THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e. g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

- This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.
- A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.
- This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.
- The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.
- When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

The University of Edinburgh

Ph.D. Thesis

Distributed Learning Techniques for Future Wireless Networks

Author: Yuchen Mu

Supervisor: Prof. Tharmalingam Ratnarajah

A thesis submitted in fulfilment of the requirements for the degree of Ph.D.

in the

School of Engineering

October 2024

Declaration of Authorship

I declare that this thesis was composed by myself, that the work contained therein is my own except where explicitly stated otherwise, and that none of the work has been submitted for any other degree or professional qualification.

Yuchen Mu

Abstract

In recent years, the rapid advancement of wireless communication technologies has spurred significant interest in distributed learning algorithms. These algorithms, such as Federated Learning (FL), designed to leverage data distributed across multiple nodes and facilitate efficient model training, are crucial for applications ranging from Internet of Things (IoT) networks to edge computing and beyond. The convergence of machine/deep learning and wireless communications promises to enhance the efficiency and scalability of data processing tasks by enabling real-time, decentralized data analysis. However, the integration of FL into wireless communication frameworks poses unique challenges, such as ensuring robust training under varying data and device heterogeneity, optimizing resource allocation, and maintaining data privacy and security. Addressing these challenges is significant for the effective deployment of distributed learning systems in practical wireless environments. Therefore, in this thesis, we timely study the feasibility of FL and investigate its advantages for emerging applications in future networks.

In the first part, we investigate a FL algorithm within a multi-user massive multiple-input multiple-output (mMIMO) 6G network. Since in FL, model weights must be updated at local users and the base station (BS). These weights are subjected to uplink (UL) and downlink (DL) transmission errors due to the limited reliability of wireless channels, we investigate the impact of imperfections in both UL and DL links. Zero-forcing (ZF) and minimum mean square error (MMSE) schemes are employed to null the intra-cell interference, and we analyze the estimation errors of weights for each round. A tighter convergence bound on the modeling error for the communication-efficient FL algorithm is derived. The analysis shows that the reliability of DL links is more critical than that of UL links, and the transmit power can be varied in the training process to reduce energy consumption. We also vary the number of local training steps, average codeword length after quantization, and scheduling policy to improve communication efficiency. Simulations with image classification problems on MNIST, EMNIST, and FMNIST datasets verify the derived bound and are useful to infer the minimum SNR required for successful convergence of the FL algorithm.

In the second part, we extend our work by considering federated distillation (FD), which is a novel distributed learning paradigm that shares the privacy-preserving nature of FL and provides possible solutions to the challenges introduced by the FL framework, such as being able to train local models with nonidentical architectures. We present a communication channel-aware FD framework for a multi-user mMIMO communication system. Unlike most existing studies, where both model parameters and model outputs (logits) are utilized for transmission, we exclusively adopt logits as the information exchanged

in the wireless links to reduce the overall communication overhead in each round. Based on the analysis, the dynamic training steps-based FD algorithm (FedTSKD) is proposed to save communication resources and accelerate the training process. Further, a group-based FD algorithm (FedTSKD-G) is proposed for the system experiencing different channel conditions like deep-fade. Simulation results on image classification tasks with ImageNette/STL-10, CIFAR-10/STL-10, and MNIST/FMNIST dataset combinations have demonstrated the proposed algorithm's effectiveness and efficiency. Comparison with the FL algorithm shows that the proposed FD algorithm only incurs 1% of FL's communication overhead to achieve the same testing performance.

In the third part, we focus on a fully distributed FL with model pruning over serverless wireless networks. In each global communication round, the K-means algorithm is deployed to perform dynamic clustering of devices based on their geographical proximity and the device-to-device (D2D) communication technique is used to share the updated model weights. Then, we mathematically analyze the computation and communication latency, and convergence of the proposed system. To improve convergence rate, guarantee learning accuracy, and decrease computation and communication latency, *Karush-Kuhn-Tucker* (KKT) conditions are employed to jointly optimize the pruning ratio and bandwidth allocation. Simulation results demonstrate that the proposed FL algorithm necessitates only 88% of the training time required by the system employing equal pruning and 80% of the training time needed by the system without pruning, while achieving a testing accuracy comparable to that of the latter.

Lay Summary

In today's interconnected world, the volume of data generated by devices like smartphones, sensors, and other wireless technologies is growing exponentially. Managing and processing this data efficiently and securely is a major challenge. In this thesis, we explore a promising solution called Federated Learning (FL), which allows multiple devices to collaboratively train machine learning models without needing to share their raw data. This approach not only protects user privacy but also enhances the efficiency of data processing in wireless networks. However, integrating FL into wireless communication systems poses several challenges, such as unreliable communication links, slow convergence, and data and model heterogeneity. This thesis focuses on addressing these challenges to enable the widespread use of FL in practical wireless environments.

To achieve this objective, first, we investigate how FL can be adapted for advanced wireless networks equipped with multiple antennas, known as massive MIMO systems. These systems are a key component of future 6G networks. We study the impact of communication errors on FL performance and develop techniques to minimize these errors and improve efficiency. This work helps establish best practices for implementing FL in high-capacity wireless networks.

Then, the second part of this thesis introduces an innovative variation of FL called Federated Distillation (FD). Unlike traditional FL, FD allows each device to use a different model architecture, providing greater flexibility and efficiency. We design an FD framework that adapts to varying communication conditions, significantly reducing the amount of data that needs to be transmitted while maintaining high model performance. This approach is particularly useful for environments where devices have different capabilities and network conditions.

Furthermore, we explore a fully decentralized FL system that operates without a central server. By enabling D2D communication between devices, this system reduces reliance on central infrastructure and improves the overall efficiency of the network. We develop algorithms to optimize how devices share data and train models collaboratively, ensuring high accuracy while minimizing resource usage.

Overall, the research presented in this thesis contributes to making FL a feasible solution for future wireless networks. By addressing key challenges and developing innovative techniques, this thesis exploits the potential for more secure, efficient, and scalable data processing, which is essential for the advancement of next-generation wireless communication technologies.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Tharmalingam Ratnarajah, for his invaluable guidance, unwavering support, and insightful mentorship throughout my Ph.D. studies. His expertise, encouragement, and dedication have been instrumental in shaping my research journey and nurturing my academic growth. I am profoundly grateful for his generosity, patience, and encouragement, which have inspired me to strive for excellence and pursue new intellectual challenges.

I would also like to express my sincere appreciation to the Institute for Imaging, Data and Communications for offering the necessary support.

Furthermore, I extend my gratitude to Dr. Navneet Garg and Dr. Xiaonan Liu, for their generous help during my research.

I am deeply thankful to my parents, Mr. Jiankang Mu and Mrs. Xiuying Feng, and my younger brother Yujia Mu, for their selfless love, encouragement, and unwavering support throughout my academic journey. My parents have sacrificed so much to ensure that I have the best opportunities possible, and I owe all that I have accomplished to their enduring dedication and love.

Additionally, I am deeply grateful to my girlfriend, Mrs. Lansha Chen, for her unwavering support and encouragement throughout this journey. Her patience, positivity, and constant reassurance, especially during my most anxious moments before the PhD viva, gave me the strength to keep going. I feel incredibly fortunate to have met you in my life - thank you for always being by my side.

Lastly, I want to acknowledge myself for never giving up and maintaining faith despite the numerous frustrations and failures I have encountered over the past ten years. While the saying "pain is wealth" might seem like nonsense, I believe it is the reflection on pain that becomes true wealth. If I could travel back to the day I did not perform well in the university entrance examination due to my right-hand medical condition and some other unexpected circumstances, I would pat the shoulder of that young boy and say to him, "Don't be afraid. The game has just begun."

"我始终如一。"

This PhD project was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/T021063/1.

Contents

Declaration of Authorship i

Abstract ii

Lay Summary iv

Acknowledgements v

Contents vi

List of Figures x

List of Tables xiii

List of Abbreviations xiv

List of Notations xvii

List of Symbols xix

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Contributions . . . 4
- 1.3 Thesis Overview . . . 6
- 1.4 List of Publications . . . 7

2 Background 9
- 2.1 Introduction . . . 9
- 2.2 MIMO Communications . . . 9
  - 2.2.1 Point-to-point MIMO systems . . . 10
  - 2.2.2 Capacity of MIMO systems . . . 12
  - 2.2.3 Multi-user MIMO (MU-MIMO) systems . . . 16
  - 2.2.4 Massive MIMO . . . 20
- 2.3 Machine/Deep Learning Techniques . . . 21
  - 2.3.1 Convolutional neural networks (CNNs) . . . 23

2.3.2 Loss function . . . 25
2.3.3 Training optimization algorithms . . . 27
2.4 Chapter Summary . . . 32

3 Federated Learning in Massive MIMO Networks 33
3.1 Introduction . . . 33
3.2 Related Work . . . 34
3.2.1 Quantization schemes . . . 35
3.2.2 Resource allocation . . . 35
3.2.3 Imperfect links . . . 36
3.2.4 Contributions . . . 36
3.3 System Model . . . 40
3.3.1 Local training . . . 41
3.3.2 Uploading model weights . . . 41
3.3.3 Aggregation . . . 42
3.3.4 Downloading updated weights . . . 42
3.4 Uplink and Downlink Communications . . . 43
3.4.1 Uplink mapping . . . 43
3.4.2 Aggregation mapping . . . 44
3.4.3 Downlink mapping . . . 45
3.4.4 Convergence analysis . . . 46
3.5 Simulation Results . . . 52
3.5.1 Simulations setup . . . 52
3.5.2 Testing accuracy for fixed learning rate . . . 54
3.5.3 Testing accuracy for varying SNR and learning rate . . . 58
3.5.4 Simulations on non-i.i.d. MNIST . . . 59
3.5.5 Simulations on EMNIST and FMNIST datasets . . . 64
3.6 Conclusion . . . 65
Appendix 3.7 Proof of Lemma 3.1 . . . 65
Appendix 3.8 Proof of Lemma 3.2 . . . 67
Appendix 3.9 Proof of Theorem 3.3 . . . 67
Appendix 3.10 Proof of Corollary 3.4 . . . 68

4 Federated Distillation in Massive MIMO Networks 69
4.1 Introduction . . . 69
4.2 Related Work . . . 70
4.2.1 Knowledge distillation-assisted FL/FD frameworks . . . 70
4.2.2 Communication-integrated FD frameworks . . . 71
4.2.3 Contributions . . . 73
4.3 System Model . . . 75
4.3.1 Assumptions on the loss functions . . . 76
4.3.2 Learning dynamics . . . 77
4.4 Uplink and Downlink Communications . . . 78
4.4.1 Uplink mapping . . . 79
4.4.2 Server aggregation and distillation mapping . . . 80
4.4.3 Downlink mapping . . . 80
4.5 Convergence Analysis . . . 81

4.6 Communication Channel-aware FD Algorithm . . . 85
4.7 Simulation Results . . . 87
4.7.1 Error-prone and error-free comparison . . . 90
4.7.2 Different degrees of non-i.i.d.-ness (α_d) . . . 91
4.7.3 With different precoding methods . . . 91
4.7.4 With training steps variations and group splitting . . . 92
4.7.5 Model heterogeneity . . . 92
4.7.6 With different numbers of clients . . . 95
4.7.7 Communication overhead comparison between FD and FL . . . 95
4.7.8 With different numbers of BS and UE antennas . . . 96
4.7.9 Simulations on ImageNette dataset . . . 97
4.8 Conclusion . . . 97
Appendix 4.9 Proof of variance for the total effective noise . . . 98
Appendix 4.10 Proof of Lemma 4.1 . . . 98
Appendix 4.11 Proof of Corollary 4.2 . . . 99
Appendix 4.12 Proof of Lemma 4.3 . . . 99
Appendix 4.13 Proof of Corollary 4.4 . . . 100
Appendix 4.14 Proof of Theorem 4.5 . . . 101

5 Adaptive Model Pruning in Distributed Federated Learning 102
5.1 Introduction . . . 102
5.2 Related Work . . . 103
5.2.1 FL in wireless communication systems . . . 103
5.2.2 Fully distributed FL . . . 103
5.2.3 Clustered FL . . . 104
5.2.4 Model pruning-enabled FL . . . 104
5.2.5 Contributions . . . 105
5.3 System Model . . . 106
5.3.1 Fully distributed federated learning process . . . 106
5.3.2 Overall computation and communication latency . . . 109
5.3.3 Optimization problem formulation . . . 110
5.4 Convergence Analysis and Problem Transformation . . . 110
5.4.1 Model pruning . . . 111
5.4.2 Convergence analysis . . . 113
5.4.3 Optimization problem transformation . . . 120
5.5 Optimization of Pruning Ratio and Bandwidth Allocation . . . 121
5.5.1 Optimization of pruning ratio . . . 121
5.5.2 Optimization of bandwidth allocation . . . 122
5.6 Simulation Results . . . 124
5.6.1 Simulation setup . . . 124
5.6.2 For different values of SNRs . . . 128
5.6.3 For different pruning scores . . . 129
5.6.4 For non-i.i.d. training datasets . . . 129
5.6.5 For different values of pruning ratios . . . 129
5.6.6 Pruning ratio and bandwidth optimization . . . 129
5.7 Conclusion . . . 131

6 Conclusion and Future Work **132**
6.1 Summary of Contributions 132
6.2 Future Work 134
6.2.1 FL in mMIMO networks 134
6.2.2 FD in mMIMO networks 136
6.2.3 Distributed FL with adaptive model pruning 138
6.2.4 Some other general extensions 138

Bibliography **140**

List of Figures

1.1 The estimated total global mobile traffic per month [1]. 2

2.1 An illustration of point-to-point MIMO system with N_t transmitting antennas and N_r receiving antennas. 10
2.2 An illustration of a multi-user MIMO system that includes K users, where the BS is equipped with M antennas and each user is equipped with N antennas. 17
2.3 BER vs. SNR for a single-cell mMIMO system with ZF precoding (DL transmission with QPSK modulation) 21
2.4 An example of deep neural network (DNN) for Iris flower classification problem, in this case, the model maps the four features (sepal length, sepal width, petal length, and petal width) to the category of the flowers. 22
2.5 Convolution operation between an image and a 3×3 filter (kernel), the stride is set to 1. After the convolution, the image patch is mapped to 19, as shown in the output. 23
2.6 Max and average pooling operations between an original feature map and a 2×2 filter, the stride is set to 2. 25
2.7 Gradient descent without momentum (red curve) vs. gradient descent with momentum (blue curve). 31

3.1 Architecture of the federated learning framework with imperfect wireless links, where channel fading, intra-cell interference, quantization noise and receiver side noise are considered. 40
3.2 Illustration of the timeline of communication rounds in federated leaning. Up and down arrows signify the uplink and downlink communications, respectively. 42
3.3 Histogram of the model weights. 44
3.4 Testing accuracy of the training progress at server side in (A), and local user side in (B), for different SNRs of UL links while DL-SNR is fixed to 10 dB; that is, $P_{DL,0}/\sigma^2 = 10$ dB. 54
3.5 Testing accuracy of the training progress at server side in (A), and local user side in (B), for different SNRs of DL links while UL-SNR is fixed to -3 dB; that is, $P_{UL,0}/\sigma^2 = -3$ dB. 55
3.6 Converged local testing accuracy versus SNRs after $R = 100$ global communication rounds. 56
3.7 Testing accuracy at local user side on MNIST i.i.d. dataset when different precoding methods are implemented. $N_{BS} = 64$. Values of DL-SNR and UL-SNR are 8 dB and -8 dB, respectively; that is, $P_{DL,0}/\sigma^2 = 8$ dB, $P_{UL,0}/\sigma^2 = -8$ dB. 57

3.8 Testing accuracy at local user side on MNIST i.i.d. dataset when full participation, UL-channel-state-based scheduling policy and l_2-norm-based scheduling policy are considered. Values of DL-SNR and UL-SNR are 8 dB and -8 dB, respectively; that is, $P_{DL,0}/\sigma^2 = 8$ dB, $P_{UL,0}/\sigma^2 = -8$ dB. 58
3.9 Progresses of testing accuracy (left-axis) and testing loss $F(\mathbf{w})$ (right axis) at the server side to compare the effects of varying and fixed SNRs through training process. The fitted line for varying SNR shows $\mathcal{O}\left(t_\tau^{-1}\sigma_z^2\right)$ behavior as proved in Theorem 3.3. 59
3.10 Testing accuracy on non-i.i.d. MNIST dataset versus system running time under different system schemes. Number of quantization bit is 16. $N_{BS} = 128$. Non-i.i.d. factor = 70%. Values of DL-SNR and UL-SNR are 7 dB and -7 dB, respectively; that is, $P_{DL,0}/\sigma^2 = 7$ dB, $P_{UL,0}/\sigma^2 = -7$ dB. 60
3.11 Visualization of communication cost under different systems schemes. Non-i.i.d. MNIST dataset is considered. 62
3.12 Testing accuracy on MNIST dataset with different non-i.i.d. factors. Number of quantization bit is 16. $N_{BS} = 128$. Values of DL-SNR and UL-SNR are 7 dB and -7 dB, respectively; that is, $P_{DL,0}/\sigma^2 = 7$ dB, $P_{UL,0}/\sigma^2 = -7$ dB. 62
3.13 Testing accuracy on non-i.i.d. MNIST dataset when two quantization schemes are used. $N_{BS} = 128$. Values of DL-SNR and UL-SNR are 7 dB and -7 dB, respectively; that is, $P_{DL,0}/\sigma^2 = 7$ dB, $P_{UL,0}/\sigma^2 = -7$ dB. . . 63
3.14 Testing accuracy at local user side on MNIST, EMNIST and FMNIST datasets. $N_{BS} = 64$ and number of local epochs is 7. Value of UL-SNR is fixed to -3 dB; that is, $P_{UL,0}/\sigma^2 = -3$ dB. 64

4.1 Architecture of the proposed massive MIMO-integrated FD framework with imperfect communication links, where quantization noise, channel fading, intra-cell interference and receiver side noise are considered. $\mathcal{D}_1$ and $\mathcal{X}$ represent the personal dataset of client 1 and public dataset, respectively. $\mathbf{z}_{p,r+1}^{[g]}$ denotes the predicted logits on the public dataset by server side model at $(r+1)^{th}$ round and $\hat{\mathbf{z}}_{p,n,r+1}^{[g]}$ stands for its corresponding noisy version, received by n^{th} user. Note that all the local users share similar procedures as shown for Client 1, detailed local procedures for Client 2 and Client N are omitted in this figure for better visualization. 75
4.2 Illustration of the timeline of communication rounds (r) in federated distillation. Up and down arrows signify the uplink and downlink communications, respectively. Abbreviations are defined as: "LT": Local user side Training, "LI": Local user side Inference, "LD": Local user side Distillation, "SD": Server side Distillation. 77
4.3 Concatenation of logits at server side. 85
4.4 Averaged testing accuracy across users with CIFAR-10/STL-10 datasets with different DL SNRs and fixed UL SNR -8 dB. 89
4.5 Testing accuracy across different users model architecture groups with CIFAR-10/STL-10 datasets with DL SNR -20 dB and UL SNR -8 dB. 89
4.6 Averaged testing accuracy across users with CIFAR-10/STL-10 datasets for UL SNR -8 dB, DL SNR -20 dB, and different non-i.i.d.-ness $\alpha_d = 0.5, 1, 5, 10$. 90
4.7 Averaged testing accuracy with MNIST/FMNIST datasets for UL SNR -8 dB, DL SNR -20 dB, and different non-i.i.d.-ness $\alpha_d = 0.5, 1, 5, 10$. . 91

4.8 Averaged testing accuracy across users for different precoding methods at fixed UL-SNR −8 dB and DL-SNR −20 dB with CIFAR-10/STL-10 datasets. . . . 92
4.9 Averaged testing accuracy across users for CIFAR-10/STL-10 datasets with fixed UL-SNR −8 dB and DL-SNR −20 dB (TG: Two Group Split, TS: Training Steps Variations). . . . 93
4.10 Converged testing accuracy with MNIST/FMNIST datasets when different numbers of clients are considered. UL SNR −8 dB, DL SNR −20 dB, $\alpha_d = 10$. . . . 94
4.11 Averaged testing accuracy versus the cumulative overhead across communication rounds for FD and FL algorithms with fixed UL SNR −8 dB, DL SNR 3 dB and with MNIST/FMNIST datasets . . . 95
4.12 Averaged testing accuracy across users for different numbers of BS/UE antennas at fixed UL-SNR −8 dB and DL-SNR 0 dB with CIFAR-10/STL-10 datasets. . . . 96
4.13 Averaged testing accuracy across users for different numbers of BS/UE antennas at fixed UL-SNR −8 dB and DL-SNR 0 dB with ImageNette/STL-10 datasets. . . . 97

5.1 Architecture of the fully distributed federated learning framework with clustering and model pruning, where each device can only communicate with its neighbors inside its assigned cluster. . . . 112
5.2 Comparison of the system performance on MNIST dataset in terms of testing accuracy for different values of SNR. . . . 126
5.3 Testing accuracy on MNIST dataset with different pruning methods. SNR is 14 dB. . . . 126
5.4 Testing accuracy on MNIST (a) and FMNIST (b) datasets with different non-i.i.d. factors. Pruning ratio is 50%. Value of SNR is 14 dB. . . . 127
5.5 Testing accuracy on MNIST datasets with different values of pruning ratios. Value of SNR is 12 dB. . . . 128
5.6 Comparison of testing accuracy between the proposed FL framework with other baselines. . . . 128

