathcal{D}_k|g_k$
EndFor
Edge sever broadcast the final $\mathbf{\Theta}_{G,t}$ to all edge devices for fine-tuning.
for each edge device in parallel **do**
  $\mathbf{\Theta}_k = \mathbf{\Theta}_{G,t}$
  for each epoch $e = 1, 2, ...$ **do**
    $\mathcal{L}_{\mathcal{D}_k}(\mathbf{\Theta}_k) = \frac{1}{|\mathcal{D}_k|}\sum_{\mathcal{D}_k^S} l(\mathbf{\Theta}_k)$
    $\mathbf{\Theta}_k = \mathbf{\Theta}_k - \alpha\nabla\mathcal{L}_{\mathcal{D}_k}(\mathbf{\Theta}_k)$
  EndFor
EndFor
return result

- The edge server selects a group of edge devices to participate in the training, which is denoted as U_t. Furthermore, the edge server initializes the global model $\mathbf{\Theta}_{G,0}$ parameters to broadcast to the selected edge devices. Then, in the t-th the communication round, the edge device receives the global model and uses $\mathbf{\Theta}_{G,t-1}$ as the initial parameters $\mathbf{\Theta}_{k,t-1}(0)$ for training. Each edge device randomly samples data from its own dataset $\mathcal{D}_k$ as support sets $\mathcal{D}_k^S$ and query sets $\mathcal{D}_k^Q$ for local model parameter updating and performance evaluating. For updating the local model parameters, the local parameters are trained on the support set $\mathcal{D}_k^S$, and loss in e-th epoch can be expressed as

$$\mathcal{L}_{\mathcal{D}_k^S}(\mathbf{\Theta}_{k,t}(e)) = \frac{1}{|\mathcal{D}_k^S|}\sum_{\mathcal{D}_k^S}\mathcal{L}(\mathbf{\Theta}_{k,t}(e-1)). \quad (44)$$

For internal updates, the edge device updates the parameters according to the gradient of the loss function obtained under the support set $\mathcal{D}_k^S$ updates the parameters, which can be expressed as

$$\mathbf{\Theta}_{k,t}^{'}(e) = \mathbf{\Theta}_{k,t}(e-1) - \alpha\nabla\mathcal{L}_{\mathcal{D}_k^S}(\mathbf{\Theta}_{k,t}(e-1)), \quad (45)$$

where α is the learning rate. When the training is complete or finished, the model $\mathbf{\Theta}_{k,t}^{'}(e)$ is evaluated using the respective query set $\mathcal{D}_k^Q$, i.e., the following loss value is calculated under the query set:

$$\mathcal{L}_{\mathcal{D}_k^Q}(\mathbf{\Theta}_{k,t}^{'}(e)) = \frac{1}{|\mathcal{D}_k^Q|}\sum_{\mathcal{D}_k^Q}\mathcal{L}(\mathbf{\Theta}_{k,t}^{'}(e)). \quad (46)$$

This loss represents the performance of that can be obtained by using $\mathbf{\Theta}_{k,t-1}(e-1)$ as the initial parameter. We need to obtain its gradient with respect to $\mathbf{\Theta}_{k,t-1}^{'}(e)$ to perform external updates, which is

$$\mathbf{\Theta}_{k,t}(e) = \mathbf{\Theta}_{k,t}(e-1) - \beta\nabla_{\mathbf{\Theta}_{k,t}(e-1)}\mathcal{L}_{\mathcal{D}_k^S}(\mathbf{\Theta}_{k,t}^{'}(e-1)), \quad (47)$$

where β is the learning rate. Following this, the selected U_t edge devices send the calculated gradient g_k to the edge server.

- At the edge server side, based on the received gradient of the loss function, the edge server starts to update the global model. Similarly, the update is given by

$$\mathbf{\Theta}_{G,t} = \mathbf{\Theta}_{G,t-1} - \frac{1}{\sum_{k\in U_t}|\mathcal{D}_k|}\sum_{k\in U_t}|\mathcal{D}_k|g_k, \quad (48)$$

When all the communication rounds are completed, all the edge devices get the final global model. Further, the edge devices can use the global model as the initial parameters to train a personalized model $\mathbf{\Theta}_k$ with excellent performance fast for a new task.

V. Simulation Results

This section begins with a description of the dataset generation process and the experimental setup parameters. Subsequently, we present the performance of the proposed neural network model in the context of centralized learning and compare it with existing networks. In addition, performance of channel feedback in the case of federation learning is also illustrated. Finally, the performance of the proposed personalized federated learning in the presence of data heterogeneity is shown.

A. Dataset Generation and Parameters Setting

Without loss of generality, we use the COST2100 channel model dataset adopted in [12], which is the dataset commonly used for channel feedback simulation experiments, with the following generation settings. It contains data from two sets of scenarios: the indoor picocellular scenario at the 5.3 GHz band, and the outdoor rural scenario at the 300 MHz band. A ULA with $N_{ES} = 32$ and N_c = 1024 is deployed at the edge device. The pre-processing of the channel matrix, i.e. sparse transformation of the angular delay domain, is done. At this point, only the first 32 rows of the transformed channel matrix

TABLE I: Parameters setting about CDL CSI generation

Parameters	Value
Number of BS transmitting antennas	32
Number of UE receiving antennas	1
Maximum number of delays	32
Center frequency	4GHz
Subcarrier spacing	30kHz
Training set size	20000
Validation set size	5000
Test set size	4000

are retained. This means that the size of $\tilde{\mathbf{H}}_k$ is 32×32. The training, validation and test sets contain 100,000, 30,000 and 20,000 samples, respectively.

