

Model Splitting Enhanced Communication-Efficient Federated Learning for CSI Feedback

Yanjie Dong, Haijun Zhang, *Fellow, IEEE*, Gaojie Chen, *Senior Member, IEEE*, Xiaoyi Fan, Victor C. M. Leung, *Life Fellow, IEEE*, and Xiping Hu

Abstract—Recent advancements have introduced federated machine learning-based channel state information (CSI) compression before the user equipments (UEs) upload the downlink CSI to the base transceiver station (BTS). However, most existing algorithms impose a high communication overhead due to frequent parameter exchanges between UEs and BTS. In this work, we propose a model splitting approach with a shared model at the BTS and multiple local models at the UEs to reduce communication overhead. Moreover, we implant a pipeline module at the BTS to reduce training time. By limiting exchanges of boundary parameters during forward and backward passes, our algorithm can significantly reduce the exchanged parameters over the benchmarks during federated CSI feedback training.

Index Terms—Communication efficiency, CSI codebook learning, federated training, pipeline parallelism.

I. INTRODUCTION

The massive multi-input and multi-output (mMIMO) technology has been standardized in the fifth generation (5G) mobile communications [1], [2]. Equipped with a large-scale antenna array, an mMIMO system can effectively suppress multi-user interference and significantly enhance system capacity by harnessing spatial diversity and multiplexing gains. However, during the downlink period, the merits of the mMIMO system depend on accurate acquisition of channel state information (CSI) at the base transceiver station (BTS) [3]. While user equipments (UEs) can accurately estimate the downlink CSI [4], [5], they need to upload the CSI estimates to the BTS via the feedback channels in the frequency division duplex (FDD) mode. When the scale of the antenna array continuously increases, downlink CSI feedback can incur a burdensome overhead that may compromise the benefits of

mMIMO technology [6]. Therefore, the downlink CSI needs to be compressed before uploading to the BTS.

Three categories of compression algorithms have been proposed to reduce the CSI feedback overhead, i.e., codebook based algorithms [7], [8], compressive-sensing based algorithms [9], and deep-learning based algorithms [10], [11]. More specifically, with a shared codebook, the UEs only need to upload the indices of codewords to the BTS to recover the downlink CSI. However, the length of codewords linearly increases with the number of antennas. On the other hand, compressive-sensing based algorithms depend on the sparsity of downlink CSI, which may not hold when advanced multiplexing schemes are used [12]. Besides, the iterative procedures of compressive-sensing based algorithms also hinder the timeliness of acquiring downlink CSI at the BTS [10], [11] and the downstream tasks [13]–[15]. Recent advancements in deep learning have promoted the use of machine learning models for compressing downlink CSI [10], [11]. For example, in CsiNet [10], CLNet [11], and Transformers [16], [17], deep neural networks are trained before deployed at the UEs and the BTS. Moreover, the correlations of different data domains [18] and radio resource management based on limited CSI feedback [19], [20] are exploited in the context of deep learning based CSI feedback.

While deep-learning based approaches for compressing CSI feedback have received significant attention from both industry and academia, the previous deep neural networks and their associated training methodologies have been primarily developed for the CSI feedback of a single UE. To address this limitation, deep neural networks have been proposed to facilitate downlink CSI feedback for multiple UEs [21], [22]. Furthermore, distributed training algorithms, which are the focus of our work, have been explored using frameworks such as federated learning [23] and gossip learning [24]. However, the previous distributed training algorithms require each UE to either upload the full model parameters to the BTS [23] or to exchange full parameters with the neighbour UEs [24]. Full-parameter exchanges can introduce a communication bottleneck when the number of UEs becomes large.

Different from the previous distributed training algorithms, we propose a new communication-efficient model splitting algorithm that is named as **CSILocal** to train the CSI feedback deep neural networks (DNNs). Our key contributions are summarized as follows.

- To reduce the communication expenditure, the proposed CSILocal algorithm divides the CSI model into three modules, i.e., encoder, decoder tail, decoder head. Dif-

This work was supported by the National Natural Science Foundation of China under Grant 62102266 and the Pearl River Talent Recruitment Program of Guangdong Province under Grant 2019ZT08X603, Public Technology Platform of Shenzhen City (GGFW2018021118145859), Shenzhen Science and Technology Innovation Commission (R2020A045, KCXFZ20201221173411032).

Y. Dong, V. C. M. Leung, and X. Hu are with the Artificial Intelligence Research Institute and Guangdong-Hong Kong-Macao Joint Laboratory for Emotional Intelligence and Pervasive Computing, Shenzhen MSU-BIT University, Shenzhen 518172, China.

H. Zhang is with the Beijing Engineering and Technology Research Center for Convergence Networks and Ubiquitous Services, University of Science and Technology Beijing, Beijing 100083, China.

G. Chen is with the School of Flexible Electronics & State Key Laboratory of Optoelectronic Materials and Technologies, Sun Yat-Sen University, Shenzhen 518107, China.