List of Tables

1.1 Comparison between 4G, 5G, and 6G communication systems. 2

2.1 Comparison between different gradient descent algorithms. 31

3.1 Comparison with other communication-efficient FL systems. Abbreviations in the table are defined as: "TP": Transmit Power, "TS": Training Step, "P/C": Precoder/Combiner, "QS": Quantization Scheme, "SP": Scheduling Policy, "RSN": Receiver Side Noise, "CF": Channel Fading, "IU": Interference from other users, "FP": Full Participation, "RS": Random Scheduling Policy, "CSI": Channel State Information-based Scheduling Policy, "l_2": l_2-norm-based Scheduling Policy. 39

3.2 Averaged testing accuracy of local user models at end of training when i.i.d. training dataset is used. 60

3.3 Averaged testing accuracy of local user models at end of training when non-i.i.d. training dataset is used. 61

4.1 Comparison with FD systems. Abbreviations in the table are defined as: "C": Convergence Analysis, "MH": Model Heterogeneity, $\mathcal{X}$ is the unlabeled public dataset that both local users and server have access to, "SD": Serve side Distillation, "LD": Local user side Distillation, "IE": Information Exchanged, $\mathcal{W}$ denotes that model weights are transmitted, $\mathcal{L}$ denotes that model output logits are transmitted, "N": Receiver Side Noise, "F": Channel Fading, "I": Interference from other users. 71

4.2 Model architectures 88

4.3 Converged testing accuracies for user groups with nonidentical model architectures when error-free and error-prone channels are considered with different degrees of non-i.i.d.-ness (α_d). CIFAR-10 + STL-10 dataset combination is utilized. 93

4.4 Converged testing accuracies for user groups with nonidentical model architectures when error-free and error-prone channels are considered with different degrees of non-i.i.d.-ness (α_d). MNIST + FMNIST dataset combination is utilized. 94

5.1 System running time of each FL scheme in seconds. 129

List of Abbreviations

5G	Fifth-generation
6G	Sixth-generation
AI	Artificial intelligence
AWGN	Additive white Gaussian noise
BC	Broadcast channel
BCE	Binary cross-entropy
BER	Bit error rate
BS	Base station
CAeC	Contextually agile eMBB communications
CFL	Clustered federated learning
CNN	Convolutional neural network
COC	Computation-oriented communications
CPU	Central processing unit
CSI	Channel state information
CV	Computer vision
D2D	Device-to-device
DL	Downlink
DNN	Deep neural network
DoF	Degree of freedom
EB	Exabytes
ECC	Error correction codes
EDuRLLC	Event-defined uRLLC
eMBB	Enhanced mobile broadband
FC	Fully connected
FD	Federated distillation

FL	Federated learning
FLOP	Floating point operations
FLOPs	Floating point operations per second
GD	Gradient descent
GPU	Graphics processing unit
HE	Homomorphic encryption
i.i.d.	Independent and identically distributed
IoT	Internet of things
KD	Knowledge distillation
KKT	Karush-Kuhn-Tucker
KL	Kullback-Leibler
LoS	Line of sight
LSM	Log-softmax
LSTM	Long short-term memory
MAC	Multiple access channel
MDE	Minimum-distortion embedding
MIMO	Multiple-input multiple-output
MISO	Multiple-input single-output
mMIMO	Massive multiple-input multiple-output
MMSE	Minimum mean square error
mMTC	Massive machine-type communications
MU-MIMO	Multi-user multiple-input multiple-output
NGAI	Next-generation air interface
NLP	Natural language processing
NOMA	Non-orthogonal multiple access
OFDMA	Orthogonal frequency-division multiple access
PDF	Probability density function
PSK	Phase-shift keying
QPSK	Quadrature phase shift keying
ReLU	Rectified linear unit
RWM	Random Waypoint Model
SBS	Small base station
SGD	Stochastic gradient descent

SIMO	Single-input multiple-output
SISO	Single-input single-output
SM	Softmax
SNR	Signal-to-noise ratio
TDMA	Time division multiple access
TPU	Tensor processing unit
UAV	Unmanned aerial vehicle
UE	User equipment
UL	Uplink
UMAP	Uniform manifold approximation and projection
uRLLC	Ultra-reliable and low-latency communications
ZF	Zero-forcing

List of Notations

a	Scalar
$\mathbf{a}$	Vector
$\mathbf{A}$	Matrix
$\mathcal{A}$	Set
$\|a\|$	Absolute value of scalar a
$\|\mathcal{A}\|$	Cardinality of set $\mathcal{A}$
$\mathbf{I}_K$	Identity matrix of size $K \times K$
$\mathbf{A}^T$	Transpose of matrix $\mathbf{A}$
$\mathbf{A}^H$	Hermitian of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	Inverse of matrix $\mathbf{A}$
$\|\mathbf{a}\|_2$	l_2-norm of vector $\mathbf{a}$
$tr\,(\mathbf{A})$	Trace of matrix $\mathbf{A}$
$(\cdot)^*$	Optimal solution
$\det(\mathbf{A})$	Determinant of matrix $\mathbf{A}$
$\mathbb{E}\{\cdot\}$	Expectation operator
$\mathbb{C}$	Set of complex numbers
$\mathbb{R}$	Set of real numbers
min	Minimize
max	Maximize
$\mathcal{B}(i)$	i^{th} element in the ordered set $\mathcal{B}$
$\mathcal{O}(\cdot)$	Order of magnitude
$[\mathbf{A}]_{ij}$	The entry at i^{th} row and j^{th} column of matrix $\mathbf{A}$
$\mathcal{N}(a,b)$	Real Gaussian random variable with mean a and variance b
$\mathcal{CN}(a,b)$	Complex Gaussian random variable with mean a and variance b
$\log_2$	Base 2 logarithm

$\log_{10}$	Base 10 logarithm
$\log$	Natural logarithm
$\nabla F(\mathbf{w})$	Gradient of function $F(\cdot)$ with respect to $\mathbf{w}$
$I(a;b)$	Mutual information
$h(\cdot)$	Entropy
$\mathcal{A} \setminus \mathcal{B}$	Set minus operation
$\mathcal{A} \cup \mathcal{B}$	Set union operation
$F_1(\cdot) \circ F_2(\cdot) \circ F_3(\cdot)$	Function composition
$\mathbf{AB} \rightarrow \mathbf{C}$	Matrix approximation
$\lfloor \cdot \rfloor$	Floor operation
$a >> b$	Much greater than

List of Symbols

e	Euler's number
$\mathbf{x}$	Transmit signal or the input of a training data sample
$\mathbf{y}$	Received signal or the output of a training data sample
$\mathbf{h}, \mathbf{H}$	Channel state information
$\mathbf{n}$	AWGN at receiver side
N_r, N_t	Numbers of receiving and transmitting antennas
σ^2	Noise variance
P_t	Transmit power
$\mathbf{R_{xx}}$	Covariance matrix of transmit signal $\mathbf{x}$
$\mathbf{R_{yy}}$	Covariance matrix of received signal $\mathbf{y}$
λ	Eigenvalues
C^D_{MIMO}	Deterministic MIMO capacity
C^E_{MIMO}	Ergodic MIMO capacity
K	Number of users in the MU-MIMO system
M	Number of antennas at BS for a MU-MIMO system
N	Number of antennas at each user for a MU-MIMO system
	Number of users in the considered systems in Chapter 3, 4, 5
$\mathbf{F}_{ZF}$	ZF precoder matrix
$\mathbf{F}_{MMSE}$	MMSE precoder matrix
g_{ZF}	Scaling factor for ZF precoder matrix
g_{MMSE}	Scaling factor for MMSE precoder matrix
$l_{BCE}(\cdot)$	Binary cross-entropy loss function
$l_{CCE}(\cdot)$	Categorical cross-entropy loss function
$l_{KL}(\cdot)$	Kullback-Leibler divergence loss function
β	Momentum factor

$F_n(\cdot)$[1]	Local training loss function for n^{th} user
$Q_n(\cdot)$	Local distillation loss function for n^{th} user
$F(\cdot)$	Global training loss function
α	Weighting factor for global loss function
α_d	Non-i.i.d. factor for Dirichlet distribution
S[2]	Number of training steps
R	Number of global communication round
$\mathcal{T}$	Set of time instances
$\mathcal{T}_{LT}, \mathcal{T}_{SD}, \mathcal{T}_{LD}$	Sets of time indices for local training, server, and local distillation
t, r, n	Index for time instance, global communication round and user
N_{BS}, N_D	Numbers of antennas at BS and user
$\mathbf{d}$	Training data sample
$\mathcal{D}_n, \mathcal{D}$	Local private dataset for n^{th} user and all users
D_n, D	Size of local private dataset for n^{th} user and all users
$\mathcal{X}, D_p$	Unlabeled public dataset and its size
$\phi_n(\cdot)$	Function characterising the impacts of overall distortion
$\phi_{n,DL}(\cdot)$	Function characterising the wireless impairments of DL
$\phi_{n,UL}(\cdot)$	Function characterising the wireless impairments of UL
$\phi_{av}(\cdot)$	Function characterising the operations taken at server
$\mathbf{v}$	Immediate result of one step SGD
$\tilde{\mathbf{v}}$	Velocity term in GD with momentum
$\mathbf{w}^*$	Optimal model weights
η	Learning rate
W	Number of model weights
$\mathbf{w}, \mathbf{W}$	Model weights
$\hat{\mathbf{w}}, \hat{\mathbf{W}}$	Estimated model weights
$\mathbf{z}, \mathbf{Z}$	Model output logits
$\hat{\mathbf{z}}, \hat{\mathbf{Z}}$	Estimated model output logits
σ_w^2	Variance of model weights
σ_z^2	Variance of effective noise in Chapter 3 and 5
	Variance of logits in Chapter 4
$\mathbf{Z}^{(UL)}, \mathbf{Z}^{(DL)}$	UL and DL AWGN

[1] $F_n(\cdot)$ also denotes the mapping between input data point and output logits but in Chapter 4 only.
[2] S also denotes the number of distillation steps but in Chapter 4 only.

P_{UL}, P_{DL}	Transmit power for UL and DL
Δ	Quantization step size
δ	Indicator function
L[3]	Smoothness function factor for $F_n(\cdot)$
ψ	Smoothness function factor for $L_n(\cdot)$
μ[4]	Convex function factor for training loss function
μ_q	Convex function factor for distillation loss function
σ_n^2	Factors of bounded SGD variance for training loss function
σ_{qn}^2	Factors of bounded SGD variance for distillation loss function
G^2	Factors of bounded gradients for training loss function
G_q^2	Factors of bounded gradients for distillation loss function
C	Number of classes in the dataset
f	Number of FLOP
E	Number of training epoch
f_n	Computing capability of n^{th} user's device
q	Number of quantization bits
l^{train}, l^{comm}	Training and communication latency
l^{comp}, l^{aggre}	Computation and aggregation latency
l^{total}	Total latency
a	Bandwidth fraction allocation factor
$\mathcal{V}$	Set of neighbors
W^p	Number of zero-valued weights in the model
ρ	Pruning ratio
$F_{en}(\cdot)$	Homomorphic encryption function
$F_{de}(\cdot)$	Homomorphic decryption function

[3] $L_n(\cdot)$ also denotes the local training loss function for n^{th} user but in Chapter 4 only.
[4] μ also denotes the chosen threshold for the water-filling algorithm but in Chapter 2 only.

“***In solitude, where we are least alone.***”

To my dearly departed friend,

66, this is for you.

Chapter 1

Introduction

1.1 Background

Wireless communication has revolutionized the way we connect and interact in the modern world, enabling the seamless transmission of data over long distances without the need for physical connections. From the early days of radio transmission to the sophisticated cellular networks and Wi-Fi systems we rely on today, wireless communication has become an integral part of our daily lives. It underpins a vast array of applications, including mobile telephony, satellite communication, and emerging technologies such as the Internet of Things (IoT). These advancements have facilitated unprecedented levels of connectivity and data exchange, resulting in an exponential increase in global mobile traffic. Fig. 1.1 illustrates the estimated total global mobile traffic in exabytes (EB) per month. It is projected that mobile data traffic will triple between 2023 and 2029.

The ongoing evolution of wireless communication technologies continues to push the boundaries of what is possible, offering faster speeds, lower latency, and more reliable connections, which are essential for supporting the growing demands of an increasingly interconnected world. With the rapid deployment of fifth-generation (5G) communication infrastructures and networks on a worldwide scale, both industry and academia have begun to transfer their research focus to beyond 5G and sixth-generation (6G) communications. In contrast to previous generations, 6G technology will be transformative, revolutionizing the shift from "connected things" to "connected intelligence". This evolution will impose increasingly demanding requirements [2]. These include the need for exceptionally high data rates, reaching up to 1Tbps, as well as the imperative for low-latency control, with end-to-end latency demands of less than 1 ms. Additionally, the operating spectrum of wireless networks is expanding significantly, spanning

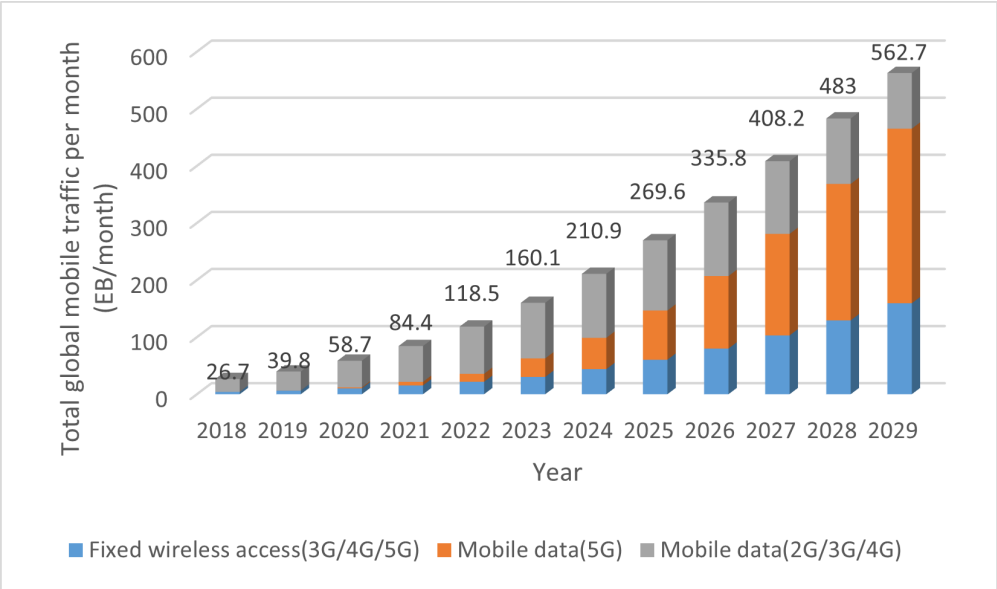


FIGURE 1.1: The estimated total global mobile traffic per month [1].

TABLE 1.1: Comparison between 4G, 5G, and 6G communication systems.

Aspects	4G	5G	6G
Data rate	1 Gbps	10 Gbps	1 Tbps
End-to-end latency	100 ms	10 ms	1 ms
Maximum frequency	6 GHz	90 GHz	3 THz
Maximum spectral efficiency	15 bps/Hz	30 bps/Hz	100 bps/Hz
THz communication	No	Very limited	Widely
Architecture	MIMO	Massive MIMO	Intelligent surface
AI	No	Very limited	Fully

from 73GHz to 140GHz and even extending to the terahertz range, from 1THz to 3THz. Table 1.1 shows a comparison between 4G, 5G, and 6G communication systems [3].

This evolution necessitates not only meeting more stringent technical benchmarks but also calls for a reevaluation and enhancement of the service types offered by the current 5G networks [2]. For instance, there is a transition from enhanced mobile broadband (eMBB) to contextually agile eMBB communications (CAeC), reflecting the increasing contextual adaptability demanded by modern applications. Similarly, the progression from ultra-reliable and low-latency communications (uRLLC) to event-defined uRLLC (EDuRLLC) underscores the importance of responsiveness to specific triggering events. Moreover, the shift from massive machine-type communications (mMTC) to computation-oriented communications (COC) signals a growing emphasis on computational capabilities within communication paradigms.

On the other hand, the past few years have also witnessed rapid growth in the field of Artificial Intelligence (AI). The global AI market, valued at $27.23 billion in 2019, is

projected to soar to $267 billion by 2027 [4]. This phenomenal growth underscores the increasing integration of AI technologies across various sectors such as computer vision (CV) [5], natural language processing (NLP) [6], and medical image analysis [7]. Several factors contribute to this rapid expansion. Firstly, advancements in computational power, particularly through the development of Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs) [8], have significantly accelerated the training and deployment of complex AI models. Secondly, the development of big data has provided the necessary fuel for machine learning algorithms to thrive [9], enabling more accurate and advanced insights. Additionally, the widespread availability of open-source AI frameworks such as TensorFlow [10] and PyTorch [11] has enabled user-friendly access to AI tools, fostering innovation and collaboration within the research community and industry.

Given the rapid advancements in AI, its integration into 6G communications is anticipated to be significantly more pervasive compared to 5G. In 6G networks, AI is expected to penetrate every aspect of heterogeneous and massive-scale networks; that is, achieving ubiquitous AI in 6G [2]. This means AI will be fundamental in managing and optimizing the complex and diverse network environments characteristic of 6G. The deployment of AI in 6G will enhance various network functions such as resource allocation, interference management, and predictive maintenance, enabling more efficient and adaptive communication systems. Furthermore, AI will play a critical role in enabling advanced features of 6G, such as ultra-low latency, high reliability, and massive connectivity. By leveraging AI, 6G networks will be able to dynamically adapt to user needs and environmental conditions, thereby providing seamless and intelligent connectivity.

Further, recently, the trend in distributed learning is pushing the deployment of training and computing towards the network edge. In this way, the processes of data acquisition and computation are decoupled, which is of interest to both academia and industry. Motivated by this, the emergence of FL also pushes to integrate two originally decoupled areas: machine learning and wireless communications. However, most of the current research in this direction is based on the assumption that the communication links are error-free, which is not realistic in practical scenarios since wireless networks are subject to various types of noises and interference, and all of them can affect the performance of FL algorithms. Therefore, it is important to investigate, redesign, and evaluate the FL algorithms as well as their variants to meet the specific requirements and challenges of future networks, including 6G. This will enable more reliable and efficient distributed learning systems, which are essential for the advanced applications and services for next-generation communication networks.

In conventional digital communication systems, error correction codes are indeed vital for ensuring reliable data transmission by mitigating the effects of channel noise. Even assuming that error correction codes (ECC) can completely eliminate communication errors and provide error-free transmission, there are still important considerations for why we investigate the direct impact of noise and wireless impairments on the transmitted weight values in FL. In FL, model updates are exchanged frequently over a large number of iterations, and even in the absence of bit errors, other factors can affect model convergence and performance.

First, quantization error during the transmission of weight values can introduce distortions, especially when compression techniques are employed to reduce the communication overhead. These quantization effects become more prominent in bandwidth-limited scenarios or when trading off between model accuracy and communication efficiency.

Second, while ECC can recover the data, it doesn't address the fundamental challenges posed by channel variations, such as fading, interference, or mobility-induced delays, which can indirectly affect the timeliness of model updates. In practical wireless environments, these factors introduce asynchronous updates and staleness in the model, potentially slowing down the convergence of the FL algorithm or leading to suboptimal global models.

Moreover, investigating the impact of noise and other impairments allows us to design FL systems that are robust to variability in resource constraints at edge devices, where the devices may have limited computational or energy capacity, leading to degraded performance under harsh conditions. By accounting for these factors, we aim to ensure that FL remains robust and efficient in diverse and challenging wireless environments, even when communication errors are corrected.

1.2 Contributions

Motivated by the potential offered by the FL paradigm, a huge amount of research has emerged in this area. Most existing studies focus on enhancing the FL algorithm from an "AI" perspective, such as designing or integrating more advanced machine learning techniques in the training framework to boost the model performance. However, as discussed previously, these studies often assume error-free links, neglecting the potential offered by advanced wireless communications. Therefore, in this thesis, we endeavor to address these critical gaps in the current literature by incorporating more "wireless communication" ingredients to the FL algorithms and their variants, i.e., investigating

FL algorithms under the wireless communication networks, which is essential for understanding their performance and reliability in practical scenarios. This includes designing and proposing communication-efficient variants of FL, conducting corresponding convergence analysis, evaluating their effectiveness in practical applications, etc. The specific contributions of this thesis are listed as follows:

- *FL in multi-user mMIMO communication systems:* We propose, implement, and evaluate a mMIMO-integrated FL framework, where a base station (BS) and multiple single-antenna users are included in a single cell. In the proposed FL framework, due to the imperfect communication links, wireless impairments such as channel fading, noise, and interference are imposed on the transmitted model weights. To address this, we utilize zero-forcing (ZF) and minimum mean square error (MMSE) precoding/combining techniques to null the intra-cell interference. Then, for the considered system, based on the statistical and stochastic properties of stochastic gradient descent (SGD) and loss functions, we conduct the corresponding convergence analysis by deriving the upper bound of the global loss function with respect to the communication round. Subsequently, motivated and underpinned by the insights provided by the derived bound, we endeavor to modify the algorithm and propose several communication-efficient FL variants, the modifications are made by varying the number of local training steps, transmit power, number of quantization bits, etc. Finally, to demonstrate both the effectiveness and efficiency of the proposed algorithms, we conduct extensive simulations on practical datasets and applications.