In order to better illustrate the proposed algorithms in a non-iid enviroment, we refer to the NR CDL channel model defined in TR38.901 of 3GPP R15 [38], and use the NR CDL channel model of 5G toolbox in MATLAB to generate CSI dataset in different scenarios. The datasets are named as CDL-A, CDL-B, CDL-C, CDL-D, CDL-E, and CDL-A2. These datasets can better represent data heterogeneity. The main common system parameters are described in Table I. It should be emphasized that the DelayProfile is different for each dataset, corresponding to their names, i.e., CDL-A, CDL-B, CDL-C, CDL-D, and CDL-E. CDL-A, CDL-B, and CDL-C represent three different channel profiles for NLOS while CDL-D and CDL-E are constructed for LOS. In particular, the speed of UE is set to 5.4km/h in dataset CDL-A2, which is approximately the speed of walking. The devices in other dataset are stationary.

Initializing step size $\rho = 0.1$ and threshold $\theta = 0.1$. All experiments are conducted on an NVIDIA GeForce RTX 3090 GPU.

We use the NMSE as the metric to evaluate the performance of the model, which is given by

$$\text{NMSE} = E\left[\frac{\|\tilde{\mathbf{H}}_k - \hat{\mathbf{H}}_k\|_2^2}{\|\tilde{\mathbf{H}}_k\|_2^2}\right]. \tag{49}$$

In order to evaluate the performance of our proposed networks, the following learning frameworks are considered:

- Centralized Learning: the edge devices send all their local data to the edge server, for training the model centrally.
- IID Federated Learning: when the data from edge devices is IID, all devices and edge servers train the model by federated learning. The indoor or outdoor data is considered and our proposed FU-ISTANet-L can be used for CSI feedback in this case.
- Non-IID federated learning: data is non-IID due to different environments of edge devices and distances to edge servers. The data contains indoor and outdoor data. The proposed P-FU-ISTANet-L is designed to provide feedback on CSI of different devices.

B. Performance Evaluation of Our Proposed Networks

In this subsection, we evaluate the performance of the proposed networks in terms of convergence, network parameter size, compression capability, and recovery accuracy.

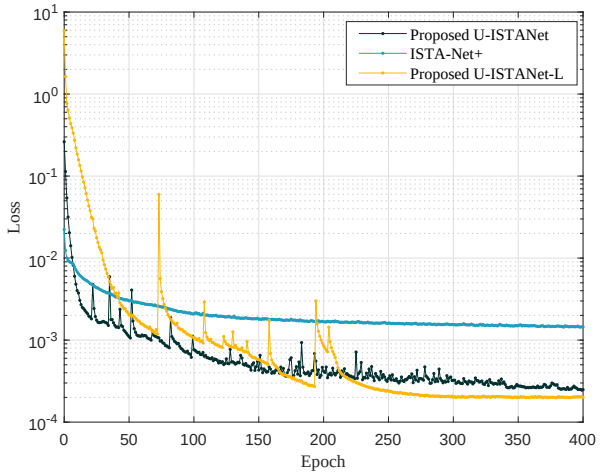
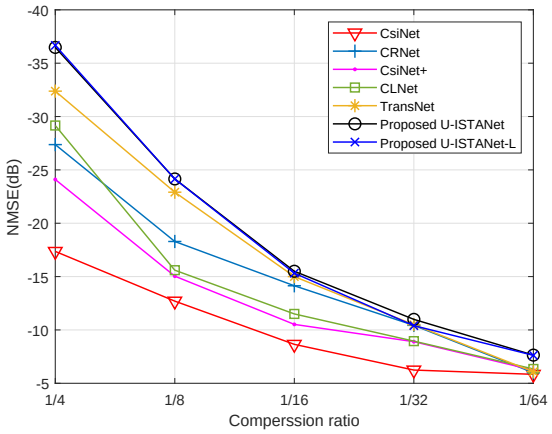


Fig. 5: Validate loss trends of proposed U-ISTANet, U-ISTANet-L and ISTA-NET+, where the compression ratio =1/4 and the indoor data are considered.

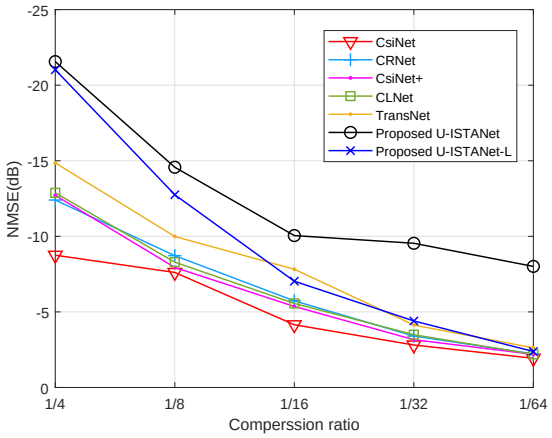
1) Our Proposed End-to-End U-ISTANet: In Fig. 5, we show the NMSE trends for ISTA-NET+, the proposed U-ISTANet, U-ISTANet-L under the validation set within 400 training epochs. In order to obtain better performance for each model individually, we train these models using different learning rates. For the ISTA-net+, we use a constant learning rate of 0.0005. Then, for the U-ISTANet, we set the learning rate as 0.0005 for the first 300 epochs, and reduced it to 0.0001 thereafter. For the U-ISTANet-L, we use the warmup learning rate scheme. The first 100 epochs of training were set as the warmup phase. In this phase, the learning rate started from a small initial value of 0.0001 and gradually increases linearly to a predetermined maximum value of 0.0005. After the warmup phase, we employ a quadratic polynomial decay strategy to finally reduce the learning rate to 0.0001.