X. Fan is with Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, and Jiangxing Intelligence Inc., Shenzhen, China.

ferent from traditional distributed training algorithms that need to exchange the full/partial model parameters, our proposed CSILOCAL algorithm only needs to exchange the local smashed data that is significantly less than the size of the CSI feedback DNN. In addition, the original CSI data are kept at each UE to preserve the privacy of UEs.

- We further split the decoder tail into several modules for pipeline parallelism such that the training duration can be further reduced.
- Numerical experiments are conducted to demonstrate that our proposed CSILOCAL algorithm can significantly reduce the communication costs between the UEs and BTS. Moreover, numerical results also verify that our proposed pipeline parallelism can reduce the training time duration.

Notations. Vectors and matrices are respectively denoted by lowercase- and uppercase-boldface letters. \mathbb{C} and \mathbb{R} respectively represent the sets of complex and real values. The operator \mathbf{X}^\dagger denotes the conjugate transpose of matrix \mathbf{X} .

II. SYSTEM DESCRIPTION

A. Communication Signaling

We consider the downlink transmission of an mMIMO system with a single BTS and N UEs. The BTS has N_t transmit antenna, and each UE has one single antenna. The BTS communicates with the UEs over N_c subcarriers. Let $\mathbf{H}_n = [\mathbf{h}_{n,1}, \mathbf{h}_{n,2}, \dots, \mathbf{h}_{n,N_c}] \in \mathbb{C}^{N_t \times N_c}$ and $\mathbf{W}_n = [\mathbf{w}_{n,1}, \mathbf{w}_{n,2}, \dots, \mathbf{w}_{n,N_c}] \in \mathbb{C}^{N_c \times N_c}$ denote the downlink CSI matrix and precoding matrix of UE n , respectively. When the input data streams of each UE n are segmented into N_c parallel streams as $\mathbf{x}_n \in \mathbb{C}^{N_c \times 1}$, the received signal at UE n is

$$\mathbf{y}_n = \mathbf{H}_n^\dagger \mathbf{W}_n \mathbf{x}_n + \mathbf{H}_n^\dagger \sum_{i \neq n}^N \mathbf{W}_i \mathbf{x}_i + \mathbf{z}_n \quad (1)$$

where $\mathbf{z}_n \in \mathbb{C}^{N_c \times 1}$ denotes the additive white Gaussian noise with each entry being independent and identically distributed (i.i.d.) complex Gaussian with mean zero and variance σ^2 .

Remark 1: The signal model in (1) demonstrates that the design of the precoding matrix \mathbf{W}_n critically depends on the accurate estimation of the downlink CSI matrices $[\mathbf{H}_n]_{n=1}^N$ as shown in [13], [15], [20]. When the mMIMO system operates in FDD mode, each UE n can estimate the downlink CSI matrix \mathbf{H}_n via the pilot-based channel estimation. The estimated CSI matrix is uploaded to the BTS via the feedback channels. However, the communication overhead associated with CSI feedback becomes significantly burdensome as the number of antennas and subcarriers increases. Therefore, we are motivated to train a DNN-based encoder to compress each CSI matrix \mathbf{H}_n into a codeword per each UE n and a DNN-based decoder to recover the compressed CSI matrices $[\mathbf{H}_n]_{n=1}^N$ at the BTS.

B. Model Splitting

Each UE can hold M private downlink CSI matrices and would not want to share them with the other entities. To preserve data privacy, the CSI feedback DNN needs to have a personal part at each UE and a shared part at the BTS

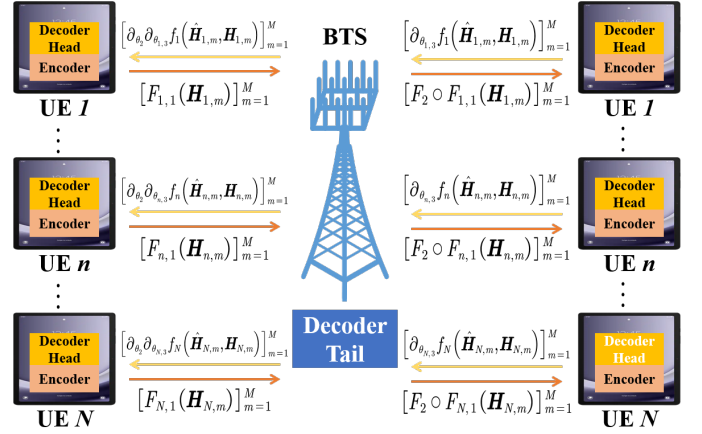


Fig. 1: The model is split into three parts with the encoder and decoder head at each UE and the decoder tail at the BTS.

to improve the training efficiency. We consider to split the model into three parts: the feature encoder and decoder head at each UE, and the decoder tail at the BTS. The UEs communicate with the BTS via digital transmission under the channel capacity. In this case, the communication link between each UE and the BTS experiences a negligible error rate [25]. We assume that all UEs have identical processing capabilities; therefore, the impact of stragglers can be ignored under digital transmission.