- *Federated Distillation (FD) in multi-user mMIMO communication systems:* We study a popular variant of FL algorithm, namely, Federated Distillation (FD) in the single-cell multi-user mMIMO communication systems. Unlike FL, in the considered framework, we exclusively adopt the model outputs, called logits, as the information exchanged between users and BS. We show how this change enables the system to accommodate nonidentical model architectures and benefit the training process when the computing capability is different across the local users, especially in a heterogeneous network. Then, we derive the upper bound on the loss function to prove the guaranteed convergence of the proposed algorithm by jointly considering the processes of training and distillation. Further, in the proposed system, the distillation process is performed at both BS and local user sides. We justify the advantages of this operation by explaining the additional logit manipulation that can be done at the BS side. Subsequently, accounting for the inconsistent channel quality among users, we design a communication channel-aware FD algorithm (FedTSKD-G), where the BS side model is aware of the effects

resulting from different channel imperfections while making predictions. Finally, in the simulations, we demonstrate the effectiveness of the proposed algorithm by comparing it to the existing baselines in the literature.

- *Fully distributed FL in D2D communication systems:* We examine a more general case of FL, namely, a fully distributed FL algorithm, where a BS or central server is not available and users can only exchange, share, and update their models with near neighbors. We consider a device-to-device (D2D) communication network. To enable accurate and efficient model weight exchange, in each global communication round, we utilize the K-means algorithm to perform dynamic clustering of users based on their geographical proximity. Further, to reduce communication overhead, we integrate the model pruning technique in the system to remove redundant model weights and ensure compact local model representations. Then, we mathematically analyze the computation and communication latency, and convergence of the proposed system. We also employ *Karush-Kuhn-Tucker* (KKT) conditions to jointly optimize the pruning ratio and bandwidth allocation, resulting in accelerated convergence rate, improved learning accuracy, and reduced computation and communication latency.

1.3 Thesis Overview

This thesis is organized as follows:

- Chapter 1 provides the motivation behind the research in this thesis, along with the contributions and organization of this thesis.
- Chapter 2 provides some background knowledge and concepts that underpin the research presented in this thesis.
- Chapter 3 investigates the FL algorithm in a single-cell multi-user mMIMO communication system, where the BS is equipped with multi number of antennas and users have a single antenna. We rigorously conduct the convergence analysis of the considered system. In order to enable communication efficiency, we also endeavor to propose several variants to reduced the total incurred communication overhead, based on the insights inferred from the convergence analysis.
- Chapter 4 focuses on FD, which is a variant of vanilla FL algorithm. We consider a dual-distillation framework, where distillation operation is taken at both the local user and server sides, and explain how this grants us more flexibility when designing the manipulation at BS side. Then, we prove the convergence of

the considered system and propose a group-based communication channel-aware mechanism.

- Chapter 5 develops a novel fully distributed FL algorithm with adaptive model pruning. We consider a serverless scenario where no BS is available and users can only share information with their neighbors. We utilize the model pruning technique to remove redundant model weights. Accounting for the communication and computing latency, KKT conditions are utilized to jointly optimize the pruning ratio and bandwidth allocation.

- Chapter 6 concludes this thesis and we suggest some potential directions for future work, based on the current research presented in this thesis.

1.4 List of Publications

- Journal papers:

  – Y. Mu, N. Grag, and T. Ratnarajah, "Federated learning in massive MIMO 6G networks: convergence analysis and communication-efficient design," *IEEE Transactions on Network Science and Engineering,* vol. 9, no. 6, pp. 4220-4234, Nov. 2022.

  – Y. Mu, N. Grag, and T. Ratnarajah, "Federated distillation in massive MIMO networks: dynamic training, convergence analysis and communication channel-aware learning," *IEEE Transactions on Cognitive Communications and Networking,* Mar. 2024.

  – Y. Mu, X. Liu, and T. Ratnarajah, "Adaptive model pruning in fully distributed federated learning," *submitted to IEEE journal.*

- Conference papers:

  – Y. Mu, N. Grag, and T. Ratnarajah, "Communication-efficient federated learning for massive MIMO systems," in *Proc. the IEEE Wireless Communications and Networking Conference*, Apr. 2022, pp. 578-583.

  – Y. Mu, N. Grag, and T. Ratnarajah, "Communication channel-aware federated distillation for massive MIMO systems," in *Proc. the IEEE 24th International Workshop on Signal Processing Advances in Wireless Communications*, Sep. 2023, pp. 436-440.

  – Y. Mu, X. Liu, and T. Ratnarajah, "Optimizing efficiency through adaptive model pruning in fully distributed federated learning," in *Proc. the IEEE 25th*

International Workshop on Signal Processing Advances in Wireless Communications, Sep. 2024, accepted.

Chapter 2

Background

2.1 Introduction

In this chapter, we provide a brief overview of the fundamental wireless communications and machine learning concepts that underpin the work presented in the rest of this thesis. First, we discuss point-to-point multiple-input multiple-output (MIMO) systems, underlining their inherent advantages and analyzing their corresponding capacity. Then, our focus shifts to multi-user systems, in particular, we introduce the concept of massive MIMO and examine its potential to substantially enhance spectrum efficiency while concurrently alleviating interference encountered in communication systems. Next, we transition to a machine-learning perspective, beginning with an introduction to prevalent architectures employed in deep learning, notably convolutional neural networks (CNNs). Following this, we offer a concise elaboration of the loss function, with particular emphasis on the cross-entropy and Kullback-Leibler Divergence Loss functions. Lastly, we present an overview of the optimization techniques utilized during training, including the gradient descent (GD) algorithm family and momentum technique.

2.2 MIMO Communications

Multiple-input multiple-output (MIMO) systems have emerged as a pivotal technology in wireless communications, leveraging the use of multiple antennas at both the transmitter and receiver to enhance communication performance. Traditionally, wireless systems relied on single-input single-output (SISO) configurations, utilizing a single antenna at each end of the communication link. However, the inherent limitations of SISO systems, such as susceptibility to multipath effects and fading, prompted the development of MIMO technology. By exploiting the spatial dimension [12] of the wireless

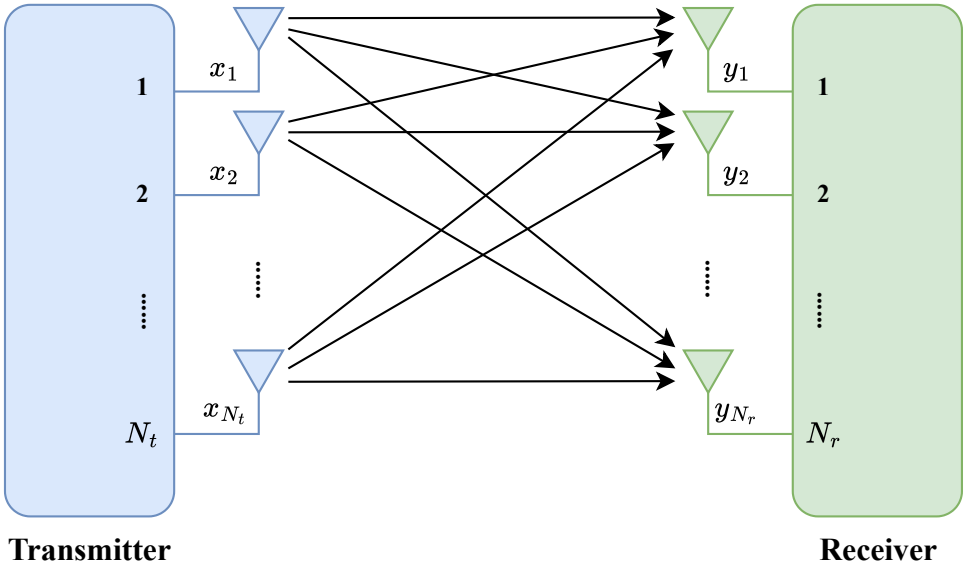


FIGURE 2.1: An illustration of point-to-point MIMO system with N_t transmitting antennas and N_r receiving antennas.

channel, MIMO systems capitalize on the effects of multipath propagation, where signals reflect off surrounding objects, leading to variations in amplitude, phase, and frequency. These variations, known as fading, can degrade signal quality and impede data transmission. Through spatial multiplexing and diversity techniques, MIMO systems enable simultaneous transmission of multiple data streams and enhance link reliability, thereby revolutionizing wireless communication capabilities.

2.2.1 Point-to-point MIMO systems

Consider a point-to-point MIMO communication system equipped with N_t and N_r antennas at the transmitter and receiver, respectively, as illustrated in Fig. 2.1. In this way, the received signal can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{2.1}$$

Eqn. (2.1) can also be expanded and interpreted as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N_r} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1N_t} \\ h_{21} & h_{22} & \dots & h_{2N_t} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_r1} & h_{N_r2} & \dots & h_{N_rN_t} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N_t} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{N_r} \end{bmatrix}, \tag{2.2}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_{N_t}]^T$ denotes the vector of signals, simultaneously transmitted by N_t antennas. $\mathbf{y} = [y_1, y_2, \ldots, y_{N_r}]^T$ represents the received signal with vector element y_{N_r} denoting the received signal at the ${N_r}^{th}$ antenna. $\mathbf{H}$ is the $N_r \times N_t$ channel matrix with entry $h_{ji} = [\mathbf{H}]_{ji}$ representing the channel gain from the i^{th} transmitting antenna to the j^{th} receiving antenna. For a rich scattering wireless environment, where there are a significant number of reflectors and no dominant propagation along the line of sight (LoS) path, $\mathbf{H}$ is usually modeled with Rayleigh fading, and the entries h_{ji} follows the distributed such that

$$h_{ji} \sim \mathcal{N}\left(0, \frac{1}{2}\right) + \sqrt{-1}\mathcal{N}\left(0, \frac{1}{2}\right). \tag{2.3}$$

The $N_r \times 1$ noise vector, $\mathbf{n}$, can be modeled as additive white Gaussian noise (AWGN), with independent and identically distributed (i.i.d.) Gaussian random variables entries n_i with zero mean and variance σ^2, i.e.,

$$n_i \sim \mathcal{N}\left(0, \frac{\sigma^2}{2}\right) + \sqrt{-1}\mathcal{N}\left(0, \frac{\sigma^2}{2}\right). \tag{2.4}$$

Compared to SISO systems, the main advantages of MIMO communication systems can be summarized as follows [12–14]

- *Spatial diversity gain:* In wireless communication systems, signal power fluctuations, or fading, are common occurrences at the receiver. Diversity serves as a method to mitigate these fading effects by providing the receiver with multiple independent copies of the transmit signal across different dimensions, such as frequency, time, or space. By increasing the number of independent copies, the likelihood of at least one copy experiencing minimal fading rises, consequently enhancing the quality and reliability of received signals. For SISO systems, only time or frequency diversity can be exploited, owing to the one-to-one nature of the link. However, MIMO systems introduce a new form of diversity called spatial diversity, enabled by the existence of multiple independent channel links. Therefore, such a system can achieve a spatial diversity of order $N_r \times N_t$, offering robustness against fading and ensuring more consistent signal reception across the communication channel.

- *Power gain:* Power gain refers to the increase in the average received signal-to-noise ratio (SNR). This can be achieved by utilizing advanced signal processing techniques at both ends of the communication link, namely transmitters and receivers. At the receiver, coherent combining of signals received from multiple antennas effectively improves the received power, resulting in an amplified SNR. Similarly, at the transmitter, intelligent power allocation strategies prioritize higher

quality links, thereby further boosting the received SNR. These strategies leverage Channel State Information (CSI) to optimize signal transmission. While obtaining CSI at the receiver end is relatively straightforward, acquiring accurate CSI at the transmitter poses greater challenges due to practical limitations and channel variability. This power gain not only improves the SNR but also extends the coverage of the wireless network by enhancing the system's resilience to noise.

- *Spatial multiplexing gain:* Spatial multiplexing gain in MIMO systems refers to the ability to transmit multiple independent data streams simultaneously using multiple transmit and receive antennas, without requiring additional power or bandwidth. This gain arises from the extra spatial dimension provided by the multiple antennas, allowing for enhanced data throughput. At high SNR regime, the capacity scales linearly with the number of spatial streams, also known as the degrees of freedom (DoF). The maximum achievable DoF is determined by the minimum of the number of transmit and receive antennas, denoted as $\min(N_r, N_t)$ for a $N_t \times N_r$ MIMO system. This capability greatly enhances the overall data transmission efficiency and spectral efficiency of the communication system, enabling higher data rates and improved performance in wireless networks.

It is worthwhile noting that, for a MISO (multiple-input single-output) or SIMO (single-input multiple-output) system, diversity and power gains can be achieved, but they do not provide DoF gain (multiplexing gain). MIMO systems have the potential to offer DoF gain, which is crucial in the high SNR regime.

2.2.2 Capacity of MIMO systems

As demonstrated in the previous section, the diversity and multiplexing gains offered by MIMO system enhance the communication throughput. In the following section, considering the schematic diagram shown in Fig. 2.1, we will derive the capacity of the MIMO system.

2.2.2.1 Deterministic capacity

Assume the channel matrix $\mathbf{H}$ in Eqn. (2.1) to be deterministic, the channel capacity is defined as the maximum achievable mutual information between the transmit signal $\mathbf{x}$ and the received signal $\mathbf{y}$ by varying the probability density function (PDF) of $\mathbf{x}$, which is expressed as [12, 15]

$$C^{D}_{MIMO} = \max_{tr(\mathbf{R_{xx}}) \leq P_t} I(\mathbf{x}; \mathbf{y}), \tag{2.5}$$

where P_t and $\mathbf{R_{xx}}$ represent the transmit power constraint and the covariance matrix of transmit signal $\mathbf{x}$, respectively. $I(\mathbf{x};\mathbf{y})$ is the mutual information between vectors $\mathbf{x}$ and $\mathbf{y}$, which can be given as

$$\begin{aligned} I(\mathbf{x};\mathbf{y}) &= h(\mathbf{y}) - h(\mathbf{y}|\mathbf{x}) = h(\mathbf{y}) - h(\mathbf{Hx}+\mathbf{n}|\mathbf{x}) \\ &\overset{(a)}{=} h(\mathbf{y}) - h(\mathbf{n}), \end{aligned} \tag{2.6}$$

where $h(\mathbf{y})$ denotes the entropy of $\mathbf{y}$ and $h(\mathbf{y}|\mathbf{x})$ represents the conditional entropy of $\mathbf{y}$ given $\mathbf{x}$. The result in (a) comes from the fact that $\mathbf{x}$ and $\mathbf{n}$ are two independent random vectors such that $h(\mathbf{y}|\mathbf{x}) = h(\mathbf{n})$. Since the entropy of noise, $h(\mathbf{n})$, is a constant, maximizing the mutual information in Eqn. (2.6) is equivalent to maximizing the entropy of the received signal $\mathbf{y}$. This requires both $\mathbf{y}$ and $\mathbf{x}$ to be zero-mean circular symmetric complex Gaussian. In this way, the entropy of $\mathbf{y}$ and $\mathbf{n}$ can be given as

$$h(\mathbf{y}) = \log_2\left[\det(\pi e \mathbf{R_{yy}})\right] \tag{2.7}$$

$$h(\mathbf{n}) = \log_2\left[\det(\pi e \sigma^2 \mathbf{I}_{N_r})\right], \tag{2.8}$$

where $\mathbf{R_{yy}}$ is the covariance of the received signal $\mathbf{y}$, and its expression is given as

$$\begin{aligned} \mathbf{R_{yy}} &= \mathbb{E}\{\mathbf{y}\mathbf{y}^H\} = \mathbb{E}\left\{(\mathbf{Hx}+\mathbf{n})(\mathbf{Hx}+\mathbf{n})^H\right\} \\ &= \mathbb{E}\left\{\left(\mathbf{Hx}\mathbf{x}^H\mathbf{H}^H\right) + \left(\mathbf{n}\mathbf{n}^H\right)\right\} \\ &= \mathbb{E}\left\{\left(\mathbf{Hx}\mathbf{x}^H\mathbf{H}^H\right)\right\} + \mathbb{E}\left\{\mathbf{n}\mathbf{n}^H\right\} \\ &= \mathbf{H}\left(\mathbb{E}\left\{\mathbf{x}\mathbf{x}^H\right\}\right)\mathbf{H}^H + \sigma^2\mathbf{I}_{N_r} \\ &= \mathbf{H}\mathbf{R_{xx}}\mathbf{H}^H + \sigma^2\mathbf{I}_{N_r}, \end{aligned} \tag{2.9}$$

where $\mathbf{R_{xx}}$ denotes the covariance matrix of transmitted signal $\mathbf{x}$. Then, with Eqn. (2.7), (2.8) and (2.9), the mutual information in Eqn. (2.6) can be calculated as

$$\begin{aligned} I(\mathbf{y};\mathbf{x}) &= \log_2\left[\det(\pi e \mathbf{R_{yy}})\right] - \log_2\left[\det(\pi e \sigma^2 \mathbf{I}_{N_r})\right] \\ &= \log_2 \det\left[\mathbf{I}_{N_r} + \frac{1}{\sigma^2}\mathbf{H}\mathbf{R_{xx}}\mathbf{H}^H\right], \end{aligned} \tag{2.10}$$

Therefore, considering Eqn. (2.5), for a deterministic MIMO channel, the maximum achievable capacity is given as

$$C^D_{MIMO} = \max_{tr(\mathbf{R_{xx}})\leq P_t}\left\{\log_2 \det\left[\mathbf{I}_{N_r} + \frac{1}{\sigma^2}\mathbf{H}\mathbf{R_{xx}}\mathbf{H}^H\right]\right\}. \tag{2.11}$$

In Eqn. (2.11), it can be witnessed that the MIMO capacity depends on the availability of the channel matrix $\mathbf{H}$, namely, the CSI. When the transmitter has no knowledge about

the CSI, the optimal transmission strategy is equal power allocation for each antenna [12], that is; $\mathbf{R_{xx}} = \frac{P_t}{N_t}\mathbf{I}_{N_t}$. In this case, the derived capacity of the MIMO channel is given as

$$C^D_{MIMO} = \log_2 \det \left[\mathbf{I}_{N_r} + \frac{P_t}{\sigma^2 N_t}\mathbf{H}\mathbf{H}^H\right], \tag{2.12}$$

where $\frac{P_t}{\sigma^2}$ is defined as the nominal SNR of the system. If we define $d = \min(N_t, N_r)$ and $\lambda_1, \lambda_2, \ldots, \lambda_d$ as the eigenvalues of $\mathbf{H}\mathbf{H}^H$, Eqn. (2.12) can be further expressed as

$$\begin{aligned} C^D_{MIMO} &= \log_2 \det \left[\mathbf{I}_{N_r} + \frac{P_t}{\sigma^2 N_t}\mathbf{H}\mathbf{H}^H\right] \\ &= \log_2 \prod_{i=1}^{d} \left(1 + \frac{P_t}{\sigma^2 N_t}\lambda_i\right) \\ &= \sum_{i=1}^{d} \log_2 \left(1 + \frac{P_t}{\sigma^2 N_t}\lambda_i\right). \end{aligned} \tag{2.13}$$

On the other hand, If the full CSI is available at the transmitter, the transmit power can be allocated adaptively to each antenna, based on the channel quality such that more power is allocated to the sub-channel with a higher gain. This method is known as water-filling algorithm [12] and it is considered as an effective technique for improving system capacity.

2.2.2.2 Ergodic capacity

In the previous section, we assume the channel matrix $\mathbf{H}$ to be deterministic. However, in wireless communications, it is more realistic to consider MIMO channels to be time-varying due to the effects of fading. Accordingly, $\mathbf{H}$ is considered to be a random matrix. In this way, consider the result we have derived in Eqn. (2.11), when MIMO channels are assumed to be time-varying, the channel capacity is calculated by averaging overall all channel realizations as

$$C^E_{MIMO} = \max_{tr(\mathbf{R_{xx}}) \leq P_t} \mathbb{E}\left\{\log_2 \det \left[\mathbf{I}_{N_r} + \frac{1}{\sigma^2}\mathbf{H}\mathbf{R_{xx}}\mathbf{H}^H\right]\right\}. \tag{2.14}$$

For an i.i.d. Rayleigh fading channel model, the optimal transmission scheme is obtained by equal power allocation among all transmit antennas, i.e., $\mathbf{R_{xx}} = \frac{P_t}{N_t}\mathbf{I}_{N_t}$, and the resulting achievable capacity is given as

$$C^E_{MIMO} \approx \sum_{i=1}^{d} \mathbb{E}\left\{\log_2 \left(1 + \frac{P_t}{\sigma^2 N_t}\lambda_i\right)\right\}. \tag{2.15}$$

At high SNR regime, this achievable capacity can be approximated as

$$C^{E}_{MIMO} \approx \sum_{i=1}^{d} \mathbb{E}\left\{\log_2\left(\frac{P_t}{\sigma^2 N_t}\lambda_i\right)\right\}$$
$$= d\log_2 \mathrm{SNR} + \sum_{i=1}^{d} \log_2 \frac{\mathbb{E}\{\lambda_i\}}{N_t}, \tag{2.16}$$

where $\mathrm{SNR} = \frac{P_t}{\sigma^2}$. The pre-log factor d in Eqn. (2.16) suggests that at a high SNR regime, the full d DoF is attained. In contrast, for a SISO (single-input single-output) system, the corresponding approximation in a high SNR regime is $\log_2 \mathrm{SNR}$. Therefore, it is observed that the channel capacity of MIMO system can be considered as a combination of d individual SISO channels, at high SNR regime.