We can see that initially the NMSE performance of ISTA-Net is better than that of our proposed U-ISTANet, but at the same maximum training epoch, the NMSE of the proposed U-ISTANet and U-ISTANet-L are all significantly lower than the lowest NMSE of ISTA-Net+. The encoder compression of ISTA-Net+ uses an artificially designed compression matrix and still uses this matrix for the decoder part, it has a significant performance in the early stages according to the theory of compressed sensing. However, since these parameters are not trainable, the performance of ISTA-Net+ cannot be improved further in the later stages of training. Our proposed U-ISTANet and U-ISTANet-L compensate for this by using multiple learnable matrices.

To demonstrate the performance of our proposed U-ISTANet and U-ISTANet-L, it is compared with several existing deep learning models for CSI feedback, including CsiNet [12], CsiNet+ [14], CLNet [17], CRNet [16], and TransNet [19], under different test conditions. As shown in Fig. 6, we evaluate the NMSE performance of each model for compressing both indoor and outdoor data at different compression rates. As can be seen from the Fig. 6 (a), our



(a) indoor data



(b) outdoor data

Fig. 6: NMSE at different compression ratios under different methods and datasets.

models U-ISTANet and U-ISTANet-L achieve better NMSE performance at different compression rates for indoor CSI. Refer to Fig. 6 (b), U-ISTANet-L also achieved satisfactory results for outdoor CSI. Furthermore, the U-ISTANet always maintains the best NMSE performance. Especially, as shown in Fig. 6 (b), the NMSE performance of U-ISTANet is far ahead of other solutions for the outdoor data set, achieving very impressive results.

Table I gives the specific NMSE (dB) and the number of parameters the model can be trained with different compression rates and scenarios. The best two results are marked in bold. As can be seen from the table, the proposed U-ISTANet has a good NMSE performance for all compression rates, both indoors and outdoors, and U-ISTANet-L is a close second. This indicates that the network designed based on CS theory has considerable advantages in terms of sparse recovery. In terms of the number of trainable parameters, our proposed U-ISTANet, does not perform well, and it has the largest number of parameters. The reason for this is that the dimension of $\mathbf{W}_2$ in our design of learnable parameters is 2048×2048, and it does not decrease by decreasing the compression rate. But U-ISTANet-L improves this problem, making it close to

TABLE II: NMSE(dB) AND THE COMPLEXITY COMPARISON IN DIFFERENT COMPRESSION RATIO AND SCENARIOS

CR	Methods	NMSE		Trainable Parameters
		Indoor (dB)	Outdoor (dB)	
1/4	CsiNet	-17.36	-8.75	2.10M
	CsiNet+	-27.37	-12.4	2.12M
	CRNet	-24.10	-12.71	**2.09M**
	CLNet	-29.16	-12.88	2.10M
	TransNet	-32.38	-14.86	2.64M
	U-ISTANet	**-36.48**	**-21.56**	6.48M
	U-ISTANet-L	**-37.37**	**-21.02**	2.28M
1/8	CsiNet	-12.7	-7.61	1.05M
	CsiNet+	-18.29	-8.72	1.07M
	CRNet	-15.04	-7.94	1.05M
	CLNet	-15.60	-8.29	1.05M
	TransNet	-22.91	-9.99	1.60M
	U-ISTANet	**-24.15**	**-14.57**	5.43M
	U-ISTANet-L	**-24.15**	**-12.75**	1.24M
1/16	CsiNet	-8.65	-4.15	0.53M
	CsiNet+	-14.14	-5.73	0.55M
	CRNet	-10.52	-5.36	0.53M
	CLNet	-11.15	-5.56	0.53M
	TransNet	-15.00	-7.82	1.07M
	U-ISTANet	**-15.50**	**-10.05**	4.91M
	U-ISTANet-L	**-15.32**	-7.20	0.71M
1/32	CsiNet	-6.24	-2.81	0.27M
	CsiNet+	-10.43	-3.4	0.29M
	CRNet	-8.90	-3.16	**0.26M**
	CLNet	-8.95	-3.49	0.27M
	TransNet	-10.49	-4.13	0.81M
	U-ISTANet	**-10.99**	**-9.54**	4.65M
	U-ISTANet-L	**-10.76**	**-4.40**	0.45M
1/64	CsiNet	-5.84	-1.93	0.14M
	CsiNet+	-5.99	-2.22	0.16M
	CRNet	-6.23	-2.19	**0.13M**
	CLNet	-6.34	-2.19	0.14M
	TransNet	-6.08	-2.62	0.68M
	U-ISTANet	**-8.00**	**-8.01**	4.51M
	U-ISTANet-L	**-7.62**	-2.39	0.32M

TABLE III: Training Parameters for FU-ISTANet-L

Parameters	Value
Learning rate α	0.0002
Batch size	100
Local training epochs	10
Communication rounds	40, 60
Device participation rate	0.8

the number of parameters of other lightweight models without reducing performance too much.

2) Our Proposed FU-ISTANet-L: We further evaluate the performance of the proposed FU-ISTANet-L, which is deployed in the federated learning framework for training in the same CSI environment and explore the impact of the number of edge devices on model training with a constant dataset size. The parameters in the training are shown in the Table III.