As shown in Fig. 1, we consider the federated CSI feedback learning where a BTS and N UEs coordinately work to minimize the normalized mean squared error as

$$\min_{\Theta} \sum_{n=1}^N \sum_{m=1}^M f_n(\hat{\mathbf{H}}_{n,m}, \mathbf{H}_{n,m}) \quad (2)$$

where $f_n(\hat{\mathbf{H}}_{n,m}, \mathbf{H}_{n,m}) = f(\theta_{n,1}, \theta_2, \theta_{n,3}; \mathbf{H}_{n,k}) := \|\hat{\mathbf{H}}_{n,m} - \mathbf{H}_{n,m}\|^2 / \|\mathbf{H}_{n,m}\|^2$ with $\hat{\mathbf{H}}_{n,m} = F_{n,3} \circ F_2 \circ F_{n,1}(\mathbf{H}_{n,m})$ denoting the m th reconstructed CSI matrix of UE n . The encoder $F_{n,1} : \mathbb{R}^{2 \times N_t \times N_c} \rightarrow \mathbb{R}^{c_1}$, the decoder tail $F_2 : \mathbb{R}^{c_1} \rightarrow \mathbb{R}^{c_2}$, and the decoder head $F_{n,3} : \mathbb{R}^{c_2} \rightarrow \mathbb{R}^{2 \times N_t \times N_c}$ are respectively parameterized by $\theta_{n,1} \in \mathbb{R}^{d_1}$, $\theta_2 \in \mathbb{R}^{d_2}$, and $\theta_{n,3} \in \mathbb{R}^{d_3}$ with $\Theta = \{[\theta_{n,1}, \theta_{n,3}]_{n=1}^N, \theta_2\}$ and $d = d_1 + d_2 + d_3$.

Remark 2: As shown in Fig. 1, the BTS needs to exchange local smashed data with the N UEs. More specifically, the boundary activation tensors $[F_{n,1}(\mathbf{H}_{n,m})]_{m=1}^M$ and $[F_2 \circ F_{n,1}(\mathbf{H}_{n,m})]_{m=1}^M$ are exchanged between the BTS and UE n during the forward pass. During the backward pass, the upper-layer gradients $[\partial_{\theta_{n,3}} f_n(\hat{\mathbf{H}}_{n,m}, \mathbf{H}_{n,m})]_{m=1}^M$ and $[\partial_{\theta_2} \partial_{\theta_{n,3}} f_n(\hat{\mathbf{H}}_{n,m}, \mathbf{H}_{n,m})]_{m=1}^M$ are exchanged between the BTS and UE n .

Remark 3: To reduce the communication expenditure between the BTS and the UEs, the dimensions at the sub-model boundaries (i.e., c_1 and c_2) should satisfy $c_1 \ll \min\{d_1, d_2\}$ and $c_2 \ll \min\{d_2, d_3\}$.

III. CSILOCAL ALGORITHM

For the model training problem (2), each UE n can calculate the local loss values during the forward passes and the gradi-

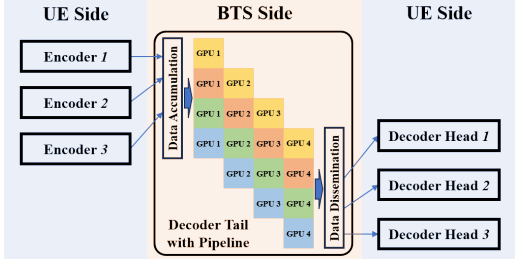


Fig. 2: An illustration of the pipeline parallelism for the federated CSI feedback learning with three UEs.

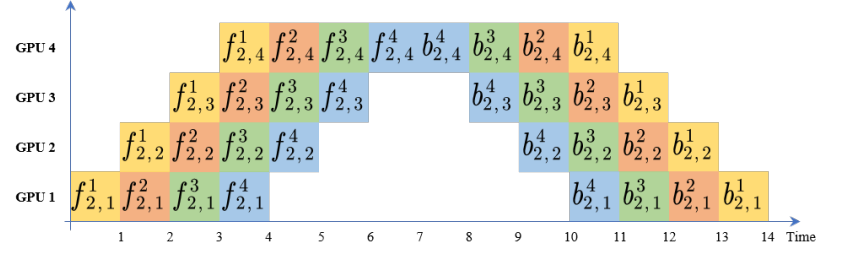


Fig. 3: An illustration of pipeline parallelism on the forward and backward passes when the BTS has four GPUs.

ents for the three parts of models during the backward passes. For each UE n , the gradients with respect to the encoder $\theta_{n,1}$, decoder tail θ_2 , and decoder head $\theta_{n,3}$ are respectively denoted by

$$\nabla_{n,1} := \nabla_{\theta_{n,1}} f_n(\hat{\mathbf{H}}_{n,m}, \mathbf{H}_{n,m}) \quad (3)$$

$$\nabla_2 := \sum_{n=1}^N \nabla_{\theta_2} f_n(\hat{\mathbf{H}}_{n,m}, \mathbf{H}_{n,m}) \quad (4)$$

and

$$\nabla_{n,3} := \nabla_{\theta_{n,3}} f_n(\hat{\mathbf{H}}_{n,m}, \mathbf{H}_{n,m}) \quad (5)$$

where m is uniformly drawn from the set $\{1, \dots, M\}$.