Similarly, under low SNR assumption, the MIMO capacity can be approximated as

$$C^{E}_{MIMO} \overset{(a)}{\approx} \sum_{i=1}^{d} \frac{\mathrm{SNR}}{N_t}\mathbb{E}\left(\lambda_i\right)\log_2 e = \frac{\mathrm{SNR}}{N_t}\mathbb{E}\left\{tr\left\{\mathbf{H}\mathbf{H}^H\right\}\right\}\log_2 e$$
$$= \frac{\mathrm{SNR}}{N_t}\mathbb{E}\left\{\sum_{i,j}|h_{ij}|^2\right\}\log_2 e = \frac{\mathrm{SNR}}{N_t}N_t N_r \log_2 e$$
$$= N_r \mathrm{SNR}\log_2 e, \tag{2.17}$$

where the result in (a) comes from the approximation that $\log_2(1+x) \approx x\log_2 e$, for small value of x. For a SISO system, the corresponding approximation at low SNR is $\mathrm{SNR}\log_2 e$. Eqn. (2.17) indicates that, in the low SNR regime, a $N_t \times N_r$ MIMO system yields power gain of N_r over a SISO system. It also suggests that, under low SNR assumption, we only need multiple receive antennas (N_r), that is; SIMO systems.

It can also be concluded from Eqn. (2.17) that in the low SNR regime, the system's capacity increases linearly with power rather than logarithmically, as it does in high SNR case (Eqn. (2.16)). This implies that in the low SNR approximation, the power gain is crucial, and adding more power to the system has a direct and significant impact on the capacity. The linear relationship between capacity and power indicates that MIMO systems offer a strong power gain at low SNR.

Further, it can also be witnessed that the capacity linearly depends on the number of receive antennas (N_r), which is a key indicator of diversity gain. On the other hand, in the low SNR regime, the multiplexing gain is diminished. This is because the capacity is dominated by power and diversity gains, rather than by the ability to transmit multiple spatial data streams. The system operates in a mode where adding more spatial streams

provides little benefit, as the available power is too limited to support independent data streams effectively.

Regarding the CSI at the receiver and transmitter, in the low SNR regime, perfect CSI at the receiver is typically assumed. The receiver uses this information to coherently detect the transmitted signal, which is vital for achieving the ergodic capacity, especially in the presence of fading. However, the impact of CSI at the transmitter is significantly reduced compared to high SNR situation. At low SNR, capacity is primarily limited by power rather than by the ability to perform spatial multiplexing. While perfect CSI at the transmitter is typically used for power allocation strategies like water-filling (discussed in the previous section), in the low-SNR regime, such techniques offer little improvement. The system is power-limited, not bandwidth- or multiplexing-limited, so uniform power allocation across all antennas is almost as effective as optimal power allocation strategies.

This also explains why in the Eqn. (2.17), the low SNR capacity approximation only depends on the number of receiver antennas N_r and not on the number of transmit antennas N_t. In the low SNR regime, the system operates in a power-limited condition, where the main challenge is detecting the weak received signal. As a result, capacity is determined primarily by the amount of signal power that can be captured by the receive antennas, which is proportional to N_r. Increasing the number of receive antennas improves the ability to gather and process incoming signal energy. In contrast, at low SNR, the multiplexing gain, which depends on N_t, becomes negligible. The low available power prevents the system from supporting multiple independent data streams effectively, meaning that even with multiple transmit antennas, the system remains constrained by its power limitations.

2.2.3 Multi-user MIMO (MU-MIMO) systems

In Fig. 2.1, we explore the dynamics of a point-to-point MIMO system, featuring a single transmitter and receiver configuration. However, the complexity of practical wireless networks extends beyond this simplistic model, often comprising multiple transmitters and receivers interacting simultaneously. This complexity gives rise to multi-user MIMO (MU-MIMO) systems, which include many of MIMO technologies tailored for modern wireless communication scenarios. In MU-MIMO systems, multiple users, equipped with either single or multiple antennas, are simultaneously served by a multiple-antenna base station (BS), accommodating varying antenna configurations at both ends of the communication link. This evolution from point-to-point to MU-MIMO architectures reflects

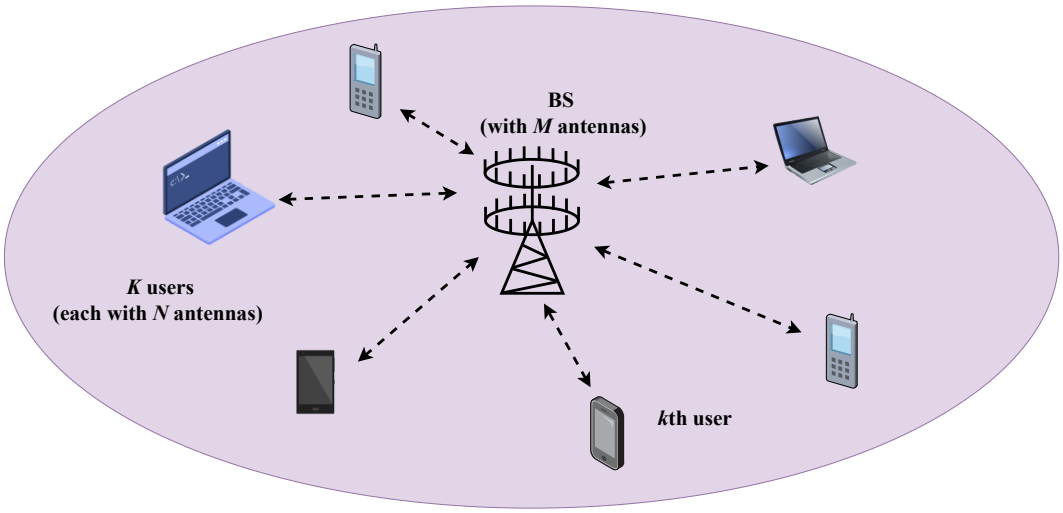


FIGURE 2.2: An illustration of a multi-user MIMO system that includes K users, where the BS is equipped with M antennas and each user is equipped with N antennas.

the growing demand for enhanced spectral efficiency and increased network capacity in wireless communications.

In a MU-MIMO system, the channels for downlink (DL) and uplink (UL) communication are commonly referred to as the broadcast channel (BC) and multiple access channel (MAC), respectively. Illustrated in Fig. 2.2 is a depiction of a multi-user MIMO communications system, where a BS equipped with M antennas engages with K independent users, each equipped with N antennas. This configuration effectively forms a virtual set of $K \times N$ antennas for the users, establishing an end-to-end scenario between the BS and users, and such configuration can be considered as a $M \times KN$ MIMO system for downlink transmission, or an $KN \times M$ MIMO system for uplink transmission. Particularly, in scenarios where $N = 1$, that is; the users are equipped with single antenna, the end-to-end BS-user configuration can be viewed as a $M \times K$ or $K \times M$ MIMO setup in the DL or UL, respectively. This system configuration indicates the adaptability of multi-user MIMO systems to cater to diverse communication requirements.

2.2.3.1 MU-MIMO downlink (DL)

For a single-cell MU-MIMO communication system, consider $[x_1, x_2, \ldots, x_M]^T$ as the DL transmit signal from the BS to K users, then, the $N \times 1$ received signal at a typical k^{th} user, $\mathbf{y}_k$ is given as

$$\mathbf{y}_k = \mathbf{H}_k^{DL}\mathbf{x} + \mathbf{n}^{DL}, \tag{2.18}$$

where $N \times M$ matrix $\mathbf{H}_k^{DL}$, with $k = 1, 2, \ldots, K$ denotes the DL channel between the k^{th} user and the BS. $\mathbf{n}_k^{DL}$ is a $N \times 1$ vector containing DL noise at user side. Consequently,

considering all K users, the overall received signals can be represented in a more compact way as

$$\underbrace{\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \vdots\\ \mathbf{y}_K\end{bmatrix}}_{NK\times 1} = \underbrace{\begin{bmatrix}\mathbf{H}_1^{DL}\\ \mathbf{H}_2^{DL}\\ \vdots\\ \mathbf{H}_K^{DL}\end{bmatrix}}_{NK\times M} \underbrace{\mathbf{x}}_{M\times 1} + \underbrace{\begin{bmatrix}\mathbf{n}_1^{DL}\\ \mathbf{n}_2^{DL}\\ \vdots\\ \mathbf{n}_K^{DL}\end{bmatrix}}_{NK\times 1}. \tag{2.19}$$

2.2.3.2 MU-MIMO uplink (UL)

Consider a typical k^{th} user, let $\mathbf{x}_k = [x_1, x_2, \ldots, x_N]^T$ be its UL transmit signal to the BS. Then, we can write the $M \times 1$ received signal at the BS, $\mathbf{y}^{UL}$ from all K users as

$$\begin{aligned}\mathbf{y}^{UL} &= \sum_{k=1}^{K} \mathbf{H}_k^{UL}\mathbf{x}_k + \mathbf{n}^{UL}\\ &= \mathbf{H}_1^{UL}\mathbf{x}_1 + \mathbf{H}_2^{UL}\mathbf{x}_2 + \ldots + \mathbf{H}_K^{UL}\mathbf{x}_K + \mathbf{n}^{UL}\\ &= \underbrace{[\mathbf{H}_1^{UL}, \mathbf{H}_2^{UL}, \ldots, \mathbf{H}_K^{UL}]}_{M\times NK} \underbrace{\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\\ \vdots\\ \mathbf{x}_K\end{bmatrix}}_{NK\times 1} + \underbrace{\mathbf{n}^{UL}}_{M\times 1}.\end{aligned} \tag{2.20}$$

From Eqn. (2.19) and (2.20), it can be witnessed that MU-MIMO systems encounter challenges posed by co-channel interference resulting from the simultaneous transmissions to different users. Therefore, effective measures must be taken to mitigate this interference, particularly in the downlink (DL) links. One such crucial strategy involves preprocessing the transmit signal, known as precoding, to nullify the co-channel interference and optimize transmission performance. Further discussion on this topic is provided in the next section.

2.2.3.3 Precoding technique

Unlike traditional point-to-point MIMO systems, where only one transmitter and one receiver are considered, MU-MIMO systems require sophisticated signal processing techniques to handle inter-user interference and efficiently allocate resources among multiple users. In this section, we will consider a single-cell downlink transmission with K users and one BS. Each user is considered to be equipped with a single antenna, i.e., $N = 1$.

The number of antennas at BS is M. The received signal at users is given as

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{bmatrix}}_{K\times 1} = \underbrace{\begin{bmatrix} \mathbf{h}_1^{DL} \\ \mathbf{h}_2^{DL} \\ \vdots \\ \mathbf{h}_K^{DL} \end{bmatrix}}_{\mathbf{H},\text{ of size } K\times M} \underbrace{\mathbf{s}}_{M\times 1} + \underbrace{\begin{bmatrix} n_1^{DL} \\ n_2^{DL} \\ \vdots \\ n_K^{DL} \end{bmatrix}}_{K\times 1}, \tag{2.21}$$

where $\mathbf{h}_k^{DL}$ denots the $1 \times M$ DL channels for k^{th} user. $\mathbf{s} = g\mathbf{F}\mathbf{x}, \mathbf{x} \sim \mathcal{CN}(0, P_t\mathbf{I}_K)$. Here, g denotes the normalizing factor to ensure the transmit power constraint, that is; $\mathbb{E}\left\|\mathbf{s}\right\|^2 = P_t$. $\mathbf{F}$ denotes the precoding matrix, which can be either linear [16] or non-linear [17]. Non-linear precoding method is usually able to achieve better performance in terms of capacity and interference suppression. However, it requires more computational complexity. In this thesis, for the considered system, we mainly adopt the linear precoding method, whose main design idea is to cancel out the operation of channel matrix $\mathbf{H}$. Consider channel inversion (or zero-forcing, ZF), the preceding matrix $\mathbf{F}_{ZF}$ is given as

$$\mathbf{F}_{ZF} = \mathbf{H}^H \left(\mathbf{H}\mathbf{H}^H\right)^{-1}, \tag{2.22}$$

and the corresponding scaling factor is expressed as

$$g_{ZF} = \frac{1}{\sqrt{tr\left[\left(\mathbf{H}\mathbf{H}^H\right)^{-1}\right]}}. \tag{2.23}$$

Note that the above scaling factor is valid for phase-shift keying (PSK) modulation and the proof of Eqn. (2.23) is derived as follows

$$\begin{aligned} P_t &= \mathbb{E}\left\|\mathbf{s}\right\|^2 = tr\left[\mathbb{E}\left\{\mathbf{s}^H\mathbf{s}\right\}\right] \\ &= g_{ZF}^2 tr\left[\mathbb{E}\left\{\mathbf{x}^H\mathbf{F}_{ZF}^H\mathbf{F}_{ZF}\mathbf{x}\right\}\right] \overset{(a)}{=} g_{ZF}^2 tr\left[\mathbb{E}\left\{\mathbf{x}^H\mathbf{x}\right\}\left(\mathbf{H}\mathbf{H}^H\right)^{-1}\right] \\ &= g_{ZF}^2 P_t tr\left[\left(\mathbf{H}\mathbf{H}^H\right)^{-1}\right], \end{aligned} \tag{2.24}$$

the result in (a) comes from the fact that $\mathbf{F}^{\mathbf{H}}{}_{ZF}\mathbf{F}_{ZF} = \left(\mathbf{H}\mathbf{H}^H\right)^{-1}\mathbf{H}\mathbf{H}^H\left(\mathbf{H}\mathbf{H}^H\right)^{-1}$. The issue with the ZF method arises during its application in the UL transmission, particularly when performing ZF combining at the BS side. In this scenario, the ZF process inadvertently amplifies the noise power along with the desired signals. The minimum mean square error (MMSE) method reduces this noise effect by including the noise power into the calculation of the precoding matrix, whose expression is derived as

$$\mathbf{F}_{MMSE} = \mathbf{H}^H \left(\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}\right)^{-1}, \tag{2.25}$$

and the corresponding scaling factor is

$$g_{MMSE} = \frac{1}{\sqrt{tr\left[\mathbf{H}\mathbf{H}^H\left(\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}\right)^{-2}\right]}}. \tag{2.26}$$

2.2.4 Massive MIMO

After elaborating on the MU-MIMO system, we now delve into the domain of massive MIMO (mMIMO). While MU-MIMO systems involve a moderate number of antennas at base stations (BS) serving multiple users simultaneously, massive MIMO takes a leap forward by deploying a significantly larger number of antennas at the BS. This augmentation promises to innovate wireless communication networks by offering unprecedented gains in spectral efficiency, coverage, and energy efficiency. Thus, the transition from MU-MIMO to massive MIMO marks a shift towards harnessing the immense potential of large-scale antenna arrays for enhanced wireless connectivity.

In a mMIMO system, the BS is usually equipped with hundreds or even thousands of antennas to serve a comparably small number of users, i.e., $M >> K$. With such a large antenna array, compared with MU-MIMO systems, the advantages of mMIMO systems can be summarized as follows [18, 19]

- *Enhanced spectral efficiency and system capacity:* With a large number of antennas, massive MIMO systems can support a larger number of concurrent users and data streams, thereby exploiting the spatial diversity to achieve higher spectral efficiency and increasing the overall system capacity compared to MU-MIMO systems.
- *Improved coverage and penetration:* The deployment of a massive number of antennas enables more efficient interference management, resulting in enhanced signal coverage and penetration. Consequently, massive MIMO systems are capable of serving users in challenging environments such as urban areas and indoor spaces more effectively than MU-MIMO systems.

In summary, mMIMO can be considered as an "upgraded" version of MU-MIMO, aiming at capitalizing on all the benefits offered by MU-MIMO, but on a larger scale. In Fig. 2.3, we consider a single-cell multi-user mMIMO system. The number of users K is set to 15 and different numbers of antennas are considered at both the BS and user sides. We utilize Quadrature Phase Shift Keying (QPSK) for modulation. For the i.i.d. Rayleigh fading channel, CSI is considered to be known at both the transmitter and

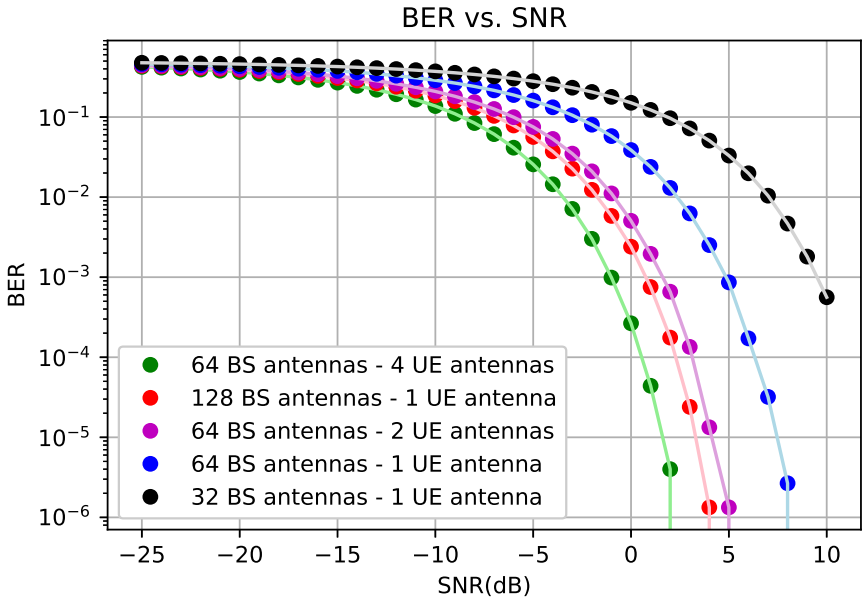


FIGURE 2.3: BER vs. SNR for a single-cell mMIMO system with ZF precoding (DL transmission with QPSK modulation)

receiver sides. We plot the bit error rate (BER) against SNR for the DL transmission with ZF precoding utilized at BS. Comparing the performance depicted by the black, blue, and red curves (assuming users with a single antenna and BSs equipped with 32, 64, and 128 antennas, respectively), it is evident that as the number of BS antennas increases, the BER demonstrates a decreasing trend, indicating improved system performance. Furthermore, comparing the blue, violet, and green curves (with a fixed number of 64 BS antennas and different numbers of user antennas), it becomes apparent that an increase in antennas at the user equipment (UE) enhances receiving diversity, consequently leading to enhanced system performance.

In the Chapter 3 and 4 of this thesis, we will mainly consider the multi-user mMIMO scenarios, where each user is either equipped with single or multiple antennas.

2.3 Machine/Deep Learning Techniques

In this section, we will give a brief introduction regarding the machine/deep learning techniques that underpin the research in this thesis. Machine learning, a subset of artificial intelligence (AI), comes from the assumption that machines are able to learn patterns and insights from data and make intelligent predictions or decisions, enabling them to learn from past computations or adapt to their environment through experiential learning.

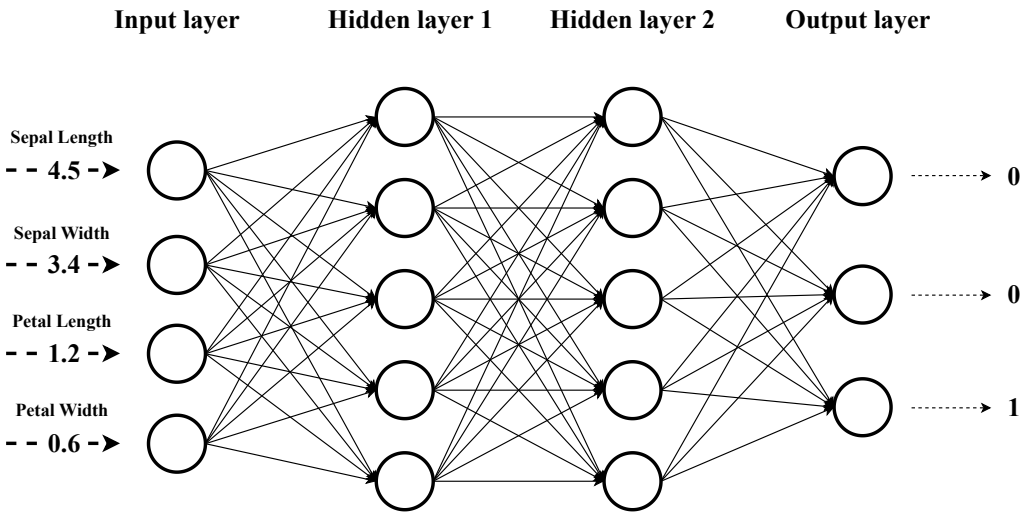


FIGURE 2.4: An example of deep neural network (DNN) for Iris flower classification problem, in this case, the model maps the four features (sepal length, sepal width, petal length, and petal width) to the category of the flowers.

In traditional machine learning approaches, feature engineering plays a crucial role, requiring manual selection and crafting of relevant features from raw data to input of the model. However, deep learning models offer a distinct advantage by enabling automatic extraction of hierarchical representations and features directly from raw data, such as images or audio signals. This capability arises from the multi-layered architecture of deep neural networks (DNN), allowing them to learn increasingly complex features as data passes through the network. As a result, deep learning models eliminate the need for manual feature engineering, making them particularly well-suited for tasks involving large, high-dimensional datasets. An example of DNN can be found in Fig. 2.4.

DNNs possess a remarkable capability, as elaborated by the universal approximation theorem, which asserts their ability to approximate any nonlinear function [20]. One of the main reasons behind this is the integration of activation functions within DNN architectures. These functions introduce nonlinearity into the model, enabling it to capture underlying relationships and effectively learn the complex mappings between inputs and outputs. For the case of Fig. 2.4, the model maps the year of birth to the month and date of birth of this thesis's author. It is worthwhile noting that the depth of a neural network correlates with its ability to capture intricate relationships within the data. It is necessary to have deeper architecture as the complexity of the underlying patterns increases. This phenomenon comes from the hierarchical nature of feature extraction in neural networks, wherein each successive layer learns increasingly abstract representations of the input data. Thus, to effectively model complex relationships, it is important to employ neural networks with a sufficient number of hidden layers, allowing for the gradual refinement and abstraction of the learned features.