Fig. 7 shows the convergent tendency of training the ISTANet+ federated learning framework and the proposed FU-ISTANet-L, respectively. Ten edge devices with i.i.d. indoor CSI data sets are considered, and their total training data is 100000 indoor CSI. In addition, we use the Fedavg as the aggregation algorithm. The local training epoch of each device is set as 10. It can be seen that both models converge relatively well, indicating that both models can learn CSI feedback effectively in the FL case. It can also be seen that

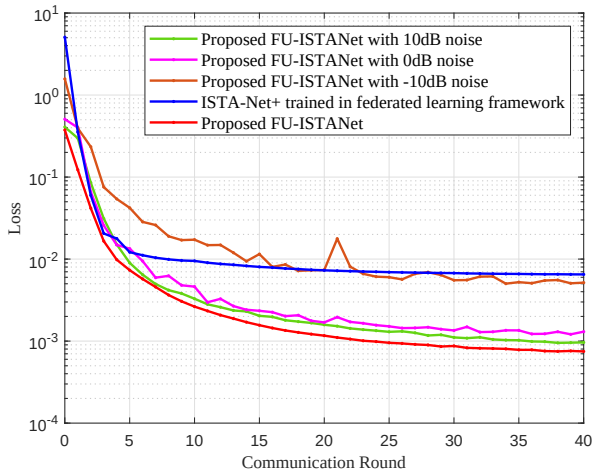


Fig. 7: Performance of CSI feedback in distributed edge networks with i.i.d. data.

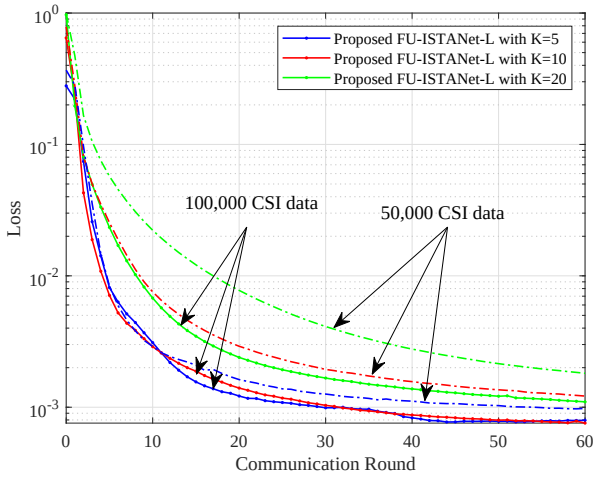


Fig. 8: Performance of CSI feedback in distributed edge networks with different numbers of edge devices containing i.i.d. data.

our model has a far better NMSE than ISTA-Net in terms of the final training trend. This further suggests that our proposed network can learn CSI feedback better and be more suitable for distributed deployment in a federated learning framework. In addition, Fig. 7 also demonstrates the performance of the distributed edge network for federated learning in Gaussian white noise environments with different SNRs. When SNR = 10dB, the convergence trend of FL remains almost unchanged and the final performance is very close to the case when training without noise. As the noise power increases, the convergence trend starts to become unstable at SNR=0dB, but the results are still satisfactory. However, when SNR=-10dB, the training effect of FL is seriously affected, the model cannot converge well, and its results are seriously deviated. it suggests that FU-ISTANet has considerable robustness to additive white Gaussian noise.

Fig. 8 illustrates the training loss trend of the proposed FU-ISTANet-L for varying numbers of edge devices with different sizes of training sets. The solid line represents training with 100000 CSI, while the dashed line represents 50000 CSI. We assume equal-sized training sets for each device when a specific number of devices is present. Specifically, when there is a total training set size of 100000, the number of edge devices $K = 5, 10, 20$. In these cases, each device owns 20000, 10000, and 5000 CSI respectively. Additionally, when the training set size reduces to 50000, the devices' datasets are 10000, 5000, and 2500 each. respectively. We set the local training epoch as 10, the number of communication rounds as 60, and the learning rate as 0.0002. And only 80% of the devices successfully communicate with the server in each communication round. As can be seen from the figure, our proposed FU-ISTANet-L converges to a considerable loss in all cases. When the total training set size is the same, the training loss gradually increases as the number of edge devices increases. This means that the training results worsen with the addition of more edge devices, but ultimately stabilize at a respectable level. Due to the constant total training set

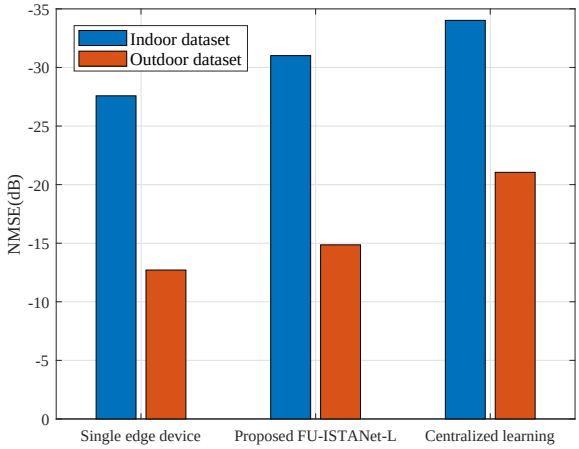
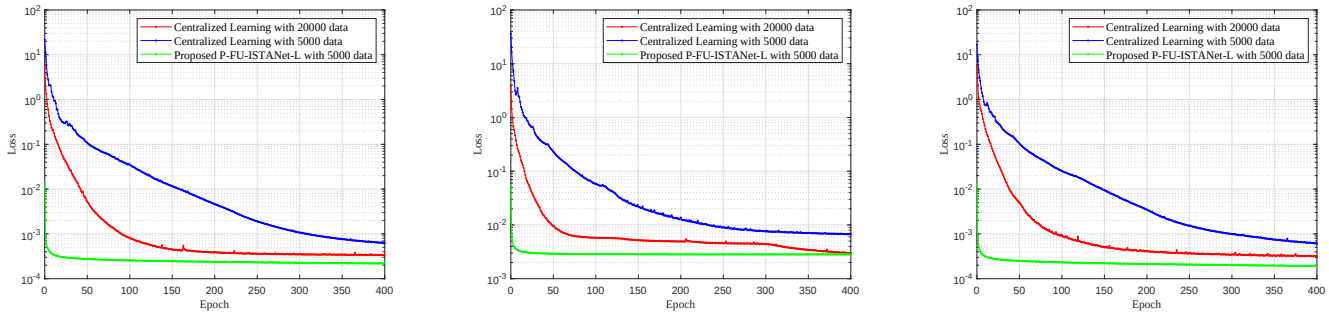


Fig. 9: The NMSE Performance of different methods and datasets, where each dataset has i.i.d. data.