By leveraging the Adam optimizer [26], each part of the model can be updated per iteration k as

$$\theta_{n,1}^{k+1} = \theta_{n,1}^k - \eta \text{Adam}(\nabla_{n,1}^k, \theta_{n,1}^k) \quad (6)$$

$$\theta_2^{k+1} = \theta_2^k - \eta \text{Adam}(\nabla_2^k, \theta_2^k) \quad (7)$$

and

$$\theta_{n,3}^{k+1} = \theta_{n,3}^k - \eta \text{Adam}(\nabla_{n,3}^k, \theta_{n,3}^k) \quad (8)$$

where $\eta > 0$ is the learning rate.

Based on the pipeline parallelism in Fig. 2, we summarize the procedures of proposed **CSILocal** in **Algorithm 1**.

Algorithm 1 CSILocal Algorithm

- 1: Each UE n initializes an Adam optimizer with learning rate η and momentum factors (β_1, β_2)
 - 2: **for** $k = 1, 2, \dots, K$ **do**
 - 3: Each UE n feeds the CSI matrix $\mathbf{H}_{n,m}$ into the encoder n and uploads the smashed data of encoder n to the BTS
 - 4: The BTS accumulates all local smashed data from the N UEs via the data accumulation module
 - 5: The BTS splits the accumulate data into 4 micro batches as shown in Fig. 2
 - 6: The BTS processes the 4 micro-batches of smashed data via the decoder tail with pipeline as shown in Fig. 2
 - 7: The BTS disseminates the output of the decoder tail with pipeline to all UEs
 - 8: Each UE n calculates the loss value based on the output of decoder head and the label of CSI matrix $\mathbf{H}_{n,m}$
 - 9: Each UE n performs backward pass to calculate the gradients with respect to encoder, decoder tail and decoder head as (3)–(5)
 - 10: Each UE n updates the parameters of encoder and decoder head via (6) and (8)
 - 11: The BTS updates the parameters of decoder tail via (7)
 - 12: **end for**
-

Note that the volume of accumulated local smashed data at the BTS scales linearly with the number of UEs. When

the number of UEs is getting large, the pipeline parallelism is required to handle the large volume of local smashed data at the BTS as shown in Fig. 3. When the BTS has four GPUs, the decoder tail F_2 is divided into four sub-models to four GPUs, i.e., $F_2 = f_{2,4} \circ f_{2,3} \circ f_{2,2} \circ f_{2,1}$. At the data accumulation module of Fig. 2, the three mini-batch of smashed data are divided into four smaller micro batches. As shown in Fig. 3, GPU 2 can calculate the activation tensors of the first micro-batch of smashed data while the GPU 1 works on the calculation of forward activation tensors of the second micro-batch. As time goes by, the four GPUs can be used simultaneously to process the four micro-batches of smashed data. These procedures can be repeated during the backward passes. At the data dissemination module of Fig. 2, the outputs of decoder tail are then divided into three mini-batch of smashed data and delivered to the corresponding three decoder heads. The corresponding backward function can be computed via the automatic symbolic differentiation [27]. The introduced pipeline parallelism can reduce the consumed time for calculating the gradient ∇_2^k by improving GPU utilization efficiency on the BTS per each iteration k . In this way, the overall wallclock time for training can be reduced.

IV. NUMERICAL RESULTS

A. Neural Network and Hyper-Parameters

Description of neural network. To facilitate the communication efficiency, we reconstruct the downlink CSI matrices via an autoencoder-based DNN that consists of encoder, decoder tail, and decoder head. The proper padding is used to ensure the output of each convolutional layer match the size of the CSI matrix. As shown in Fig. 4, the detailed information of the three parts is as follows.

- **Encoder:** We reuse the encoder structure of CsiNet where the layer sequence is a convolutional layer with 2 kernels and kernel size as 3×3 , a batch-normalization layer, a leaky rectified linear (LeakyReLU) activation with negative slope as 0.3, and a fully connected layer.
- **Decoder Tail:** To enhance the performance of reconstruction, the decoder tail consists of two fully connected layers, a convolutional layer with 2 kernels and kernel size as 3×3 , a LeakyRelu activation with negative slope as 0.3, and two CRBlocks [28] with kernel sizes as 1×3 and 1×5 .
- **Decoder Head:** The decoder head uses the fully connected layer to recover the structure of the CSI matrices