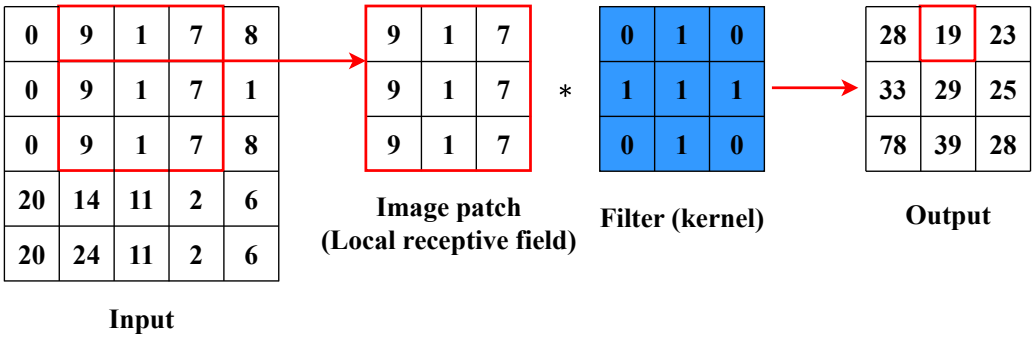


FIGURE 2.5: Convolution operation between an image and a 3×3 filter (kernel), the stride is set to 1. After the convolution, the image patch is mapped to 19, as shown in the output.

2.3.1 Convolutional neural networks (CNNs)

A Convolutional Neural Network (CNN) [21] is a specialized form of DNN designed for analyzing data with grid-like structures, such as images. It can automatically extract and learn spatial patterns inherent in these data by leveraging the arrangement of pixels within an image. CNN draws inspiration from biological processes, where neuron connections mirror the structure of the visual cortex. Each cortical neuron reacts to stimuli within a limited area called the receptive field. These receptive fields of neurons overlap, collectively covering the entire visual field.

For a typical CNN model, the following components are included

- *Convolutional layer:* These layers are the core building blocks of CNNs. They apply convolution operations to the input data, effectively extracting features through the use of learnable filters or kernels. These filters slide across the input data, capturing spatial patterns and features. An illustration of convolution operation is depicted in Fig. 2.5. It is noted that Fig. 2.5 only demonstrates the convolution computation of a single filter. However, in practice, a convolutional layer typically comprises numerous filters, each tasked with detecting specific patterns or features within the input data. By employing multiple filters parallelly and independently, the convolutional layer can effectively produce a set of feature maps. To introduce nonlinearity, outputs of the convolutional layer will also go through an activation layer, such as rectified linear unit (ReLU), sigmoid, and hyperbolic tangent functions (tanh).

- *Pooling layer:* Pooling layers are often interspersed between convolutional layers. They serve to downsample the feature maps generated by the convolutional layers, and they are crucial in CNNs for the following reasons [22, 23]

  - *Dimension reduction:* Pooling reduces the spatial dimensions (width and height) of the feature maps, which helps in controlling the number of parameters and computational complexity in the network. In Chapter 5 of this thesis, we will focus on another deep learning technique known as model pruning. This approach, similar to pooling, aims to reduce network complexity. However, instead of altering spatial dimensions, model pruning identifies and eliminates insignificant model weights to achieve network simplification.
  - *Translation invariance:* Pooling creates a summarized version of the features, making the network less sensitive to the precise location of features in the input data. This enhances the model's ability to recognize patterns regardless of their exact position in the input.
  - *Feature Generalization:* By aggregating similar features within a neighborhood, pooling layers help to capture the most important information while discarding irrelevant details. This promotes feature generalization and robustness against variations in the input.
  - *Computation Efficiency:* By reducing the size of feature maps, pooling reduces the computational cost of subsequent layers, allowing the network to process larger volumes of data more efficiently.

  Common pooling operations include max pooling and average pooling. An illustration of these operations can be found in Fig. 2.6.

- *Fully connected layer:* After passing through multiple convolutional and pooling layers, at the latter stages of a CNN, the extracted features are flattened and fed into one or more fully connected (FC) layers. The FC layer serves a crucial role in feature aggregation and classification, converting the spatial information extracted by the preceding layers into a compact representation suitable for decision-making. In other words, FC layers can be considered as the "interface" between the previous hidden layers and the output layer, providing the necessary transformation to map the learned features to the desired output space.

  Moreover, FC layers introduce flexibility in neural network design. By adjusting the size and structure of these FC layers, we can tailor the network architecture to suit the specific requirements of the task. In Chapter 4, we will show how we leverage this inherent characteristic of FC layers to design a self-aware framework that is able to adapt to different channel conditions.

Additionally, CNN architectures may include other components such as normalization layers and dropout layers to enhance model performance and robustness. In Chapter 3 and 5, the model trainable weights (mainly from the convolutional and FC layers) are

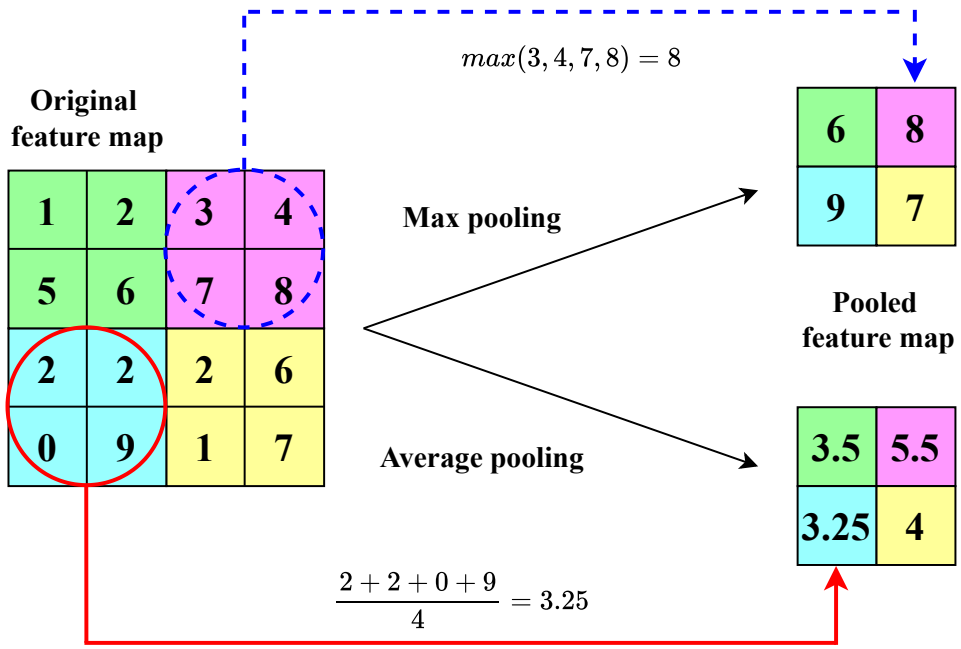


FIGURE 2.6: Max and average pooling operations between an original feature map and a 2×2 filter, the stride is set to 2.

considered as the information exchanged through wireless communication links, while in Chapter 4, the outputs of the FC layer, namely logits, are utilized for transmission.

2.3.2 Loss function

With the architecture of CNNs established, the next critical aspect in the training process is the choice and implementation of appropriate loss functions. While CNNs are capable of automatically extracting hierarchical features from data, their effectiveness in various applications heavily relies on the optimization of model parameters. In this regard, selecting suitable loss functions becomes important as they guide the network towards learning the desired representations and making accurate predictions.

Loss functions are also known as cost functions or objective functions. They provide a quantitative measure of how well the model is performing on the training data. This is achieved by quantifying the difference between predicted outputs and true labels, allowing the network to adjust its learnable parameters through training optimization algorithms (which will be discussed in the following section) to minimize this difference.

2.3.2.1 Binary cross-entropy loss

The binary cross-entropy (BCE) loss function, often used in binary classification tasks, calculates the discrepancy between the predicted probability and the actual binary labels.

If we have D samples in total in our training set, then the value of BCE, denoted by l_{BCE}, calculated over all the training samples can be given as

$$l_{BCE}(y, \hat{y}) = -\frac{1}{D}\sum_{i=1}^{D}\left[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)\right], \tag{2.27}$$

where y_i represents the true label of the i^{th} sample, (taking the value of either 1 or 0), $\hat{y}_i$ denotes the predicted probability that i^{th} sample belongs to class 1.

BCE loss functions offer several advantages and disadvantages that are important to consider in machine learning tasks. The simplicity and effectiveness of BCE make it a popular choice for binary classification problems. However, its direct application is limited to tasks with only two classes, and it may not generalize well to multi-class classification tasks. Additionally, BCE loss is also sensitive to imbalanced datasets, where one class significantly outnumbers the other. In such cases, the model may become biased towards the majority class, leading to suboptimal performance, particularly in terms of recall for the minority class. The imbalance in the dataset can skew the optimization process, causing the model to prioritize minimizing the loss for the majority class at the expense of the minority class. Consequently, the model may struggle to accurately classify samples belonging to the minority class, resulting in a degraded model performance.

2.3.2.2 Categorical cross-entropy loss

The categorical cross-entropy loss function is a widely used metric in classification tasks, especially when dealing with multi-class classification problems.

Consider a dataset with D training samples and C classes in total, then the categorical cross-entropy function can be expressed as

$$l_{CCE}(y, \hat{y}) = -\frac{1}{D}\sum_{i=1}^{D}\sum_{j=1}^{C} y_{ij} \log \hat{y}_{ij}, \tag{2.28}$$

where y_{ij} is the indicator function that takes on value 1 when i^{th} sample belongs to j^{th} class and 0 otherwise. $\hat{y}_{ij}$ signifies the predicted probability that the i^{th} sample belongs to the j^{th} class.

It is worthwhile noting that, when using categorical cross-entropy loss in practice, it's commonly paired with the softmax activation function after the final FC layer of the neural network. Softmax is used to convert the raw output scores from the network into meaningful probability values, ensuring they sum up to 1, representing a probability distribution over the classes. This is essential for interpreting the network's output as class probabilities. Then, the output of the softmax function is used in Eqn. (2.28), that is

$$\hat{y}_{ij} = \text{softmax}(\mathbf{z}_i)_j = \frac{e^{z_{ij}}}{\sum_{m=1}^{C} e^{z_{im}}}, \tag{2.29}$$

where $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iC})$ is the vector of raw output scores (logits) from the neural network for each class, given the i^{th} sample. Through this thesis, in Chapter 3, 4, and 5, we will utilize categorical cross-entropy loss as the training loss function.

2.3.2.3 Kullback-Leibler divergence loss

Kullback-Leibler Divergence (KL Divergence), also known as relative entropy, is a measure of how one probability distribution diverges from a second, reference probability distribution. Mathematically, it can be expressed as

$$l_{KL}(P \parallel Q) = \sum_{i} P(i) \log \frac{P(i)}{Q(i)}, \tag{2.30}$$

where $P(i)$ and $Q(i)$ represent the probability of occurrence of event i under distributions P and Q, respectively. KL Divergence Loss is often used in scenarios where the objective is to learn a distribution that approximates a given target distribution. One of the key properties of KL Divergence is that it is non-negative and becomes zero if and only if the two distributions P and Q are identical. In Chapter 4 of this thesis, we will use this metric as the distillation loss function and endeavor to train the model when the ground-truth labels are not available.

2.3.3 Training optimization algorithms

Training optimization is the process of adjusting the neural network parameters iteratively to minimize the loss function and it aims to find the optimal set of parameters that minimize the difference between predicted outputs and actual labels, thus improving the model's performance on the training data.

2.3.3.1 Batch gradient descent

Gradient descent (GD) is a first-order iterative optimization algorithm for finding the minimum of the loss function. At each training step, GD calculates the gradients of the loss function with respect to the model's trainable parameters. These gradients indicate the direction of steepest ascent in the loss landscape. To minimize the loss, GD updates the model parameters by taking a step in the opposite direction of the gradients, scaled by a hyperparameter called learning rate, that is,

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla F(\mathbf{w}_t), \tag{2.31}$$

where $\mathbf{w}_t$ is the model trainable weights at t^{th} step. η is the learning rate or the step size, reflecting the "speed" at which the loss function moves towards a minimum and $\nabla F(\mathbf{w}_t)$ denotes the gradient of loss function at $\mathbf{w}_t$.

In batch gradient descent (batch GD), at each step t, the gradient of the loss function with respect to the model parameters, $\nabla F(\mathbf{w}_t)$, is calculated by taking the average of gradients of all the training samples available in the training dataset, that is,

$$\nabla F(\mathbf{w}_t) = \frac{1}{D} \sum_{i=1}^{D} \nabla F(\mathbf{w}_t, \mathbf{d}_i), \tag{2.32}$$

where $\mathbf{d}_i$ denotes the i^{th} training sample in the training dataset. This mean gradient is used in Eqn. (2.31) to update the model parameters. It can be noticed that, in batch gradient descent, the model parameters are updated only after the processing of the entire training dataset, resulting in a single training step per epoch. Here, an "epoch" refers to a complete pass through all training samples. At the same time, since all the training samples are processed simultaneously, batch gradient descent is able to provide an unbiased and precise estimate of the gradient. Additionally, for convex optimization problems, batch gradient descent is guaranteed to converge to a global minimum of the loss function.

While batch gradient descent provides an accurate estimate of the gradient, its reliance on processing the entire training dataset in each iteration can be computationally intensive, especially when large datasets are considered. This can result in slow convergence and increased computational costs. Additionally, batch gradient descent is prone to overfitting, particularly when the model's capacity exceeds the complexity of the dataset. In such cases, the model may learn to fit the training data too closely, leading to poor generalization performance on unseen data.

2.3.3.2 Stochastic gradient descent (SGD)

Stochastic gradient descent (SGD) is a variant of gradient descent that addresses some of the limitations of batch gradient descent, particularly in terms of computational efficiency and generalization. Unlike batch gradient descent, which computes the gradient of the loss function using the entire training dataset, SGD updates the model parameters after processing each individual training sample. This means that instead of waiting until the entire dataset is processed before updating the parameters, SGD updates the parameters more frequently, leading to faster convergence and D training steps per epoch, where D denotes the number of training samples in the dataset. In this way, the updating process in Eqn. (2.31) can be rewritten as

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla F(\mathbf{w}_t, \mathbf{d}_i), \tag{2.33}$$

One of the key advantages of SGD is its ability to handle large datasets and non-convex optimization problems more efficiently. By updating the parameters based on individual samples, SGD can escape from local minima and saddle points more easily, potentially leading to better convergence properties. Moreover, the random shuffling of training samples in each epoch introduces stochasticity into the optimization process, which can help the algorithm explore different regions of the parameter space and avoid getting stuck in suboptimal solutions. However, the stochastic nature of SGD can also introduce noise into the optimization process, which can lead to oscillations in the loss function.

Considering binary cross entropy loss function, we provide a case example as follows. As explained in the previous section, the binary cross-entropy function for one data sample is denoted by

$$l_{BCE}(y, \hat{y}) = -\left[y \log \hat{y} + (1-y) \log(1-\hat{y})\right], \tag{2.34}$$

where y represents the true label of the sample, (taking the value of either 1 or 0), $\hat{y}$ denotes the predicted probability that the sample belongs to class 1. The binary cross entropy loss function is often used jointly with the sigmoid function. This means that $\hat{y} = \frac{1}{1+e^{-z}}$ with logits $z = \mathbf{w}^T\mathbf{x}$, where $\mathbf{x}$ is the input feature vector and $\mathbf{w}$ is the weight factor.

To calculate the gradient of l_{BCE} with respect to the weights $\mathbf{w}$, we use the chain rule as follows.

$$\frac{\partial l_{BCE}}{\partial \mathbf{w}} = \frac{\partial l_{BCE}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial \mathbf{w}} = \frac{\hat{y} - y}{\hat{y}(1-\hat{y})} \hat{y}(1-\hat{y})\mathbf{x} = (\hat{y} - y)\mathbf{x}. \tag{2.35}$$

Therefore, plugging Eqn. (2.35) into Eqn. (2.33) gives us

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta(\hat{y} - y)\mathbf{x}. \tag{2.36}$$

2.3.3.3 Mini-batch gradient descent

Mini-batch gradient descent (mini-batch GD) presents a compromise between the computational inefficiency of batch GD and the high variance of SGD by processing a small subset, or mini-batch, of the training data in each training step. Correspondingly, Eqn. (2.31) can be updated as

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla F(\mathbf{w}_t, \mathcal{D}_t), \tag{2.37}$$

where $\nabla F(\mathbf{w}_t, \mathcal{D}_t) = {}^{1}/_{|\mathcal{D}_t|} \sum_{\mathbf{d}_i \in \mathcal{D}_t} \nabla F(\mathbf{w}_t, \mathbf{d}_i)$ denotes the estimation of gradient on $\mathcal{D}_t$, which is a subset of the overall training dataset $\mathcal{D}$. In mini-batch GD, each epoch contains $\frac{D}{|\mathcal{D}_t|}$ training steps. It can also be witnessed that SGD and batch GD can be viewed as special cases of mini-batch gradient descent when the batch size $|\mathcal{D}_t|$ is set to one and D, respectively.

Mini-batch GD inherits the stability of updates from batch GD while mitigating the high variance of updates in SGD. By computing gradients on mini-batches rather than individual samples, mini-batch GD produces more stable updates to the model parameters, leading to smoother convergence trajectories. This stability is crucial for achieving robust optimization in the presence of noisy gradients, especially in scenarios where the training data may contain outliers or noisy samples.

Another advantage of mini-batch GD is its ability to generalize well across different datasets and architectures. By updating the model parameters based on mini-batches that are representative of the entire dataset, mini-batch GD produces updates that capture the underlying structure of the data more accurately than SGD. This helps prevent overfitting and improves the model's ability to generalize to unseen data, making mini-batch GD a preferred optimization algorithm for training deep learning models.

In Table 2.1, we compare the aforementioned batch GD, SGD, and mini-batch GD. The comparison is made for processing speed, memory usage, stability of updates, etc. Through this thesis, we will utilize the statistic and stochastic properties introduced by these algorithms to conduct our convergence analysis.

2.3.3.4 GD with momentum

As discussed in the previous section, both SGD and mini-batch GD suffer from instability of updates and susceptibility to local minima. One potential enhancement technique to

TABLE 2.1: Comparison between different gradient descent algorithms.

Aspects	Batch GD	SGD	Mini-batch GD
Processing speed	Slow	Fast	Moderate to fast
Memory usage	High	Low	Moderate
Stability of updates	Stable	High variance	Moderate variance
Convergence speed	Slow	Fast	Moderate to fast
Generalization	Prone to overfitting	Moderate	Moderate to good
Handling large datasets	Inefficient	Efficient	Efficient
Computation complexity	High	Low	Moderate
Robustness to noise	High	Low	Moderate

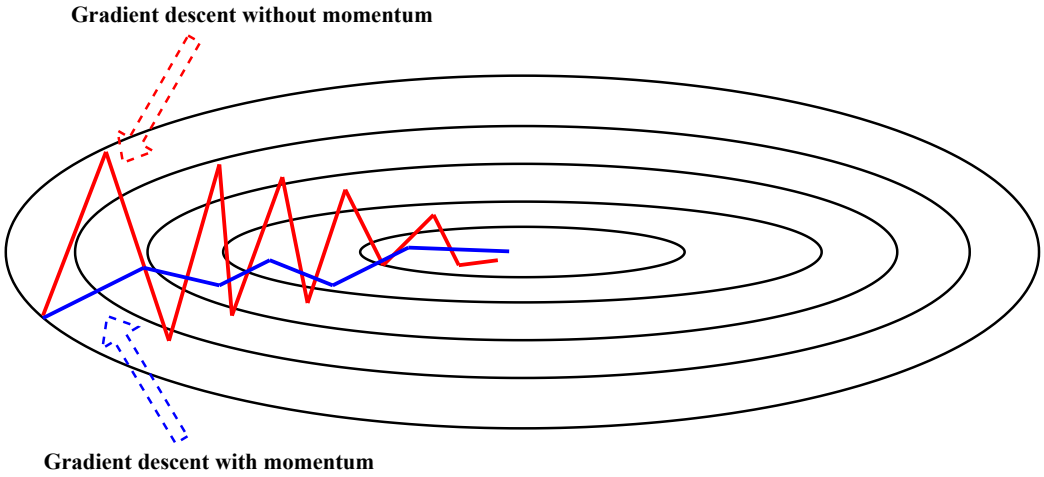


FIGURE 2.7: Gradient descent without momentum (red curve) vs. gradient descent with momentum (blue curve).

improve these convergence properties is to incorporate a memory of previous gradients and add an additional term, called "momentum", to the parameter updates.

Specifically, the momentum term is a moving average of the past gradients, which helps the algorithm to build up speed in directions with consistent gradients, and mitigate the oscillations in directions with high curvature. The update rule for gradient descent with momentum can be expressed as:

$$\tilde{\mathbf{v}}_t = \beta \tilde{\mathbf{v}}_{t-1} + (1-\beta)\nabla F(\mathbf{w}_t), \tag{2.38}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \tilde{\mathbf{v}}_t, \tag{2.39}$$

where $\tilde{\mathbf{v}}_t$ denotes the velocity term at t^{th} training step and β is the momentum factor, ranging from 0 to 1. In Fig. 2.7, we plot the updating trajectory of GD without and with momentum (in red and blue curves, respectively) to illustrate how the momentum helps to accelerate convergence and reduce the oscillations.

2.4 Chapter Summary

In the first part of this chapter, we focus on MIMO communication systems. We first give a brief review of the basic concept of the point-to-point MIMO system, underlining its advantages in terms of spatial diversity, power, and multiplexing gains. Next, we derive the deterministic and ergodic capacity of this MIMO system, respectively. Then, we extend our discussion to the multi-user MIMO (MU-MIMO) systems, emphasizing the necessity of doing precoding in this system and introducing two linear precoding techniques, namely, zero-forcing (ZF) and minimum mean square error (MMSE). Finally, we conclude our discussion on MIMO systems with a discussion on massive MIMO (mMIMO) systems, where a large number of antennas are equipped at BS side, stressing its potential to enhance system capacity, spectral, and energy efficiency.