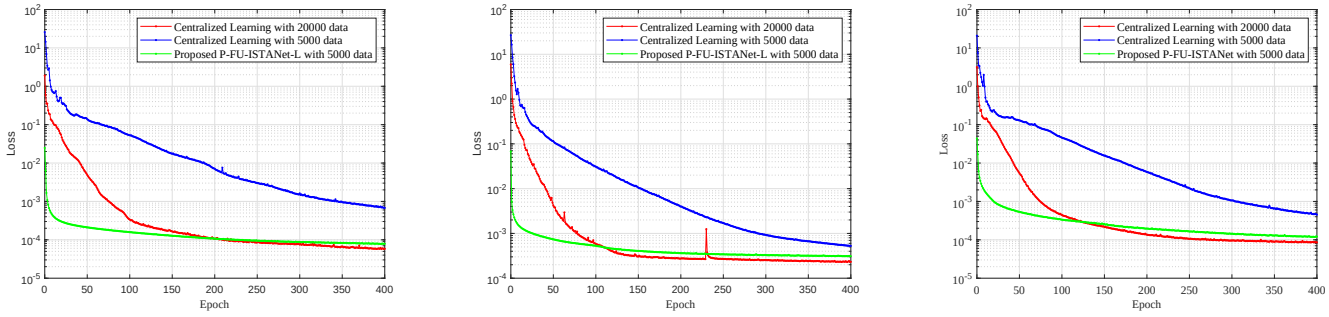
size, the local dataset size decreases as the number of devices increases, resulting in local training and aggregation being compromised. On the other hand, if we compare the outcomes of the proposed model for the same number of devices but with different-sized training sets, we can observe that increasing the size of local training sets causes a substantial decrease in training loss. This indicates that the proposed model effectively learns the data features, and increasing the dataset size can further enhance training outcomes without imposing additional communication burdens.

Fig. 9 shows the NMSE performance comparison for three training frameworks:

- **Single Edge Device**: only one edge device with 10,000 data and one edge server are considered. The proposed end-to-end U-ISTANet is adopted for CSI feedback from



(a) CDL-A. (b) CDL-B. (c) CDL-A2



(d) CDL-D. (e) CDL-C. (f) CDL-E.

Fig. 10: Loss trends of centralized learning and proposed P-FU-ISTANet-L with different datasets.

the edge device to the edge server.

- **Federated Learning**: ten edge devices and one edge server are considered. Each edge device has 10,000 data. All edge devices train one CSI feedback model with a federated learning framework. The proposed FU-ISTANet-L is considered.
- **Centralized Learning**: the edge server centralized trains the model, and the data set size is 100,000.

It can be seen from the figure that centralized learning is the most effective for both indoor and outdoor data, the proposed FU-ISTANet-L is the second most effective, and the one edge device approach with the smallest dataset is the least effective. This indicates that the federated learning framework can gain from large data through multi-edge devices collaboration, but its performance cannot be better than that of centralized learning trained with the same data size.

3) Our Proposed P-FU-ISTANet-L: Finally, we conduct experiments of the proposed P-FU-ISTANet-L. We use CDL-A, CDL-B, CDL-D and CDL-A2 train P-FU-ISTANet then fine-tuning with a small amount of data from 5,000 corresponding scenarios. For centralised learning, we simulated training the network with the full local 20,000 data as well as training the network with a small amount of 5,000 data. Table IV presents the relevant parameters used during the training process, while the default values are adopted for the parameters that are not specified. Furthermore, we evaluate the performance of FU-ISTANet-L and cluster FL [31] with non-iid datasets.

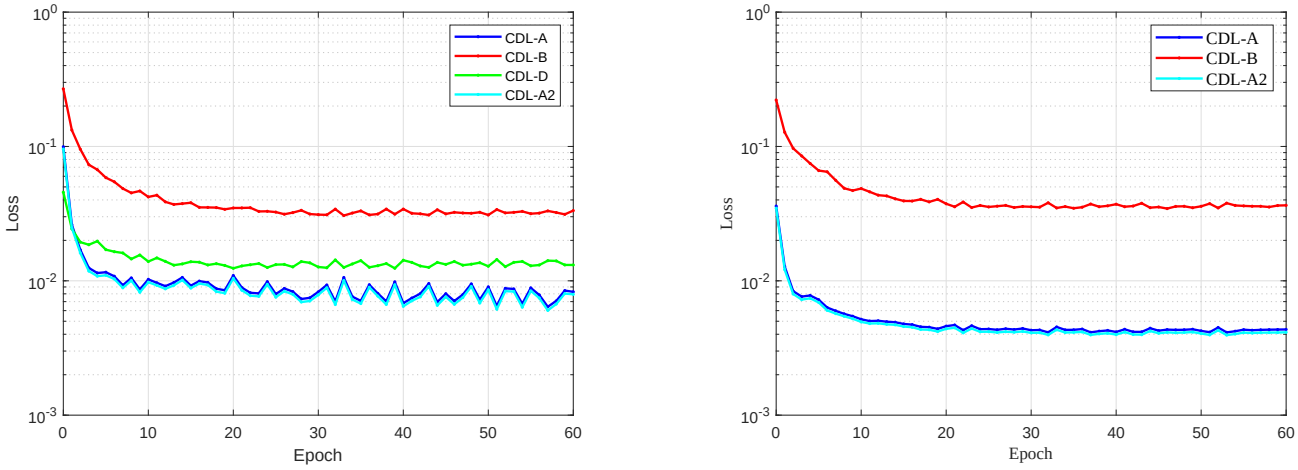
Fig. 10 shows the loss trend of different algorithms trained in different scenarios. Overall, from the final loss convergence position, the proposed P-FU-ISTANet with 5,000 data and Centralized learning with 20,000 data are similar and both