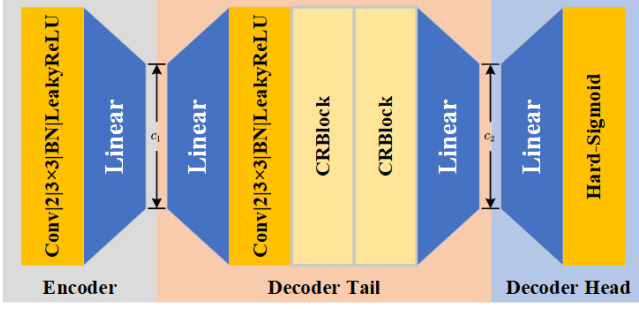


Fig. 4: An illustration of used deep neural network that have encoder, decoder tail, and decoder head. Each UE is equipped with encoder and decoder head, and the BTS is equipped with the decoder tail. In order to reduce the communication overhead, the encoding dimension $c_1 = c_2$ are set smaller than the dimension of each data sample.

and leverage the hard-Sigmoid activation to reconstruct the CSI matrices.

Hyper-parameter setup. We consider six benchmarks: FedAvg, FedAvgPer, FedProx, FedProxPer, FedGrad, and FedGradPer [2], [29]. More specifically, the FedAvg, FedGrad, and FedProx algorithms require the exchange of full model parameters (or gradients) between the UEs and the BTS in each iteration. In contrast, the FedAvgPer, FedGradPer, and FedProxPer algorithms only require the exchange of the model parameters (or gradients) of the decoder heads and tails between the UEs and the BTS per each iteration. For fair comparison, we use the same dataset as the CsiNet [10]. The generated CSI matrices are converted to angular-delay domain via the two-dimensional discrete fourier transform. We combine the original 100,000 training and 30,000 validation samples are the training set, and use the remaining 20,000 samples as the testing set. Each data sample has size 32×32 . Unless otherwise specified, the mMIMO system is configured with 10 UEs. We set the training iterations as 20,000, the learning rate as 8×10^{-5} , and the momentum-factor tuple for the Adam optimizer as (0.9, 0.95). Since we have limited number of GPUs, we set the number of micro batches as 2. We use the normlized mean-squared error loss. Unless otherwise specified, we set the mini-batch size as 800, and the dimensions of sub-model boundaries c_1 and c_2 as 256.

Non-IID setup. In order to demonstrate the impact of data heterogeneity in CSI data, we construct a synthetic dataset consisting of 130,000 training samples and 20,000 testing samples. More specifically, the ratio of indoor to outdoor CSI data is set to 1 : 1 in both the training and testing sets. In the mMIMO system, each UE is assigned 13,000 data samples, and the indoor-to-outdoor CSI ratios for the 10 UEs are set as follows: 9.5 : 0.5, 8.5 : 1.5, 7.5 : 2.5, 6.5 : 3.5, 5.5 : 4.5, 4.5 : 5.5, 3.5 : 6.5, 2.5 : 7.5, 1.5 : 8.5, and 0.5 : 9.5.

B. Pipeline vs. No Pipeline

To implement the pipeline parallelism with two micro-batches, the decoder tail is divided into two sub-models by splitting at the connection between two CRBlocks. Each sub-model is allocated to an independent GPU accelerator. For the “no pipeline” baseline, the entire decoder tail is deployed on

TABLE I: Duration per iteration for training CSILocal model with and without pipeline parallelism at the server (measured in seconds).

Environments	Indoor			
Encoding Dimension	256		512	
Batch Size per UE	400	800	400	800
CsiLocal with Pipeline	0.0560	0.0952	0.0574	0.0974
CsiLocal w/o Pipeline	0.0583	0.1160	0.0597	0.1184
Environments	Outdoor			
Encoding Dimension	256		512	
Batch Size per UE	400	800	400	800
CsiLocal with Pipeline	0.0561	0.0951	0.0562	0.0972
CsiLocal w/o Pipeline	0.0582	0.1160	0.0596	0.1184

a single GPU accelerator. By setting the encoding dimension (i.e., $c_1 = c_2$) to 256 and 512, we compare the wallclock durations of the “pipeline” and “no pipeline” methods using indoor CSI data and outdoor CSI data. The results are presented in Table I. When the mini-batch size per UE is set as 400 and encoding dimension is set as 256, wallclock durations for training 20,000 iterations is approximately reduced by 4.11% for indoor data and 3.74% for outdoor data. When the mini-batch size per UE is set as 800 and encoding dimension is set as 256, wallclock durations for training 20,000 iterations is approximately reduced by 21.85% for indoor data and 21.98% for outdoor data. We have the similar observations when setting the encoding dimension as 512. Therefore, we conclude that the wallclock reduction can become even bigger when the mMIMO system has more UEs and larger mini-batch size. We will use the mini-batch size as 800 for the remaining numerical experiments.