In the second part of this chapter, we move to a machine/deep-learning perspective to introduce the techniques that underpin the research presented in this thesis. First, we provide a brief comparison between the concept of machine learning and deep learning, underlining the ability offered by deep learning to automatically extract underlying patterns and features of the data. Then, we briefly introduce the working mechanism of deep neural networks (DNNs), with a special focus on convolutional neural networks (CNNs) and their major components. Subsequently, we include the concept of loss function by introducing several popular examples used in the classification task, namely cross entropy and Kullback-Leibler divergence loss functions. Finally, we conclude this part by discussing the gradient descent (GD) algorithms family, introducing their variants and limitations in the training process, and how these limitations can be mitigated with the help of the momentum technique.

Chapter 3

Federated Learning in Massive MIMO Networks

3.1 Introduction

As stated in Chapter 1, rapid evolution in the field of machine and deep learning during the past few years has had a great impact on our daily lives. AI is set to be a key enabler in future wireless communication systems. By optimizing resource management, energy usage, and real-time processing, AI will enhance network performance and support ultra-low latency, crucial for emerging technologies like autonomous vehicles, smart cities, and IoT. Its integration will be vital for reducing the growing complexity of future communication demands and unlocking new possibilities in industries such as healthcare, immersive Augmented Reality (AR), and beyond [2]. For the proper working of a machine/deep learning-based application, it requires a powerful machine with huge computational resources, and a significant amount of data in order to learn reliable models, which not only incurs the higher costs but also damages the privacy while sharing users' data. Recently, federated learning (FL) [26] models have been studied, where models are learned in a distributed manner with limited cooperation without sharing data. Thus, compared with traditional "one node learning", where a central server/device collects all the data needed and performs training, federated learning has the benefits of utilizing local users' computing resources, such as graphics processing unit (GPU) on a local user's PC or central processing unit (CPU) on a smartphone. Since FL is a privacy-preserving algorithm [27–30], each user's dataset never leaves its local device. This is especially helpful in privacy-critical scenarios, where local users' data

Work in this chapter has been published in IEEE Transactions on Networking Science and Engineering, November 2022 [24], with a preliminary version presented at IEEE WCNC April 2022 [25].

contains sensitive or personal information that they may not be willing to share, e.g., medical images, different users' keyboard-typed letters, browser and searching history, etc. Another advantage of the FL algorithm in a practical communication setting is the requirement of significantly fewer physical resources (transmit power and bandwidth), which are limited in a typical wireless system.

Regarding FL methods, averaging based FL algorithm (FedAvg) was first proposed in [26] and was mathematically analyzed in [26] and [31], where local users run several local stochastic gradient descent steps in parallel on their local training sets, respectively and send the trained model weights to the central server. The server aggregates the received weights by averaging or combining linearly, followed by broadcasting of the updated global model weights back to users for further training. Such iterations continue until the weights converge. Detailed discussion on this process will be provided in the later sections of this chapter.

The rest of this chapter is organized as follows. Section 3.2 provides a review of the existing literature. Section 3.3 details the proposed system model. The communication process included in the proposed framework and convergence analysis are given in Section 3.4. Simulation results are presented in Section 3.5. Section 3.6 concludes this chapter.

3.2 Related Work

As highlighted in Chapter 1, compared with 5G, in future communication systems, penetration of AI into every aspect of heterogeneous and massive-scale networks is required; that is, achieving ubiquitous AI in next-generation communication systems [2]. With its privacy-preserving nature, FL has been recognized as one of the most promising solutions to resolve and fulfill the challenges in future communication networks [32]. In this way, works like [33–38] investigate FL framework under a 5G or 6G scenario. Among these, the data privacy leakage issues are given in [33] and [34], while authors in [35–38] provide the key technical challenges that FL may face in 6G scenarios. Meanwhile, these works have also demonstrated that the advantages provided by 6G, such as high performance networking, hyper-flexible architecture and AI-enhanced security [38], are also vital for accomplishing the requirements of FL frameworks in future communication networks; that is, future communication systems and FL can benefit each other mutually.

In the direction to integrate FL algorithms with communication-efficient frameworks under different wireless network settings, a general discussion can be seen in [39]. These design investigations can be categorized as follows.

3.2.1 Quantization schemes

This includes the effective quantization scheme for model weights in order to optimize the number of symbols for transmissions. The work in [40] considers a layered quantization scheme for downlink transmission, where various quantization gains are used to quantize the weights of different layers inside a deep learning model. Similarly, lattice quantization [41], sparsification [42], ternary gradients [43] and sparse ternary compression [44] are employed. These quantization schemes show their effectiveness in making the FL framework efficient. Among those, sparse ternary compression [44] outperforms others since it not only compresses both uplink and downlink streams with high compression rate, but also stays robust to the scenario where non-i.i.d. local users data is considered.

3.2.2 Resource allocation

Optimizing the communication resources, such as transmit and local computation energy consumption, is investigated in [45–50]. In [45], the loss function is minimized with the constraints on packet loss as well as bandwidth of wireless medium, and optimal transmit power for each user is derived. The work in [46] decomposes the non-convex energy optimization problem into several sub-problems and derives corresponding closed-form solutions. The convergence analysis of the framework proposed in [46] was later conducted by authors in [47]. In [48], a local computation latency constraint was considered in the optimization, and it was shown that training completion time in [48] was less than that in [46]. Two communication protocols namely, non-orthogonal multiple access (NOMA) and time division multiple access (TDMA) were considered in [49] to optimize the transmission rate and CPU frequency of local user devices. In [50], a resource-constrained FL framework for heterogeneous data is proposed and corresponding convergence analysis is also provided. Another resource allocation problem is to design the scheduling policies to allow a subset of the users to update the model such that the overall cost of resources can be reduced. In [51], an age-based scheduling policy has been proposed, which accounts for the recency of updates and the instantaneous channel quality of users. Proportional fair based scheduling policy is considered in [52], where users are selected based on signal-to-noise ratio (SNR). In [53], A greedy algorithm

for scheduling the devices is proposed by analyzing the trade-off between local computation, communication latencies and whole FL system learning time. In [54], scheduling problem in FL is investigated under a unmanned aerial vehicles (UAVs) communication scenario. A joint optimization problem, which includes device selection, UAVs placement, and resource management, is formulated and solved to speed up the convergence. Similarly, [55] investigates user scheduling problem in a UAV-assisted FL framework. In this work, based on the multi-dimensional contract incentive design, a subset of the users will be selected by UAV in each global communication round. In [56], differences between transmitting model weights and the weights updates are compared analytically and in simulations.

3.2.3 Imperfect links

Many of the existing literature [57–60] assume only UL links to be imperfect and error-prone, while considering DL links to be error-free; that is, the fact that the aggregated model being broadcasted in DL links also experiences fading and interference, has not been taken into consideration in the existing works. In this chapter, we focus on the massive MIMO scenario, where both the UL and DL links are error-prone, which is of interest, since our results show that the reliability of DL-links has higher impact on the performance of FL algorithm.

3.2.4 Contributions

In this chapter, we consider a massive MIMO network consisting of a base station (BS) and multiple single antenna users. In this system, we integrate FL framework, where the parameters of deep learning models are updated at both BS (or server) side and local user side. In FL, at regular intervals, weights are transmitted via UL channels, updated by aggregation and transmitted back via DL channels. The frequency of this update interval for transmitting the weights can vary depending on the system's configuration and the specific application. Typically, it is defined based on a trade-off between communication overhead and model performance. In some cases where low-latency and real-time data processing are necessary, such as autonomous driving and real-time navigation in vehicle-to-vehicle (V2V) communication, the update interval frequency should be carefully selected to achieve the end-to-end latency of 1 ms [61]. In our system, we explore how varying the frequency of update interval can accelerate training while maintaining performance. The main contributions of this chapter can be summarized as follows.

- *FL with massive MIMO 6G network*: We design and implement a FL framework with massive MIMO 6G communication network, considering imperfections in both UL and DL links. Employing zero-forcing (ZF) and minimum mean square error (MMSE) precoding/combining methods, we analyze the communication performance of this system and compare it with two simulation baselines in the existing literature.

- *Simpler convergence proof and tighter bound*: For the system considered, we derive the upper bound on the convergence rate, which is tighter than that of FedAvg. The bound also provides interesting insights about UL and DL communications. Our analysis shows that SNR in UL-DL communications can be reduced at the lower iterations. This leads to improved power consumption and is also analyzed. It can also be concluded that the reliability of DL links is more important than that of UL links. These analytical conclusions are verified via simulations.

- *Communication-efficient system design*: Motivated by the convergence analysis, several variants of the proposed FL framework with different degrees of communication-efficiency are designed and implemented where we adjust local training steps, transmit power, scheduling policy, and average codeword length after quantization. These modifications could be leveraged to reduce the total communication cost incurred during the training process. Detailed analysis and corresponding simulation results are provided, and communication cost comparison is also presented.

We consider image classification problem with MNIST, EMNIST and FMNIST datasets for deep federated learning, both i.i.d. and non-i.i.d. training dataset distributions are considered. Extensive simulations are carried out to include the effects of quantization, modulation, number of antennas, uplink and downlink SNR, ZF and MMSE precoding/combining, scheduling policy, fading and noise. Performance of the proposed system is also evaluated at different values of non-i.i.d. factor, which is a metric that quantitatively evaluates the non-i.i.d.-ness of users' training dataset distributions and will be explained in details later in Section 3.5. The connection between our theoretical and simulation results is highlighted. We compare our simulation results with two baselines: *1)* Both communication links (DL and UL) are assumed to be error-free. *2)* Only the UL is considered to be error-prone; that is, DL is assumed to be error-free. It is demonstrated that with UL-DL imperfections, FL converges similar to the convergence of baseline, with a very small constant gap. In Table 3.1, we compare our communication-efficient FL framework with existing works in the literature, the comparison is made for convergence analysis, simulation settings, types of noise considered for communication, techniques used for improving communication-efficiency, datasets

used for evaluations, etc. It can be witnessed that most of the work listed above try to improve communication-efficiency of a FL framework from only one perspective. On the other hand, some of the existing work investigate on multiple aspects, such as the work in [44] (quantization scheme and scheduling policy) and the work in [45, 57, 58] (transmit power and scheduling policy). However, the importance of DL transmission is not included or highlighted in the corresponding convergence analysis (the work in [45] considers the imperfections in DL channel, but non-i.i.d. dataset distribution is not considered and the proposed framework is only evaluated on one dataset).

TABLE 3.1: Comparison with other communication-efficient FL systems. Abbreviations in the table are defined as: "TP": Transmit Power, "TS": Training Step, "P/C": Precoder/Combiner, "QS": Quantization Scheme, "SP": Scheduling Policy, "RSN": Receiver Side Noise, "CF": Channel Fading, "IU": Interference from other users, "FP": Full Participation, "RS": Random Scheduling Policy, "CSI": Channel State Information-based Scheduling Policy, "l_2": l_2-norm-based Scheduling Policy.

Work	Convergence Analysis	Non-i.i.d. Dataset	Design Focus					Uplink			Downlink			Scheduling				Datasets			
			TP	TS	P/C	QS	SP	RSN	CF	IU	RSN	CF	IU	FP/RS	CSI	l_2	Others	MNIST	EMNIST	FMNIST	Others
[40]	✓	✓	–	–	–	✓	–	✓	–	–	✓	–	–	✓	–	–	–	✓	–	–	✓
[41]	✓	–	–	–	–	✓	–	✓	–	–	✓	–	–	✓	–	–	–	–	–	–	✓
[42]	–	–	–	–	–	✓	–	✓	–	–	✓	–	–	✓	–	–	–	✓	–	–	✓
[43, 62]	✓	–	–	–	–	✓	–	✓	–	–	–	–	–	✓	–	–	–	✓	–	–	✓
[44]	✓	✓	–	–	–	✓	✓	✓	–	–	✓	–	–	✓	–	–	✓	✓	–	✓	✓
[45]	✓	–	✓	–	–	–	✓	✓	✓	✓	✓	✓	✓	✓	–	–	✓	✓	–	–	–
[47, 48]	✓	✓	✓	–	–	–	–	✓	✓	–	–	–	–	✓	–	–	–	✓	–	–	✓
[49]	✓	–	✓	–	–	–	–	✓	✓	✓	–	–	–	✓	–	–	–	✓	–	–	–
[51]	–	–	–	–	–	–	✓	–	✓	–	–	–	–	✓	✓	–	✓	✓	–	–	–
[52]	✓	–	–	–	–	–	✓	✓	✓	✓	–	–	–	✓	–	–	✓	✓	–	–	–
[53]	–	✓	–	–	–	–	✓	✓	✓	–	–	–	–	✓	–	–	✓	✓	–	–	–
[58]	✓	✓	✓	–	–	–	✓	✓	✓	✓	–	–	–	✓	–	–	✓	✓	–	–	–
[57]	✓	✓	✓	–	–	–	✓	✓	✓	✓	–	–	–	✓	✓	–	✓	✓	–	–	–
[63]	✓	✓	–	–	✓	–	–	✓	✓	✓	–	–	–	✓	✓	–	–	✓	–	–	–
[64]	–	✓	–	–	–	–	✓	–	–	–	–	–	–	✓	–	–	✓	–	✓	–	✓
Proposed	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	✓	–

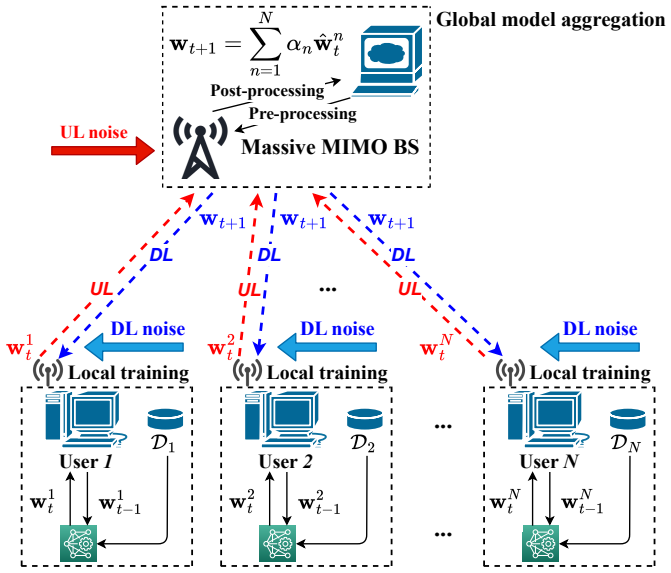


FIGURE 3.1: Architecture of the federated learning framework with imperfect wireless links, where channel fading, intra-cell interference, quantization noise and receiver side noise are considered.

3.3 System Model

We consider a wireless single-cell multi-user cellular 6G network, in which N users and a server or base station (BS) are included, as illustrated in Fig. 3.1. Details about the procedures presented in Fig. 3.1 are given in the subsequent subsections. The BS has N_{BS} number of antennas, while each user device is equipped with N_D antennas. The n^{th} user stores a local training dataset $\mathcal{D}_n = \{\mathbf{d}_i\}_{i=1}^{D_n}$ with size denoted by D_n. Each training data point $\mathbf{d}$ is stored as input-output pair $\{\mathbf{x}, y\}$, where $\mathbf{x}$ is the input to the model and y denotes the corresponding output. In this way, the total dataset across users is denoted by $\mathcal{D} = \cup_{n=1}^{N} \mathcal{D}_n$ with size $D = \sum_{n=1}^{N} D_n$. We also assume all users' datasets are not independent and identically distributed (non-i.i.d.). Let $f(\mathbf{w}, \mathbf{d})$ denote the loss function to measure the loss between the model output based on $\mathbf{w}$ and labeled data point $\mathbf{d}$. For the n^{th} user, the local loss function is calculated over all its local training data as

$$F_n(\mathbf{w}) \coloneqq \frac{1}{D_n} \sum_{\mathbf{d} \in \mathcal{D}_n} f(\mathbf{w}, \mathbf{d}). \tag{3.1}$$

The overall loss function for the data $\mathcal{D}$ can be defined as

$$F(\mathbf{w}) \coloneqq \frac{1}{D} \sum_{n=1}^{N} \sum_{\mathbf{d} \in \mathcal{D}_n} f(\mathbf{w}, \mathbf{d}) = \sum_{n=1}^{N} \alpha_n F_n(\mathbf{w}), \tag{3.2}$$

where $\alpha_n = {}^{D_n}/_{D}$ is the weight of the n^{th} device such that $\alpha_n \geq 0, \forall n$ and $\sum_{n=1}^{N} \alpha_n = 1$. The optimization problem for finding optimal model parameters $\mathbf{w}^*$ such that minimizes

the overall loss function can be given as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} F(\mathbf{w}). \tag{3.3}$$

Therefore, the BS minimizes the overall loss $F(\mathbf{w})$, while each user optimizes its local loss function $F_n(\mathbf{w})$. In this chapter, we use cross entropy as local training loss function. Accounting the imperfections in UL-DL links, the major procedures of the FL algorithm for a typical (say t^{th}) communication round can be described as follows.

3.3.1 Local training

Let S denote the number of training steps in one global communication round, and $\mathcal{T} = \left\{S_1, S_1 + S_2, \ldots, \sum_{r=1}^{R} S_r\right\}$ denote the time instances for each round. Then, the instances for local updates are given as $\mathcal{S} = \left\{0, 1, \ldots, S_1, \ldots, S_1 + S_2, \ldots, \sum_{r=1}^{R} S_r\right\} \backslash \mathcal{T}$, where R is the total number of communication rounds. In literature, $\mathcal{T}$ is selected uniformly, i.e., $S_r = S, r = 1, \ldots, R$, However, we set non-uniformly based on the training dataset and the communication errors.

At the start of t^{th} round, each n^{th} user initializes the local weights by the received model from BS, sent via DL channels in the $(t-1)^{th}$ round. After initialization, each n^{th} user performs local stochastic gradient descent (SGD) steps using its local dataset as

$$\mathbf{w}_{t+1}^n = \mathbf{w}_t^n - \eta_t \nabla F_n(\mathbf{w}_t^n, \xi_t^n), t \in \mathcal{S}, \tag{3.4}$$

where η_t is learning rate; ξ_t^n denotes a mini-batch, that is a subset chosen uniformly and independently from n^{th} user's local training dataset $\mathcal{D}_n$. This local training process is also presented in Fig. 3.1. At the end of S steps, the locally trained model, $\mathbf{w}_t^n$, is evaluated on the local testing set, and the averaged testing accuracy over all users is defined as the local testing accuracy for the t^{th} round.

3.3.2 Uploading model weights

Each user uploads the local trained model $\mathbf{w}_t^n$ to the central server or BS through UL channel (red dashed line in Fig. 3.1). For uplink transmissions, each user quantizes the local weights and employs suitable modulation (e.g., QPSK or QAM). Through the multi-path fading channel, the BS receives the noisy and distorted version of weights $\mathbf{w}_t^n, \forall n = 1, \ldots, N$. This uplink process can be represented using the mapping $\phi_{n,UL}\left(\mathbf{w}_t^n, \forall n\right)$.

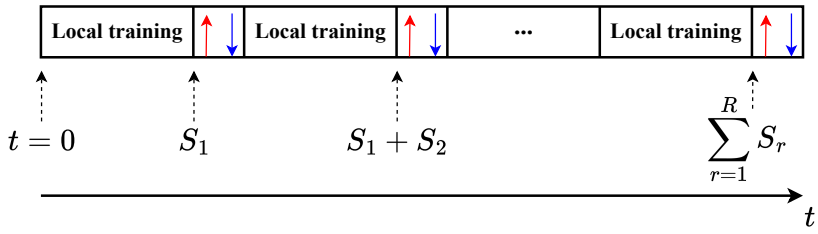


FIGURE 3.2: Illustration of the timeline of communication rounds in federated leaning. Up and down arrows signify the uplink and downlink communications, respectively.

3.3.3 Aggregation

After receiving uploaded parameters from all users, the BS first combines, demodulates and dequantizes the received weights ("post-processing" in Fig. 3.1), and the new updated global weights are obtained via aggregating these noisy weights. This aggregation step can be represented using the mapping $\phi_{av}(\cdot)$. At the end of aggregation, the BS has $\phi_{av}(\phi_{n,UL}(\mathbf{w}_t^n, \forall n))$. At this point, the global testing accuracy is defined as the accuracy of the aggregated model tested on the testing set at server, which is different from the testing sets of users. Note that for the simplicity of aggregation, it is also assumed that the model weights of each user ($\mathbf{w}_t^n \in \mathbb{R}^{W \times 1}$) are real-valued and of same size $W \times 1$.

3.3.4 Downloading updated weights

In this step, BS broadcasts the aggregated model to all users via DL channel (blue dashed line in Fig. 3.1). The received model at the n^{th} user is denoted by the mapping $\phi_{n,DL}(\phi_{av}(\phi_{n,UL}(\mathbf{w}_t^n, \forall n)))$, where $\phi_{n,DL}(\cdot)$ represents the downlink communication process for user n. This process includes quantization, modulation and precoding ("pre-processing" in Fig. 3.1) at BS, and the effects of transmission through the n^{th} multi-path downlink channel.

In this way, each round progresses until the global (or local) testing accuracy reaches convergence. An illustration of the FL timeline can be found in Fig. 3.2. In this figure, t denotes the time index for local step. $S_1, S_1 + S_2, \ldots, \sum_{r=1}^{R} S_r \in \mathcal{T}$ are time instances when UL communication (red up arrow), weights aggregation and DL communication (blue down arrow) take place. Detailed analysis about these three mappings will be explained in the following section.