TABLE IV: Parameters setting about CDL CSI generation

Parameters	Value
Learning rate α	0.0004
Learning rate β	0.0002
Batch size	100
Local training epochs	10
Leaning episode	40
Learning rate in fine-tune	0.0002

are much smaller than Centralized learning with 5,000 data. For the latter, because only 5,000 pieces of data participate in training, it does not converge as fast or as fruitful as the former. In addition, it can be found that the proposed P-FU-ISTANet with 5,000 data can converge quickly in all scenarios. And in particular, due to the use of the datasets CDL-A, CDL-B, CDL-A2, and CDL-D together for the training of the initial P-FU-ISTANet model, as can be seen in Fig.10 (a), (b), (c) and (d), The proposed P-FU-ISTANet converges to values close to, or even better than, those of centralized learning with 20,000 data after a very few epoch fine-tuning. Not only that, for the CDL-C, and CDL-E scenarios, where the network model was never exposed to these data before the fine-tuning, P-FU-ISTANet can still make the loss converge quickly to a level close to that of the centralised learning with 20,000 data with a small amount of data fine-tuning. This means that even if the channel environment is transformed and the degree of data non-iid is large, our proposed P-FU-ISTANet can obtain a high-performance personalised model with a small training cost and dataset after federated learning.

Fig. 11 shows the loss trend of FU-ISTANet-L and Cluster FL with non-iid data when trained with non-iid data. They are



(a) FU-ISTANet-L

(b) Cluster FL with U-ISTANet-L

Fig. 11: Performance of FU-ISTANet-L and Cluster FL with non-iid data

TABLE V: The proximity matrix of four datasets

Algorithm	CDL-A	CDL-B	CDL-D	CDL-A2	CDL-C	CDL-E
FU-ISTANet-L	-20.30	-14.87	-17.82	-20.33	\	\
Cluster FL	-23.27	-14.13	-41.98	-23.32	\	\
Centralized learning with 5,000 data	-32.01	-20.56	-31.10	-31.63	-31.57	-32.28
Centralized learning with 20,000 data	-34.70	-23.61	**-41.98**	-34.40	**-35.10**	**-40.39**
P-FU-ISTANet-L fine-tune with 5,000 data	**-36.65**	**-23.74**	-40.54	**-36.71**	-33.46	-38.92

also trained with four devices and their dataset are CDL-A, CDL-B, CDL-D and CDL-A2. In cluster FL, by analyzing the similarity of the datasets, the algorithm places CDL-A, CDL-B, and CDL-A2 in a cluster, and CDL-A2 trained separately as a separate cluster alone, and the proximity matrix of four datasets is shown in the Table VI whose entries are the exact principal angles between each pairs of these datasets' subspaces. This division is justified in terms of the final results of the clustering, since CDL-A, CDL-B and CDL-A2 are NLOS channel model and CDL-D is LOS channel models. Fig. 11 (a), it can be noticed that the model never converges stably when FL is completed directly using the four datasets, which is exactly the problem caused by data heterogeneity. Comparing with cluster FL in Fig. 11 (b), it can be found that the loss gradually tends to be stable, indicating that cluster FL can effectively improve the convergence problem brought by heterogeneity, but its final value of the loss does not significantly decrease.

TABLE VI: The proximity matrix of four datasets

Dataset	CDL-A	CDL-B	CDL-D	CDL-A2
CDL-A	0	0.02798	2.29508	0.04423
CDL-B	0.02798	0	2.34268	0.06561
CDL-D	2.29508	2.34268	0	2.2879
CDL-A2	0.04423	0.06561	2.2879	0

Table V give the final NMSE performance of various algorithms in a heterogeneous environment, and optimal values are shown in bold. As can be learned from the table, the results of proposed FU-ISTANet-L as well as cluster FL are far from the optimum. In contrast, the NMSE performance of our proposed P-FU-ISTANet can approach the centralized learning using only a few amount of data fine-tuning. Even though these data networks have never been learned before, Since, the datasets CDL-C and CDL-E did not participate in training prior to fine-tuning. Furthermore, in CDL-A, CDL-B and CDL-A2, P-FU-ISTA outperforms the centralised learning, this suggests that P-FU-ISTANet can learn common features in the LOS environment and thus gain.

VI. CONCLUSIONS

In this paper, we first proposed an end-to-end U-ISTANet and its lightweight version U-ISTANet for CSI feedback, which uses the deep learning method to determine the compression matrix and sparse variation without having to choose too many hyperparameters. The proposed network combines the advantages of compression-awareness and neural networks to perform fast and accurate CSI compression and recovery. We then extended the proposed end-to-end U-ISTANet-L to FU-ISTANet-L with the federated learning framework, which can obtain a model with better performance by enlarging the data set through distributed learning. In particular, each edge device trains a local model using a small local dataset and sends it to the edge device for aggregation of model parameters. The proposed framework reduces the transmission overhead and increases the training speed while obtaining close to the performance of centralized learning. Finally, we proposed a P-FU-ISTANet-L with personalized federation learning to solve the training problem in different wireless channel environments. Specially, a pre-trained model was obtained and each edge device fine-tunes the model using

only a small amount of data to obtain a personalized model suitable for the local channel environment. The experimental results demonstrated that the NMSE performance of the proposed networks outperforms the existing models. Moreover, the proposed federation learning and personalized federation learning frameworks can be well applied to CSI feedback.