C. Impacts of Compression Ratio

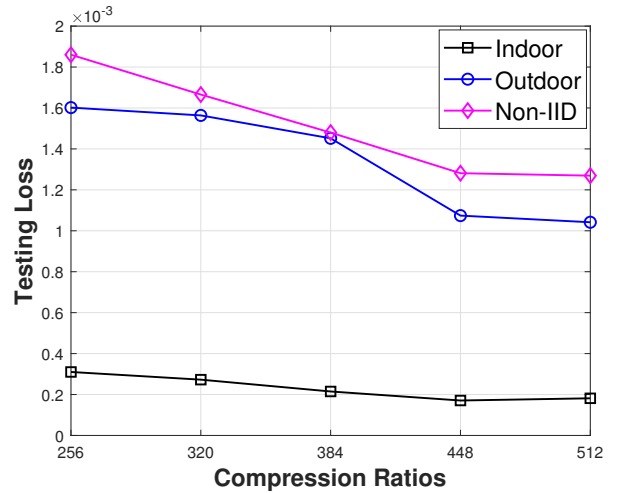


Fig. 5: The impact of compression ratios on the CSI reconstruction accuracy.

Figure 5 illustrates the impact of varying compression ratios on the testing loss across three distinct data distributions: Indoor, Outdoor, and Non-IID. We observe that the proposed CSILocal algorithm effectively reduces testing loss and underscores its ability to balance the trade-off between CSI reconstruction accuracy and communication efficiency. Notably, the Non-IID setting consistently exhibits the highest testing loss,

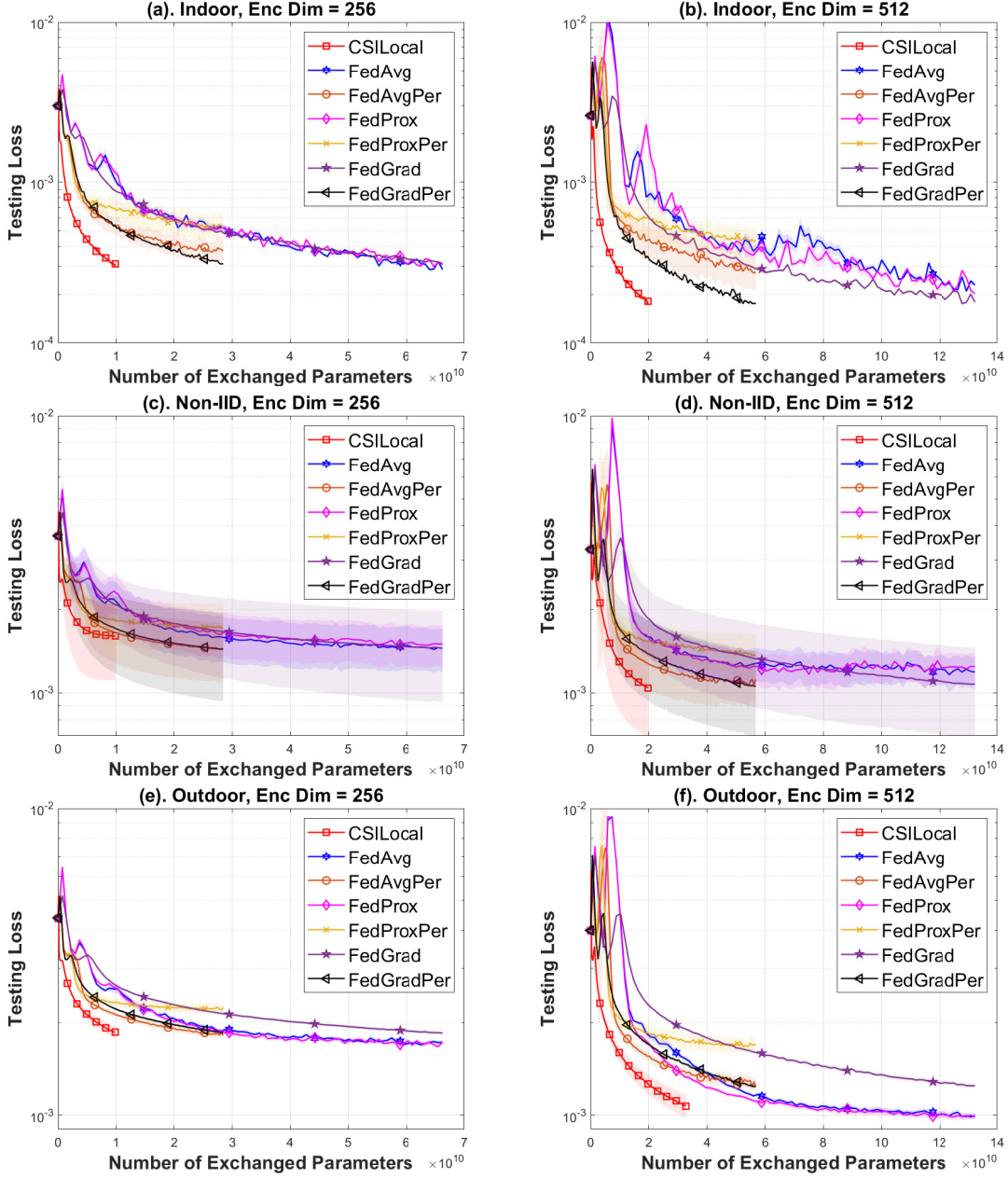


Fig. 6: The convergence of testing loss over the communication overhead for the different environments and encoding dimensions.

which highlights the adverse effects of data heterogeneity on model performance. Furthermore, the results in Fig. 5 confirm that increased compression ratios generally lead to improved reconstruction accuracy with the degradation being particularly pronounced under Non-IID conditions.