3.4 Uplink and Downlink Communications

Since BS is equipped with a large number of antennas, we consider ZF approach at the BS, and analyze the impacts on model weights using channel hardening effect. Performance of ZF techniques in massive MIMO system has been investigated in [65] and [66]. First, the update step of FL algorithm in a round can be expressed as

$$\mathbf{v}_{t+1}^n = \mathbf{w}_t^n - \eta_t \nabla F_n(\mathbf{w}_t^n, \xi_t^n), \tag{3.5}$$

$$\mathbf{w}_{t+1}^n = \begin{cases} \mathbf{v}_{t+1}^n, & \text{if } t+1 \notin \mathcal{T}, \\ \phi_n\left(\mathbf{v}_{t+1}^n, \forall n\right), & \text{if } t+1 \in \mathcal{T}, \end{cases} \tag{3.6}$$

where the variable $\mathbf{v}_t$ is introduced to denote the immediate result of one step SGD. We define $\mathbf{g}_t = \sum_{n=1}^N \alpha_n \nabla F_n(\mathbf{w}_t^n, \xi_t^n)$ and $\bar{\mathbf{g}}_t = \sum_{n=1}^N \alpha_n \nabla F_n(\mathbf{w}_t^n)$. Therefore, $\mathbb{E}\left\{\mathbf{g}_t - \bar{\mathbf{g}}_t\right\} = \mathbf{0}$. The function $\phi_n(\cdot)$ represents the distortion arising due to uplink and downlink communications, which can be decomposed into uplink operation $\phi_{n,UL}(\cdot)$, averaging $\phi_{av}(\cdot)$, and downlink broadcast operation $\phi_{n,DL}(\cdot)$, i.e., a composition of three mappings as $\phi_n\left(\mathbf{w}_t^n, \forall n\right) = \phi_{n,DL} \circ \phi_{av} \circ \phi_{n,UL}\left(\mathbf{w}_t^n, \forall n\right)$.

3.4.1 Uplink mapping

In the uplink process, W real-valued weights are resized into complex-valued data streams for transmissions, say $\mathbf{W}_t^n \in \mathbb{C}^{N_D \times \tau_{UL}}$, where $\tau_{UL} = \frac{W}{2N_D}$. The received signal at the massive MIMO BS can be written as

$$\mathbf{Y}_t^{(UL)} = \sum_{n=1}^N \mathbf{H}_t^n \mathbf{W}_t^n \sqrt{\frac{P_{UL}}{N_D \sigma_w^2}} + \mathbf{Z}_t^{(UL)}, \tag{3.7}$$

where entries of $\mathbf{W}_t^n$ are assumed to have zero mean and σ_w^2 variance; $\mathbf{Z}_t$ denotes zero-mean Gaussian noise with σ^2 variance, and P_{UL} is the transmit power constraint for each device, i.e., $tr\mathbb{E}\left\{\mathbf{W}_t^n \mathbf{W}_t^{nH}\right\} = \tau_{UL} N_D \sigma_w^2$. With the zero forcing receiver with $N_{BS} \geq N \cdot N_D$ number of antennas, the estimated weights at the BS can be given as

$$\hat{\mathbf{W}}_t = \sqrt{\frac{N_D \sigma_w^2}{P_{UL}}} \left(\mathbf{H}_t^H \mathbf{H}_t\right)^{-1} \mathbf{H}_t^H \mathbf{Y}_t^{(UL)} = \mathbf{W}_t + \sqrt{\frac{N_D \sigma_w^2}{P_{UL}}} \left(\mathbf{H}_t^H \mathbf{H}_t\right)^{-1} \mathbf{H}_t^H \mathbf{Z}_t^{(UL)},$$

where $\mathbf{H}_t = \left[\mathbf{H}_t^1, \ldots, \mathbf{H}_t^N\right]$ and $\mathbf{W}_t^T = \left[\mathbf{W}_t^{1T}, \ldots, \mathbf{W}_t^{NT}\right]$. Each column of the estimation error term has zero mean and covariance matrix as $\frac{N_D \sigma_w^2 \sigma^2}{P_{UL}} \left(\mathbf{H}_t^H \mathbf{H}_t\right)^{-1}$. Owing to channel hardening $\frac{1}{N_{BS}} \mathbf{H}_t^H \mathbf{H}_t \to \mathbf{I}$, we simplify the covariance matrix of the effective noise above as $\frac{N_D \sigma_w^2 \sigma^2}{P_{UL}} N_{BS}^{-1} \mathbf{I}_{N_D}$. The variable $\hat{\mathbf{W}}_t$ can be used to express the estimated model weight for each user at the BS as $\hat{\mathbf{w}}_t^n = \mathbf{w}_t^n + \mathbf{z}_{t,UL}^n$, where $\mathbf{z}_{t,UL}^n \sim \mathcal{N}\left(\mathbf{0}, \frac{N_D \sigma_w^2 \sigma^2}{2 P_{UL} N_{BS}} \mathbf{I}\right)$ and the factor of 2 comes from converting the complex random variables to the real ones.

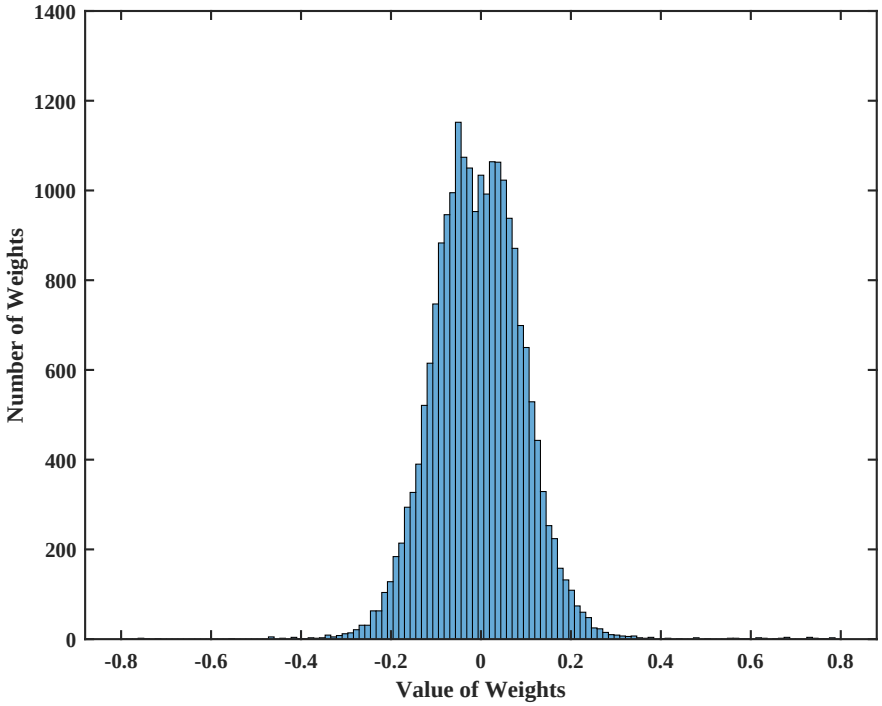


FIGURE 3.3: Histogram of the model weights.

Remark: In the above analysis, we assume that the entries of $\mathbf{W}_t^n$ follow a Gaussian distribution. We make the justification as follows. During the training, weights are updated as combinations of many independent updates (gradients from mini-batches). According to the central limit theorem (CLT) [67], when a large number of independent variables are combined linearly, their sum tends to follow a Gaussian distribution. This argument is also explained in [68], especially for large models that have long training periods and employ regularization techniques. In Fig. 3.3, we also give a visualization for the histogram of the model weights, which supports the suitability of the Gaussian assumption.

3.4.2 Aggregation mapping

Having $\hat{\mathbf{W}}_t$ at the BS, the averaging operation can be given as

$$\phi_{av}\left(\hat{\mathbf{W}}_t\right) = \sum_{n=1}^{N} \alpha_n \hat{\mathbf{w}}_t^n = \bar{\mathbf{w}}_t + \bar{\mathbf{z}}_{t,UL}, \tag{3.8}$$

where $\bar{\mathbf{w}}_t = \sum_{n=1}^{N} \alpha_n \mathbf{w}_t^n$, $\bar{\mathbf{z}}_{t,UL} \sim \mathcal{N}\left(\mathbf{0}, \frac{\|\boldsymbol{\alpha}\|_2^2 \sigma_w^2}{2} \frac{\sigma^2 N_D}{P_{UL} N_{BS}} \mathbf{I}\right)$ and $\|\boldsymbol{\alpha}\|_2^2 = \sum_{n=1}^{N} \alpha_n^2$.

3.4.3 Downlink mapping

The aggregated vector, say $\hat{\bar{\mathbf{w}}}_t = \phi_{av}\left(\hat{\mathbf{W}}_t\right)$, is then broadcasted to each user device via the base station through the DL channel. Since each device can receive only N_D data streams, the vector $\hat{\bar{\mathbf{w}}}_t$ is resized into a complex-valued matrix $\hat{\bar{\mathbf{W}}}_t \in \mathbb{C}^{N_D \times \tau}$ with $\tau = \frac{W}{2N_D}$. The received signal at the n^{th} device can be given as

$$\mathbf{Y}_{n,t}^{(DL)} = \mathbf{H}_t^{nH}\mathbf{F}_t\hat{\bar{\mathbf{W}}}_t p_{DL} + \mathbf{Z}_{n,t}^{(DL)},$$

where channel reciprocity is assumed; and $\mathbf{F}_t$ denotes the $N_{BS} \times N_D$ precoder such that $\mathbf{F}_t^H\mathbf{F}_t = \mathbf{I}_{N_D}$. The transmit power constraint (P_{DL}) is given as

$$p_{DL}^2 = \frac{P_{DL}\tau}{tr\mathbb{E}\left\{\mathbf{F}_t\hat{\bar{\mathbf{W}}}_t\hat{\bar{\mathbf{W}}}_t^H\mathbf{F}_t^H\right\}} = \frac{P_{DL}}{\hat{\sigma}_w^2 N_D},$$

where $\hat{\sigma}_w^2 = \|\boldsymbol{\alpha}\|_2^2\sigma_w^2\left(1 + \frac{\sigma^2 N_D}{P_{UL}N_{BS}}\right)$. Then, the estimates at the n^{th} user can be obtained as $\left(\mathbf{H}_t^{nH}\mathbf{F}_t\right)^{-1}\mathbf{Y}_{n,t}^{(DL)}p_{DL}^{-1}$. The corresponding covariance matrix for each column of the estimation error $\left(\mathbf{H}_t^{nH}\mathbf{F}_t\right)^{-1}\mathbf{Z}_{n,t}^{(DL)}p_{DL}^{-1}$ can be computed as

$$\begin{aligned}
&\frac{\hat{\sigma}_w^2 N_D\sigma^2}{P_{DL}}\left(\mathbf{H}_t^{nH}\mathbf{F}_t\right)^{-1}\left(\mathbf{H}_t^{nH}\mathbf{F}_t\right)^{-H} \\
&= \frac{\hat{\sigma}_w^2 N_D\sigma^2}{P_{DL}}\left(\mathbf{F}_t^H\mathbf{H}_t^n\mathbf{H}_t^{nH}\mathbf{F}_t\right)^{-1} \\
&\approx \frac{\hat{\sigma}_w^2 N_D\sigma^2}{P_{DL}}N_D^{-1}\mathbf{I} = \frac{\hat{\sigma}_w^2\sigma^2}{P_{DL}}\mathbf{I},
\end{aligned}$$

where $\mathbf{H}_t^n\mathbf{H}_t^{nH}$is approximated to $N_D\mathbf{I}_{N_{BS}}$, owing to channel hardening effect. Thus, we can write $\phi_{n,DL}\left(\hat{\bar{\mathbf{w}}}_t\right) = \hat{\bar{\mathbf{w}}}_t + \bar{\mathbf{z}}_{t,DL}^n$, where $\bar{\mathbf{z}}_{t,DL}^n \sim \mathcal{N}\left(\mathbf{0}, \frac{\hat{\sigma}_w^2\sigma^2}{2P_{DL}}\mathbf{I}\right)$ is the Gaussian noise at the typical user.

Overall, one can express the composite uplink-downlink aggregation process as

$$\begin{aligned}
\phi_n\left(\mathbf{v}_t^n, \forall n\right) &= \phi_{n,DL} \circ \phi_{av} \circ \phi_{n,UL}\left(\mathbf{v}_t^n, \forall n\right) && \text{(3.9a)} \\
&= \phi_{n,DL} \circ \phi_{av}\left(\hat{\mathbf{V}}_t\right) && \text{(3.9b)} \\
&= \phi_{n,DL}\left(\bar{\mathbf{v}}_t + \bar{\mathbf{z}}_{t,UL}\right) && \text{(3.9c)} \\
&= \bar{\mathbf{v}}_t + \underbrace{\bar{\mathbf{z}}_{t,UL} + \bar{\mathbf{z}}_{t,DL}^n}_{\bar{\mathbf{z}}_t^n}, && \text{(3.9d)}
\end{aligned}$$

where $\bar{\mathbf{v}}_t = \sum_{n=1}^N \alpha_n\mathbf{v}_t^n$, and the variance for the effective noise $\bar{\mathbf{z}}_t^n$, σ_z^2, can be seen in the proof of Lemma 3.1.

Remark: In the above mapping, the weights can be considered with quantization and modulation by introducing quantization error, e.g., with uniform quantization for real

and imaginary parts in Eqn. (3.7), the resulting error variance can be given as sum of noise variance σ^2 and the variance of the quantization error $\frac{\Delta^2}{6}$, where Δ is the quantization step size. The total effective variance can be obtained similarly with additive quantization error variance, which affects the convergence in the same way as the noise variance σ_z^2. It should be noted that, in the above analysis, we simplify certain factors, such as quantization and modulation, to focus on key aspects of system behavior and derive analytical insights. However, in the simulation, we employ the practical QPSK modulation to assess the system's performance in realistic wireless communication scenarios. The use of QPSK in simulations allows us to capture the impact of modulation on performance, which might not be fully reflected in the theoretical analysis. Despite this difference, the theoretical results still provide valuable insights that complement the simulation findings.

Lemma 3.1. *For UL-DL imperfect communication, given any model* $\mathbf{w}^*$*, the squared* l_2*-norm of the model error is bounded as*

$$\begin{aligned} &\mathbb{E}\left\|\bar{\mathbf{w}}_{t+1}-\mathbf{w}^*\right\|_2^2 \leq \mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}-\mathbf{w}^*\right\|_2^2 + W\sigma_z^2\delta_{t+1\in\mathcal{T}}, \\ &\sigma_z^2 = \frac{\|\boldsymbol{\alpha}\|_2^2\sigma_w^2}{2}\frac{\sigma^2}{P_{DL}}\left[\left(\frac{P_{DL}}{\sigma^2}+\|\boldsymbol{\alpha}\|_2^2\right)\frac{\sigma^2 N_D}{P_{UL}N_{BS}}+\|\boldsymbol{\alpha}\|_2^2\right], \end{aligned} \tag{3.10}$$

where $\delta_{t\in\mathcal{T}}$ *is an indicator function, taking value 1, when* $t\in\mathcal{T}$*, and* 0 *otherwise.*

Proof is given in Appendix 3.7. The above result shows that if the UL-DL links are perfect, i.e., $\sigma_z^2=0$, the models can be aggregated perfectly, and leads to similar errors. In other words, when $\sigma_z^2=0$, $\bar{\mathbf{v}}_{t+1}=\bar{\mathbf{w}}_{t+1}$. Regarding the effective noise variance σ_z^2, it can be seen that the downlink reliability is more important than the uplink's, since in the above expression of σ_z^2, the downlink SNR $\frac{P_{DL}}{\sigma^2}$ is the main factor in division.

3.4.4 Convergence analysis

Following assumptions for the loss function $F_n(\cdot), n=1,\ldots,N$ are made for the convergence analysis [31, 40, 69, 70].

Assumption 1. (Smoothness)*:* $F_1,\ldots,F_N$ *are all* L*-smooth, for* $\forall\mathbf{v},\mathbf{w}$*,* $F_n(\mathbf{v})\leq F_n(\mathbf{w})+(\mathbf{v}-\mathbf{w})^T\nabla F_n(\mathbf{w})+\frac{L}{2}\|\mathbf{v}-\mathbf{w}\|_2^2$.

Assumption 2. (Convexity)*:* $F_1,\ldots,F_N$ *are all* μ*-strongly convex, for* $\forall\mathbf{v},\mathbf{w}$*,* $F_n(\mathbf{v})\geq F_n(\mathbf{w})+(\mathbf{v}-\mathbf{w})^T\nabla F_n(\mathbf{w})+\frac{\mu}{2}\|\mathbf{v}-\mathbf{w}\|_2^2$.

Assumption 3. (Bounded variance for SGD)*: The variance of SGD satisfies*

$$\mathbb{E}\left\|\nabla F_n\left(\mathbf{w}_t^n,\xi_t^n\right)-\nabla F_n\left(\mathbf{w}_t^n\right)\right\|_2^2\leq\sigma_n^2, \tag{3.11}$$

which leads to $\mathbb{E}\left\|\sum_{n=1}^N\alpha_n\nabla F_n(\mathbf{w}_t^n,\xi_t^n)-\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2\leq\sum_n\alpha_n^2\sigma_n^2$.

Assumption 4. (Uniformly bounded gradient)*: the squared l_2-norm of gradients satisfies*

$$\mathbb{E}\left\|\nabla F_n\left(\mathbf{w}_t^n, \xi_t^n\right)\right\|_2^2 \leq G^2, \forall n, \tag{3.12}$$

which provides $\mathbb{E}\left\|\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2 \leq \|\boldsymbol{\alpha}\|_2^2 G^2$.

Assumptions 1 and 2 are the standard ones for l_2-norm regularized linear regression, logistic regression, and softmax classifier. Assumptions 3 and 4 are based on bounded gradients and are helpful in bounding the model-error variance while analyzing the convergence. The following result utilizes μ-strong convexity assumption (Assumption 2) to bound the next step error.

Remark: In this thesis, cross-entropy (CE) is selected as the loss function. We show that the cross-entropy function satisfies the Assumption 1 and 2 via the following derivation.

Assumption 1:

First, note that CE: $\mathbb{R}^n \to \mathbb{R}$ is differentiable, therefore, to prove that CE is L-smooth, it is sufficient to find L such that $\|\nabla \mathrm{CE}(x)\|_2 \leq L$. Since in deep learning, the multi-class CE function is used jointly with the softmax function (as discussed and explained in Chapter 2), we have $\mathrm{CE}(x) = -\log \mathrm{softmax}(x)_i = -\log\left(\frac{e^{x_i}}{\sum_j e^{x_j}}\right)$, where the only non-zero element of the 1-hot ground truth vector is at i^{th} index. We also let $s = \sum_k e^{x_k}$. Then, applying the calculus rules gives us

$$\nabla \mathrm{CE}(x)_i = -\frac{\sum_{l\neq i} e^{x_l}}{s} \quad \text{(true class)} \tag{3.13}$$

$$\nabla \mathrm{CE}(x)_j = \frac{e^{x_j}}{s}, \forall j \neq i, \quad \text{(other classes)} \tag{3.14}$$

Therefore,

$$\begin{aligned}\|\nabla \mathrm{CE}(x)\|_2^2 = \frac{(\sum_{l\neq i} e^{x_l})^2 + \sum_{l\neq i} e^{2x_l}}{s^2} \overset{(a)}{\leq} 2\frac{(\sum_{l\neq i} e^{x_l})^2}{s^2} \\ \overset{(b)}{\leq} \frac{(\sum_{l\neq i} e^{x_l})^2}{(\sum_{l\neq i} e^{x_l})^2} = 2,\end{aligned} \tag{3.15}$$

where in (a), the fact that $\sum_i v_i^2 \leq (\sum_i v_i)^2$ for $v_i > 0$ is used. In (b), for the denominator s, we remove the i^{th} term e^{x_i}, that is $(\sum_{l\neq i} e^{x_l})^2 < s^2$, therefore further decreasing the value of equation. And this concludes our proof for the L-Lipschitz continuity of CE function used in this thesis.

Assumption 2:

For the cross-entropy function, we start by considering an arbitrary input x, then the loss function has the form $-\sum_{j=1}^{C} y_j \log(\hat{y}_j)$. C is the number of classes, y_j takes on value 1 when j is the index of the correct class and 0 otherwise. In this way, considering all the classes, the loss function is just $F(x) = -\log(x)$, which is obviously convex since the second derivative is $\frac{1}{x^2} > 0$.

For the general case where multiple input samples are considered, the loss function has the form $-\frac{1}{D}\sum_{i=1}^{D}\sum_{j=1}^{C} y_{ij} \log \hat{y}_{ij}$, which is the summation of loss function for all input samples. Since the summation of the convex function is still a convex function, this concludes our proof for the convexity of the CE function used in this thesis.

Assumption 3 and 4:

For detailed proof of the suitability of these two assumptions. The readers are referred to [71]. As a brief sketch of the proof in [71], a distribution $\{0, 1\}$ with $P(X = 0) = P(X = 1) = \frac{1}{2}$ is considered. By utilizing the asymptotic expansion, it is shown that the results in Assumption 3 and 4 are suitable for practical machine learning problems when the loss functions satisfy Assumption 1 and 2, which have been proved above. We would also like to note that, these assumptions are used widely in the existing work such as [31, 62, 69, 70, 72–74], where the training tasks and loss functions used align with the ones used in this thesis.