REFERENCES

[1] E. Bjrnson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: ten myths and one critical question," *IEEE Commun. Magazine*, vol. 54, no. 2, pp. 114–123, 2016.

[2] P. Yang, Y. Xiao, M. Xiao, and S. Li, "6G wireless communications: Vision and potential techniques," *IEEE Network*, vol. 33, no. 4, pp. 70–75, 2019.

[3] F. A. Pereira de Figueiredo, "An overview of massive MIMO for 5G and 6G," *IEEE Latin America Transactions*, vol. 20, no. 6, pp. 931–940, 2022.

[4] C. Tan, D. Cai, Y. Xu, Z. Ding, and P. Fan, "Threshold-enhanced hierarchical spatial non-stationary channel estimation for uplink massive MIMO systems," *IEEE Trans. Wireless Communications*, pp. 1–1, 2023.

[5] S. Qiu, D. Chen, D. Qu, K. Luo, and T. Jiang, "Downlink precoding with mixed statistical and imperfect instantaneous CSI for massive MIMO systems," *IEEE Trans. Vehicular Technology*, vol. 67, no. 4, pp. 3028–3041, 2018.

[6] F. Fang, Y. Xu, Z. Ding, C. Shen, M. Peng, and G. K. Karagiannidis, "Optimal resource allocation for delay minimization in NOMA-MEC networks," *IEEE Transactions on Communications*, vol. 68, no. 12, pp. 7867–7881, 2020.

[7] J. Hoydis, C. Hoek, T. Wild, and S. ten Brink, "Channel measurements for large antenna arrays," in *2012 International Symposium on Wireless Communication Systems (ISWCS)*, 2012, pp. 811–815.

[8] M. E. Eltayeb, T. Y. Al-Naffouri, and H. R. Bahrami, "Compressive sensing for feedback reduction in MIMO broadcast channels," *IEEE Trans. Commun.*, vol. 62, no. 9, pp. 3209–3222, 2014.

[9] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Overview of deep learning-based CSI feedback in massive MIMO systems," *IEEE Trans. Commun.*, vol. 70, no. 12, pp. 8017–8045, 2022.

[10] S. Tang, J. Xia, L. Fan, X. Lei, W. Xu, and A. Nallanathan, "Dilated convolution based CSI feedback compression for massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 11 216–11 221, 2022.

[11] Z. Wei, H. Liu, B. Li, and C. Zhao, "Joint massive MIMO CSI estimation and feedback via randomized low-rank approximation," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7979–7984, 2022.

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

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

[14] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Convolutional neural network-based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2827–2840, 2020.

[15] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 449–458.

[16] Z. Lu, J. Wang, and J. Song, "Multi-resolution CSI feedback with deep learning in massive MIMO system," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[17] S. Ji and M. Li, "CLnet: Complex input lightweight neural network designed for massive MIMO CSI feedback," *IEEE Wireless Commun. Letters*, vol. 10, no. 10, pp. 2318–2322, 2021.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000–6010.

[19] Y. Cui, A. Guo, and C. Song, "TransNet: Full attention network for CSI feedback in FDD massive MIMO system," *IEEE Wireless Commun. Letters*, vol. 11, no. 5, pp. 903–907, 2022.

[20] Y. Yang, F. Gao, G. Y. Li, and M. Jian, "Deep learning-based downlink channel prediction for FDD massive MIMO system," *IEEE Commun. Letters*, vol. 23, no. 11, pp. 1994–1998, 2019.

[21] J. Zeng, J. Sun, G. Gui, B. Adebisi, T. Ohtsuki, H. Gacanin, and H. Sari, "Downlink CSI feedback algorithm with deep transfer learning for FDD massive MIMO systems," *IEEE Trans. Cognitive Commun. and Networking*, vol. 7, no. 4, pp. 1253–1265, 2021.

[22] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[23] Y. Cui, J. Guo, X. Li, L. Liang, and S. Jin, "Federated edge learning for the wireless physical layer: Opportunities and challenges," *China Communications*, vol. 19, no. 8, pp. 15–30, 2022.

[24] A. M. Elbir and S. Coleri, "Federated learning for channel estimation in conventional and RIS-assisted massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4255–4268, 2022.

[25] M. M. Wadu, S. Samarakoon, and M. Bennis, "Federated learning under channel uncertainty: Joint client scheduling and resource allocation," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[26] A. M. Elbir and S. Coleri, "Federated learning for hybrid beamforming in mm-wave massive MIMO," *IEEE Commun. Letters*, vol. 24, no. 12, pp. 2795–2799, 2020.

[27] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," 2021. [Online]. Available: https://openreview.net/forum?id=g0a-XYjpQ7r

[28] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," 2021, publisher Copyright: © 2021 ICLR 2021 - 9th International Conference on Learning Representations. All rights reserved.; 9th International Conference on Learning Representations, ICLR 2021 ; Conference date: 03-05-2021 Through 07-05-2021.

[29] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *IEEE Transactions on Information Theory*, vol. 68, no. 12, pp. 8076–8091, 2022.

[30] F. Sattler, K.-R. Mller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3710–3722, 2021.

[31] S. Vahidian, M. Morafah, W. Wang, V. Kungurtsev, C. Chen, M. Shah, and B. Lin, "Efficient distribution similarity identification in clustered federated learning via principal angles between client data subspaces," in *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. [Online]. Available: https://doi.org/10.1609/aaai.v37i8.26197

[32] J. Scarlett, R. Heckel, M. R. D. Rodrigues, P. Hand, and Y. C. Eldar, "Theoretical perspectives on deep learning methods in inverse problems," *IEEE Journal on Selected Areas in Information Theory*, pp. 1–1, 2023.