D. Communication Efficiency

Figure 6 illustrates the convergence of testing loss with respect to the number of exchanged parameters under three distinct data distributions: Indoor, Outdoor, and Non-IID. We observe from Fig. 6 that our proposed CSILocal algorithm can always achieve the lowest testing loss while maintaining a similar testing loss over the benchmarks. In the indoor scenario with 256 encoding dimension (Fig. 6(a)), the proposed CSILocal achieves relatively stable and low testing loss. In contrast, the outdoor scenario with 256 encoding dimension

(Fig. 6(e)) demonstrates slightly higher loss values and greater variability which reflects the increased channel fluctuation and reduced predictability typical of outdoor settings. The most challenging scenario appears in the heterogeneous case (Fig. 6(c)), where data is non-IID across UEs. More specifically, Fig. 6(c) illustrates that the heterogeneous scenario achieves a consistently higher testing loss than indoor and outdoor scenarios, which confirms the compounded effect of data heterogeneity and information loss due to compression. All the three settings indicate that the proposed CSILocal algorithm can maintain acceptable reconstruction performance under low-to-moderate compression. These insights validate CSILocal's robustness in various channel conditions and its potential limitations when handling heterogeneous UE data. The non-IID scenario can experience a more severe CSI reconstruction degradation over the indoor and outdoor scenarios. Besides,

the non-IID data scenario can induce higher fluctuation of testing loss across different UEs over the indoor and outdoor scenarios. However, as shown in Fig. 6(b), Fig. 6(d), and Fig. 6(f), the CSI reconstruction degradation and the fluctuation of testing loss can be compensated by increasing encoding dimensions. Although FedAvg and FedProx outperform our CSILocal in Fig. 6(f), they require significantly larger volume of exchanged parameters. Specifically, CSILocal necessitates only 19.66 billion exchanged parameters, whereas FedAvg and FedProx respectively require 47.06 billion and 38.24 billion parameters to achieve the same testing loss (e.g., 1.3×10^{-3}).

V. CONCLUDING REMARKS

A novel communication-efficient model splitting algorithm (i.e., CSILocal) was presented to reduce the communication overhead between the UEs and the BTS in the mMIMO systems. The CSILocal algorithm allows each UE to exchange only local smashed data with the BTS, and thereby significantly minimizes the communication overhead while preserving local data privacy. To further optimize system performance, the pipeline parallelism was integrated into the decoder tail at the BTS. The pipeline parallelism allows efficient processing of the accumulated smashed data and reduces the overall wallclock duration required for model training. By distributing the computational tasks across multiple processing stages, pipeline parallelism can enhance the scalability of the system. Empirical evaluations had demonstrated that the proposed approach achieves substantial reductions in wallclock duration with the observed reduction ratio increasing proportionally to both the number of UEs involved in the system and the mini-batch size allocated per UE. These results highlighted the scalability and effectiveness of the proposed model-splitting algorithm in large-scale mMIMO deployments.