[33] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, pp. 399–406.

[34] J. Liu, X. Chen, Z. Wang, and W. Yin, "ALISTA: Analytic weights are as good as learned weights in LISTA," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=B1lnzn0ctQ

[35] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1828–1837.

[36] Y. Wang, X. Chen, H. Yin, and W. Wang, "Learnable sparse transformation-based massive MIMO CSI recovery network," *IEEE Communications Letters*, vol. 24, no. 7, pp. 1468–1471, 2020.

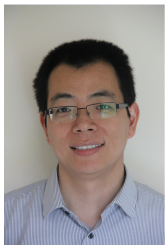
[37] L. Liu, C. Oestges, J. Poutanen, K. Haneda, P. Vainikainen, F. Quitin, F. Tufvesson, and P. D. Doncker, "The COST 2100 MIMO channel model," *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 92–99, 2012.

[38] 3GPP TR 38.901, "Study on channel model for frequencies from 0.5 to 100 GHz," Tech. Rep.

Chongyang Tan (S'22) received the B.S. degree from the Henan University of Technology, Zhengzhou, China, in 2021. He is currently pursuing the M.S. degree in cyberspace security at the College of Information Science and Technology of Jinan University. His research interests include signal processing algorithms designs for massive MIMO systems, deep learning for massive MIMO systems and physical layer security.

Donghong Cai (M'20) received the B.S. degree from the School of Mathematics and Information Sciences, Shaoguan University, Shaoguan, China, in 2012, and the M.S. and Ph.D. degrees from Southwest Jiaotong University, Chengdu, China, in 2015 and 2020, respectively. From October 2017 to October 2018, he was a visiting Ph.D. student with Lancaster University, Lancaster, U.K., and the University of Manchester, Manchester, U.K. He served as a Guest Editor for a special issues in Physical Communication. He is currently a Lecturer with the College of Information Science and Technology/College of Cyber Security, Jinan University, Guangzhou, China. His current research interests include signal detection, security coding, distributed Internet of Things, privacy preservation, machine learning, and nonorthogonal multiple access.

Fang Fang (M'18-SM'23) is currently an Assistant Professor in the Department of Electrical and Computer Engineering and the Department of Computer Science, Western University, Canada. She received the Ph.D. degree in electrical engineering from the University of British Columbia (UBC), Canada. Her current research interests include machine learning for intelligent wireless communications, non-orthogonal multiple access (NOMA), reconfigurable intelligent surface (RIS), multi-access edge computing (MEC), Semantic Communications and Edge AI, etc. Dr. Fang has been serving as a general chair for EAI GameNets 2022, a publications chair for IEEE VTC 2023 Fall, and a symposium chair for IEEE Globecom 2023, IEEE VTC 2024 Spring and IEEE ICC 2025. Dr. Fang currently serves as an Editor for IEEE Communication Letters and IEEE Open Journal of the Communications Society (OJ-COM). She received the Exemplary Reviewer Certificates from the IEEE Transactions on Communications in 2017 and 2021, as well as the Exemplary Editor Certificates from IEEE OJ-COM in 2021 and 2023. Dr. Fang won the IEEE SPCC Early Achievement Award in 2023.



Zhiguo Ding (S'03-M'05-F'20) received his B.Eng from the Beijing University of Posts and Telecommunications in 2000, and the Ph.D degree from Imperial College London in 2005. He is currently a Professor in Communications at Khalifa University, and has also been affiliated with the University of Manchester and Princeton University.

Dr Ding's research interests are B5G networks, game theory, cooperative and energy harvesting networks and statistical signal processing. He is serving as an Area Editor for *IEEE Transactions on Wireless Communications*, and *IEEE Open Journal of the Communications Society*, an Editor for *IEEE Transactions on Vehicular Technology* and *IEEE Communications Surveys & Tutorials*, and was an Editor for *IEEE Wireless Communication Letters*, *IEEE Transactions on Communications*, *IEEE Communication Letter*s from 2013 to 2016. He recently received the EU Marie Curie Fellowship 2012-2014, the Top IEEE TVT Editor 2017, IEEE Heinrich Hertz Award 2018, IEEE Jack Neubauer Memorial Award 2018, IEEE Best Signal Processing Letter Award 2018, Friedrich Wilhelm Bessel Research Award 2020, and IEEE SPCC Technical Recognition Award 2021. He is a Fellow of the IEEE, a Distinguished Lecturer of IEEE ComSoc, and a Web of Science Highly Cited Researcher in two categories 2022.

Pingzhi Fan (M'93-SM'99-F'15) received the M.Sc. degree in computer science from Southwest Jiaotong University, China, in 1987, and the Ph.D. degree in electronic engineering from Hull University, U.K., in 1994. He is currently presidential professor of Southwest Jiaotong University (SWJTU), honorary dean of the SWJTU-Leeds Joint School, and a visiting professor of Leeds University, UK (1997-). He served as an EXCOM member for the IEEE Region 10, IET (IEE) Council, and the IET Asia Pacific Region. He is a recipient of the UK ORS Award (1992), the National Science Fund for Distinguished Young Scholars (1998, NSFC), IEEE VT Society Jack Neubauer Memorial Award (2018), IEEE SP Society SPL Best Paper Award (2018), IEEE/CIC ICCC2020 Best Paper Award, IEEE WCSP2022 Best Paper Award, and IEEE ICC2023 Best Paper Award. He served as a chief scientist of the National 973 Plan Project (National Basic Research Program of China) between 2012.1-2016.12. His research interests include high mobility wireless communications, massive random-access techniques, signal design & coding, etc. He is an IEEE VTS Distinguished Speaker (2019-2025), a fellow of IEEE, IET, CIE and CIC.