REFERENCES

- [1] Z. Chen, G. Chen, J. Tang, S. Zhang, D. K. So, O. A. Dobre, K.-K. Wong, and J. Chambers, "Reconfigurable-intelligent-surface-assisted B5G/6G wireless communications: Challenges, solution, and future opportunities," *IEEE Commun. Mag.*, vol. 61, no. 1, pp. 16–22, Jan. 2023.
- [2] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, Y. C. Eldar, and M. Debbah, "Edge learning for B5G networks with distributed signal processing: Semantic communication, edge computing, and wireless sensing," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 9–39, Jan. 2023.
- [3] B. Liu, X. Liu, S. Gao, X. Cheng, and L. Yang, "LLM4CP: Adapting large language models for channel prediction," *J. Commun. Inform. Netw.*, vol. 9, no. 2, pp. 113–125, June 2024.
- [4] D. Verenzuela, E. Björnson, X. Wang, M. Arnold, and S. ten Brink, "Massive-MIMO iterative channel estimation and decoding (MICED) in the uplink," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 854–870, Feb. 2020.
- [5] Y. Gao, H. Hu, J. Chen, X. Wang, X. Chu, and J. Zhang, "A matching-based pilot assignment algorithm for cell-free massive MIMO networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 1453–1457, Jan. 2024.
- [6] 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, Dec. 2022.
- [7] R. M. Dreifuerst and R. W. Heath, "Massive MIMO in 5G: How beamforming, codebooks, and feedback enable larger arrays," *IEEE Commun. Mag.*, vol. 61, no. 12, pp. 18–23, Dec. 2023.
- [8] 3GPP, "TSG-RAN WG1 #89: R1-1709232 WF on type I and II CSI codebooks," Tech. Rep., 2017.
- [9] P. Liang, J. Fan, W. Shen, Z. Qin, and G. Y. Li, "Deep learning and compressive sensing-based CSI feedback in FDD massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9217–9222, Aug. 2020.
- [10] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748–751, Oct. 2018.
- [11] S. Ji and M. Li, "CLNet: Complex input lightweight neural network designed for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 10, no. 10, pp. 2318–2322, Oct. 2021.
- [12] Y. Chi, L. Liu, Y. Ge, X. Chen, Y. Li, and Z. Zhang, "Interleave frequency division multiplexing," *IEEE Wireless Commun. Lett.*, vol. 13, no. 7, pp. 1963–1967, July 2024.
- [13] Y. Dong, H. Zhang, J. Li, F. R. Yu, S. Guo, and V. C. M. Leung, "An online zero-forcing precoder for weighted sum-rate maximization in green CoMP systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 7566–7581, Sept. 2022.
- [14] X. Li, L. Liu, R. Zhou, N. Zhang, C. Wu, M. Atiquzzaman, and M. Guizani, "Connectivity analysis for V2I communications in cognitive vehicular networks," *IEEE Trans. Veh. Technol.*, to be published, 2024, doi: 10.1109/TVT.2024.3481300.
- [15] Y. Dong, L. Wang, J. Wang, X. Hu, H. Zhang, F. R. Yu, and V. C. M. Leung, "Accelerating wireless federated learning via Nesterov's momentum and distributed principal component analysis," *IEEE Trans. Wireless Commun.*, vol. 23, no. 6, pp. 5938–5952, June 2024.
- [16] Y. Wang, Z. Gao, D. Zheng, S. Chen, D. Gündüz, and H. V. Poor, "Transformer-empowered 6G intelligent networks: From massive MIMO processing to semantic communication," *IEEE Wireless Commun.*, vol. 30, no. 6, pp. 127–135, Dec. 2023.
- [17] J. Huang *et al.*, "Foundation models and intelligent decision-making: Progress, challenges, and perspectives," *The Innovation*, vol. 6, no. 6, May 2025.
- [18] S. Zhang, W. Xu, S. Jin, X. You, D. W. K. Ng, and L.-C. Wang, "Dual-propagation-feature fusion enhanced neural CSI compression for massive MIMO," *IEEE Trans. Commun.*, vol. 71, no. 9, pp. 5182–5198, Sept. 2023.
- [19] F. Sahrabi, K. M. Attiah, and W. Yu, "Deep learning for distributed channel feedback and multiuser precoding in FDD massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4044–4057, July 2021.
- [20] Z. Gao, M. Wu, C. Hu, F. Gao, G. Wen, D. Zheng, and J. Zhang, "Data-driven deep learning based hybrid beamforming for aerial massive MIMO-OFDM systems with implicit CSI," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 10, pp. 2894–2913, Oct. 2022.
- [21] M. B. Mashhadi, Q. Yang, and D. Gündüz, "Distributed deep convolutional compression for massive MIMO CSI feedback," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2621–2633, Apr. 2021.
- [22] J. Guo, X. Yang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based CSI feedback for RIS-assisted multi-user systems," *arXiv preprint arXiv:2003.03303v4*, Mar. 2024.
- [23] Y. Cui, J. Guo, C.-K. Wen, and S. Jin, "Communication-efficient personalized federated edge learning for massive MIMO CSI feedback," *IEEE Trans. Wireless Commun.*, vol. 23, no. 7, pp. 7362–7375, July 2024.
- [24] J. Guo, Y. Zuo, C.-K. Wen, and S. Jin, "User-centric online gossip training for autoencoder-based CSI feedback," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 559–572, Apr. 2022.
- [25] J. Yao, W. Xu, Z. Yang, X. You, M. Bennis, and H. V. Poor, "Wireless federated learning over resource-constrained networks: Digital versus analog transmissions," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 14 020–14 036, Oct. 2024.
- [26] A. Défossez, L. Bottou, F. Bach, and N. Usunier, "A simple convergence proof of Adam and Adagrad," *Trans. Mach. Learn. Research*, 2022.
- [27] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu *et al.*, "Gpipe: Efficient training of giant neural networks using pipeline parallelism," in *Proc. NeurIPS*, vol. 32, Vancouver, BC, Canada, Dec. 2019.
- [28] Z. Lu, J. Wang, and J. Song, "Multi-resolution CSI feedback with deep learning in massive MIMO system," in *Proc. IEEE ICC*, Dublin, Ireland, June 2020, pp. 1–6.
- [29] K. Pillutla, K. Malik, A.-R. Mohamed, M. Rabbat, M. Sanjabi, and L. Xiao, "Federated learning with partial model personalization," in *Proc. ICML*, Baltimore, MD, USA, July 2022, pp. 17 716–17 758.