Lemma 3.2. *Let Assumptions 1-4 hold. Then, with respect to the optimized model weights* $\mathbf{w}^*$*, the squared* l_2*-norm of the next step model update error can be bounded as*

$$\mathbb{E}\left\|\bar{\mathbf{v}}_{t+1} - \mathbf{w}^*\right\|_2^2 \leq (1 - \eta_t \mu)\, \mathbb{E}\left\|\bar{\mathbf{w}}_t - \mathbf{w}^*\right\|_2^2 + \eta_t^2 \left(\sum_n \alpha_n^2 \left(\sigma_n^2 + G^2\right)\right). \tag{3.16}$$

Proof is provided in Appendix 3.8. The above result shows that if η_t and μ are chosen properly, one can show the model updates lead to non-increasing errors. This important result is proved in the following theorem, which employs the smoothness of F_k (Assumption 1).

Theorem 3.3. *Let Assumptions 1-4 hold and* σ_z^2 *be defined therein. Then, the proposed FL algorithm satisfies*

$$\mathbb{E}\left\{F\left(\bar{\mathbf{w}}_t\right)\right\} - F^* \leq \frac{L}{2} \cdot \frac{\nu}{t + \beta_1}, \tag{3.17}$$

where $\eta_t = \frac{\beta_0}{t+\beta_1}$, $\nu \geq \frac{\beta_0^2 C}{\beta_0 \mu - 1}$, $\beta_1 > 0$, *and* $C = \sum_n \alpha_n^2 \left(\sigma_n^2 + G^2\right) + W \frac{\sigma_z^2}{\eta_t^2} \delta_{t+1 \in \mathcal{T}}$, $\beta_0 > \frac{1}{\mu}$.

Proof is provided in Appendix 3.9. Theorem 3.3 tells that the modeling error for the UL-DL FL algorithm reduces in $\mathcal{O}\left(t_\tau^{-1}\sigma_z^2\right)$ manner, where for $t_0 < t, t_\tau = t - t_0$ denotes the system elapsed time. And this indicates that the algorithm converges in time, and the accuracy is inversely proportional to communication error variance σ_z^2, whose expression accounts for and explains the effects of the communication constraints on the

convergence. We can reduce the communication cost via two factors, viz., varying the transmit power (equivalently varying SNRs) and reducing the number of communicating rounds (equivalently varying the number of local steps). Note that the quantization error also affects the variance σ_z^2, proving the number of quantization bits to be the third factor. The convergence bound linearly depends on the L-smoothness, whereas it is inversely proportional to the μ-convexity; that is, the stronger the convexity is, the lower the convergence error is. For a FL training process, we can define the overall learning cost as:

$$\begin{aligned}&\text{Overall communication cost}\\&= \#\text{ of communication rounds} \times \#\text{ of model parameters}\\&\times \#\text{ of quantization bits} \times \text{normalized transmit power.}\end{aligned} \tag{3.18}$$

In our simulation, we make attempt to vary these factors as follows.

Communication rounds: From the convergence analysis, we see that weights converge as iterations progress, which means there is less need to communicate weights at later iterations. Keeping this fact in mind, we set two modifications for improving the communication cost. The first one includes the decreasing number of local steps, i.e., $S_1 \geq S_2 \geq \cdots \geq S_R$, where the equality $S_r = S_{r+1}$ implies that for much larger iterations near convergence, reducing local iterations cannot further improve the performance, e.g., including the case $S_r = 1$ for r large. The second modification is to select users by comparing the l_2-norm of difference in weights, that is, at $t = \sum_{\hat{r}=1}^{r} S_{\hat{r}}$, sort the values of $\left\|\mathbf{w}_t^n - \mathbf{w}_{t-S_r}^n\right\|_2, \forall n$ and choose $N_0 < N$ users with higher values. It provides more opportunities to users who have less convergence with respect to others. These modifications not only improve the communication cost of transmission, but also provide other users to have more diversity in physical layer communications, i.e., more reliable channels.

Remark: Since in practice, distances between users and BS are different, thus, it could be more reasonable to design a update policy $(S_{n1}, \ldots, S_{nR})$ based on a user's local situation rather than the global updating policy $(S_1, \ldots, S_R)$, e.g., within a global communication round, users that are closer to the BS may be able to perform more local training steps compared with those who are far from the BS.

Transmit power: Since σ_z^2 depends on uplink and downlink SNRs, this gap reduces to a constant if $\sigma_z^2 \propto R$. It also shows that the SNR in UL-DL communications can be increased as the iterations progress, that is, the effective noise variance σ_z^2 should be proportional to the step size η_t^2, that is, $\sigma_z^2 \propto \eta_t^2 \propto \frac{1}{(t+\beta_1)^2}$. At the same time, In the learning process, at lower iterations, when weights are not converged, we can decrease the transmit power, since they are being averaged for more updates. Similarly, for higher iterations, when weights are close to convergence, one can increase the transmit power

Algorithm 1 Communication-efficient FL algorithm

Initialization:
  Initialize $T, N, N_0, P_{UL,0}, P_{DL,0}$,
for $r = 1, \ldots, R$ **do**
  if $r = 1$ **then**
    All users initialize its local weights $\mathbf{w}_0^n, \forall n$
  else
    Downlink transmission:
      1. BS quantizes, modulates and precodes $\hat{\tilde{\mathbf{w}}}_t$
      2. Value of $P_{DL,r}$ is updated as:
        $P_{DL,r} = P_{DL,0}(1 - e^{-\gamma r})$
      3. BS broadcasts $\hat{\tilde{\mathbf{w}}}_t$ with power $P_{DL,r}$.
  end if
  for $n = 1, \ldots, N$ **do**
    if $r \neq 1$ **then**
      $\mathbf{w}_t^n = \phi_{n,DL}(\hat{\tilde{\mathbf{w}}}_t)$
    end if
    for $t = \sum_{\hat{r}=1}^{r} S_{\hat{r}} - S_r, \ldots, \sum_{\hat{r}=1}^{r} S_{\hat{r}} - 1$ **do**
      Update local weights as:
        $\mathbf{w}_{t+1}^n = \mathbf{w}_t^n - \eta_t \nabla F_n(\mathbf{w}_t^n, \xi_t^n)$
    end for
  end for
  Sort the values of $\left\|\mathbf{w}_{t+1}^n - \mathbf{w}_{t+1-S_r}^n\right\|_2, \forall n$.
  Select $N_0 < N$ users with higher values.
  Uplink transmission:
    1. Selected users quantize and modulate weights.
    2. Value of $P_{UL,r}$ is updated as:
      $P_{UL,r} = P_{UL,0}(1 - e^{-\gamma r})$
    3. Users upload weights with power $P_{UL,r}$.
    4. BS combines, demodulates and dequantizes the received weights.
    5. BS performs aggregation to produce $\hat{\tilde{\mathbf{w}}}_{t+1}$.
end for

for accurate communications and averaging. Towards that, let $P_{DL,r} = P_{DL,0}(1 - e^{-\gamma r})$ and $P_{UL,r} = P_{UL,0}(1 - e^{-\gamma r})$.

Therefore, for R rounds, the incurred energy consumption is given as

$$\mathcal{E}_{1R} = N \sum_{r=1}^{R} P_{UL,0}(1 - e^{-\gamma r}) + \sum_{r=1}^{R} P_{DL,0}(1 - e^{-\gamma r}) \tag{3.19}$$

$$= (N P_{UL,0} + P_{DL,0}) \left(R - e^{-\gamma} \frac{1 - e^{-\gamma R}}{1 - e^{-\gamma}} \right). \tag{3.20}$$

On the other hand, conventionally, when constant power P_0 is used for all communication rounds, the total incurred power consumption is $\mathcal{E}_{2R} = (N P_{UL,0} + P_{DL,0})\, R > \mathcal{E}_{1R}$. Therefore, the above analysis shows that by varying SNR, the energy consumption of the system can be improved without losing the performance at the convergence. The

normalized value of transmit power $\hat{P}$ can be defined as:

$$\hat{P} = \frac{\mathcal{E}_{1R}}{\mathcal{E}_{2R}} = \frac{\left(R - e^{-\gamma}\frac{1-e^{-\gamma R}}{1-e^{-\gamma}}\right)}{R}. \tag{3.21}$$

In our communication cost comparison (see in next section), this variable serves as a weighting factor that reflects the improvement of transmit power consumption introduced by varying P_r during training.

Quantization: We employ the uniform quantization scheme for the model weights prior to transmission. We vary the number of quantization bits to reduce the overhead in each round. In subsubsection 3.5.4.5, we also consider another quantization scheme based on stochastic rounding [40]. Specifically, the procedures of this quantization scheme include scale up, stochastic rounding, limit and scale down.

Overall, the aforementioned features and factors in our system can be summarized in the Algorithm 1. Similar to Fig. 3.1, Algorithm 1 explains the procedures of proposed FL framework, where lines 9-10 and 26-27 denote transmit power variation, followed by the transmission process (line 11 and line 28, for DL and UL, respectively). At local user side, lines 17-20 represent the local training procedures, where local steps variation is applied. Lines 22-23 stand for the l_2-norm scheduling policy before UL transmission.

Further, in Theorem 3.3, as $t \to \infty$, $\eta_t \to 0$, that is, for convergence of FL, the UL-DL communication links should be reliable, where the importance of DL links' reliability can be inferred from Lemma 3.1. Moreover, it can be noted that the above bound in Eqn. (3.17) is much tighter than the one, originally expressed in literature [31][1].

The following result expresses the convergence bound for the case when the learning rate is constant throughout the training process.

Corollary 3.4. *When learning rate is set to be a constant through whole training process, i.e., $\eta_t = \eta$, the average model accuracy is bounded by a constant gap with respect to the optimum model as*

$$\mathbb{E}\left\{F\left(\bar{\mathbf{w}}_T\right)\right\} - F^* \leq \frac{L}{2} \cdot \frac{\eta C}{\mu}, \tag{3.22}$$

where $C = \sum_n \alpha_n^2 \left(\sigma_n^2 + G^2\right) + W\frac{\sigma_z^2}{\eta^2}\delta_{t+1\in\mathcal{T}}$.

Proof is derived in Appendix 3.10. The above bound shows that for the constant learning rate η, the FL learning algorithm converges to a constant gap.

[1]For perfect UL-DL links ($\sigma_z^2 = 0$), the bound from [31] can be written as $\mathbb{E}\left\{F\left(\bar{\mathbf{w}}_R\right)\right\} - F^* \leq \frac{L}{2}\frac{\nu}{R+\beta_1}$, where $\eta_t = \frac{\beta_0}{t+\beta_1}$, $\nu = \frac{2}{\mu}\left(\frac{2}{\mu}C + \frac{\mu}{2}\gamma\mathbb{E}\left\|\mathbf{w}_1 - \mathbf{w}^*\right\|_2^2\right)$, $\beta_1 = \gamma$, $\beta_0 = \frac{2}{\mu}$, $\Gamma = F^* - \sum_{n=1}^{N}\alpha_n F_n^*$ and $C = \sum_n \alpha_n^2 d\sigma_n^2 + 6L\Gamma + 8(S-1)^2G^2$.

3.5 Simulation Results

3.5.1 Simulations setup

For an image classification task, we choose MNIST [75], EMNIST [76] and FMNIST [77] datasets for both i.i.d. and non-i.i.d. training data distributions. MNIST contains gray-scale images of size 28×28 for handwriting of numbers, ranging from 0 to 9. EMNIST is a more challenging dataset, from which we choose EMNIST-letters dataset containing gray-scale images of size 28×28 for handwriting of English letters, i.e., 26 classes. Similar to but more challenging than MNIST, FMNIST also has 10 classes, which correspond to 10 categories of clothes.

3.5.1.1 Training data preparation

For data augmentation purpose, we apply random rotation (ranging from -20 degree to 20 degree), random horizontal translation (ranging from -3 pixels to 3 pixels) and random vertical translation (ranging from -3 pixels to 3 pixels) for each training image of MNIST, EMNIST and FMNIST datasets.

3.5.1.2 Communication setup

As practiced and investigated in [45, 57] and [78], we consider $N = 25$ users, each with single $N_D = 1$ antenna. The learning weights of local and global models are stored in 32-bit float point format. A uniform quantization is employed to quantize the model weights, followed by Quadrature Phase Shift Keying (QPSK) for modulation. For i.i.d. Rayleigh fading model, channel state information (CSI) is assumed constant within each communication round and is known at the BS side. Note that for the purpose of simplicity, in this chapter we only consider perfect CSI. For the scenario with imperfect CSI, such as [79], similar convergence analysis and simulations can be conducted by including the channel estimation error in the noise variance.

3.5.1.3 Local model

For MNIST dataset, the local deep learning model comprises of a convolutional neural network (CNN) model with three convolutional layers, a fully connected layer, and a final classification layer with softmax function. Each convolutional layer is followed by a batch normalization layer, a ReLU activation layer, and a 2×2 max-pooling layer. Three convolutional layers contain 8, 16 and 32 filters, respectively. All filters are of size 3×3. When EMNIST and FMNIST datasets are considered, the local deep learning

model is same to that used in MNIST scenario except that no batch normalization layers are used [2].

3.5.1.4 MNIST

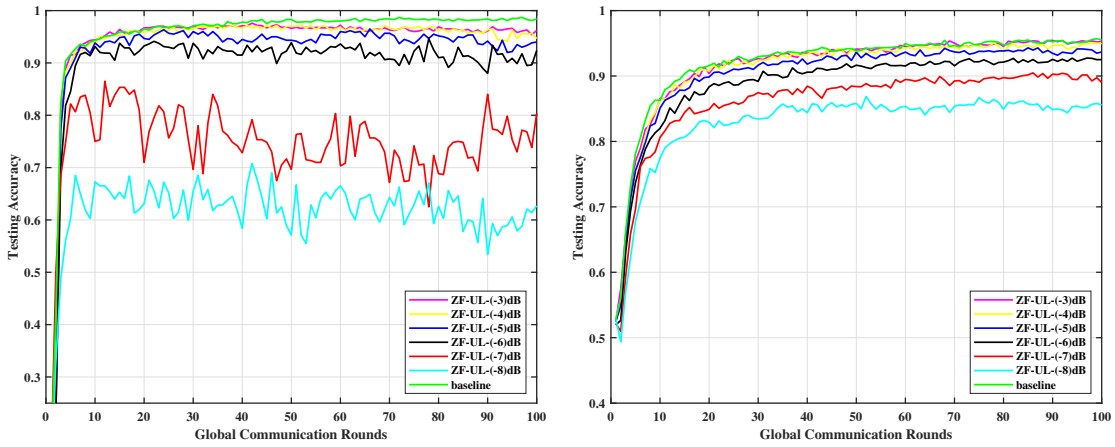
Each user has 50 training samples locally, and has a testing set of 200 samples. To demonstrate the effectiveness of the proposed system, we deliberately use a smaller number of training samples compared to testing samples to show that our system can perform well even in data-limited training scenarios. While increasing the number of training samples generally improves model robustness, the number of testing samples does not directly influence the model's inherent performance or its ability to generalize to unseen data. Instead, having more testing samples provides a more reliable estimate of the model's true performance. For i.i.d. case, overall training and testing sets are randomly shuffled and evenly splitted up among users in the cell, whereas in the non-i.i.d. case, for each local user, two labels are randomly selected for dominating labels. We define the fraction of the training set occupied by the samples with dominating labels as "non-i.i.d. factor". Note that the testing images in testing sets still have uniform distribution, that is, i.i.d. testing data samples. At each user, training parameters are set as follows. In the "fix local steps" scheme, learning rate is set to be 0.01, the number of local epochs is set to be 3 and for 50 training images, batch size is 10, leading to a total number of local updates to be $S = 3 \times 5 = 15$. In the scenario where number of local steps is varied, learning rate and batch size are set to be 0.01 and 10, respectively. S_r decreases according to $S_r = \frac{D_n}{|\xi_t^n|}\left(6 - \lfloor \frac{r-1}{15} \rfloor\right)$, where $\lfloor\rfloor$ denotes floor operation. In this chapter, the testing accuracy is defined as the ratio of correctly classified images to the total number of images in the testing set. Mathematically, testing accuracy can be expressed as:

$$\text{Accuracy} = \frac{\text{Number of correctly classified images}}{\text{Total number of images in the testing set}}$$

3.5.1.5 EMNIST and FMNIST

For these two datasets, each user has 100 training samples locally, and has a testing set of 200 samples. At each user, training parameters are set as follows. Local learning rate 0.01, the numbers of local training epochs and batch size are 7 and 10, respectively.

[2] When EMNIST dataset is used, dimension of fully connected layer and final classification layer will also be changed accordingly.



(A) (B)

FIGURE 3.4: Testing accuracy of the training progress at server side in (A), and local user side in (B), for different SNRs of UL links while DL-SNR is fixed to 10 dB; that is, $P_{DL,0}/\sigma^2 = 10$ dB.

3.5.1.6 Server/BS setup

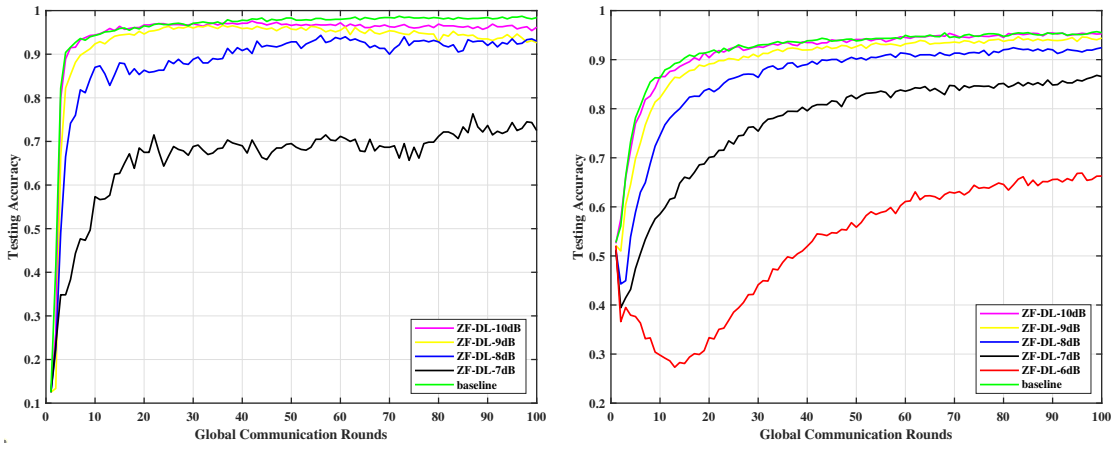
The server, or the massive MIMO base station, is considered to be equipped with 64 of antennas (except in subsections 3.5.3 and 3.5.4, where 128 antennas are considered at BS). Note that 64 and 128 are common choices for 5G and 6G massive MIMO communication scenario [80–82]. Unless otherwise noted, the number of global communication rounds is set to $R = 100$. The BS contains a global testing set, which contains 200 testing samples, and is used for getting global server side testing accuracy in each round.

In the following subsections, we will present our simulation results. We start with simulations on i.i.d. MNIST dataset (subsections 3.5.2 and 3.5.3), where we mainly focus on the impacts introduced by UL and DL transmission errors with different values of SNRs. After that, simulation results on non-i.i.d. MNIST under different system schemes proposed in this chapter are compared and analyzed (subsection 3.5.4), where transmit power, number of local steps, quantization bits and non-i.i.d. factor are varied. Simulations on EMNIST and FMNIST datasets are provided in subsection 3.5.5.

3.5.2 Testing accuracy for fixed learning rate

3.5.2.1 For different UL-SNRs and a fixed DL-SNR

Fig. 3.4 plots the progress of testing accuracy in different communication rounds at both server and user sides (in 3.4a and 3.4b respectively), when UL-SNR is varied and DL-SNR is kept fixed to 10 dB. At the BS side, massive MIMO gain (antenna gain) increases the signal power, that is, $10\log_{10} 64 \approx 18$. The values of the SNR for simulation are



(A) (B)

FIGURE 3.5: Testing accuracy of the training progress at server side in (A), and local user side in (B), for different SNRs of DL links while UL-SNR is fixed to -3 dB; that is, $P_{UL,0}/\sigma^2 = -3$ dB.

chosen to operate the receiver in the mid-SNR range. Simulations are averaged over 10 times.

From both Figs. 3.4a and 3.4b, it can be seen that increasing the SNR improves the performance of the trained models. At higher SNRs, testing accuracy is similar to that of baseline approach (perfect links). In Fig. 3.4, the gap at the convergence, is due to an additional term of σ_z^2 arising in Theorem 3.3 and Corollary 3.4. At the server side, there are more fluctuations in accuracy as compared to that at user side. It is due to the fact that the server receives the noisy version of locally updated weights owing to imperfect UL links. Also, the model at the server side is updated once every S steps. The almost steady behavior at local users is due to the reason that the accuracy is computed just before uplink transmission, that is, after the S steps local updates of received model weights. Note that green curves in Figs. 3.4a and 3.4b denote the baseline considered in [31], where both UL and DL channels are considered to be error-free. It can also be witnessed that, when UL SNR reaches -3 dB, the converged accuracy closely matches that of the baseline with error-free communication links. This suggests that at this SNR level, the communication is nearly fully accurate and reliable for the system under consideration, indicating that the system reaches saturation at this point. Similar trend can also be witnessed for the simulation in Chapter 4.

Remark: In the simulations, we assume that all users experience the same SNR value. This assumption simplifies the analysis, allowing us to focus on key system behaviors, such as model convergence and training dynamics. Additionally, the SNR value used in the simulation can be interpreted as an average SNR across users, as we plot the average testing accuracy across all users. This approach effectively captures the system's overall performance, as it is equivalent to the scenario where individual users have different SNR values but their collective impact is reflected in the averaged results. By

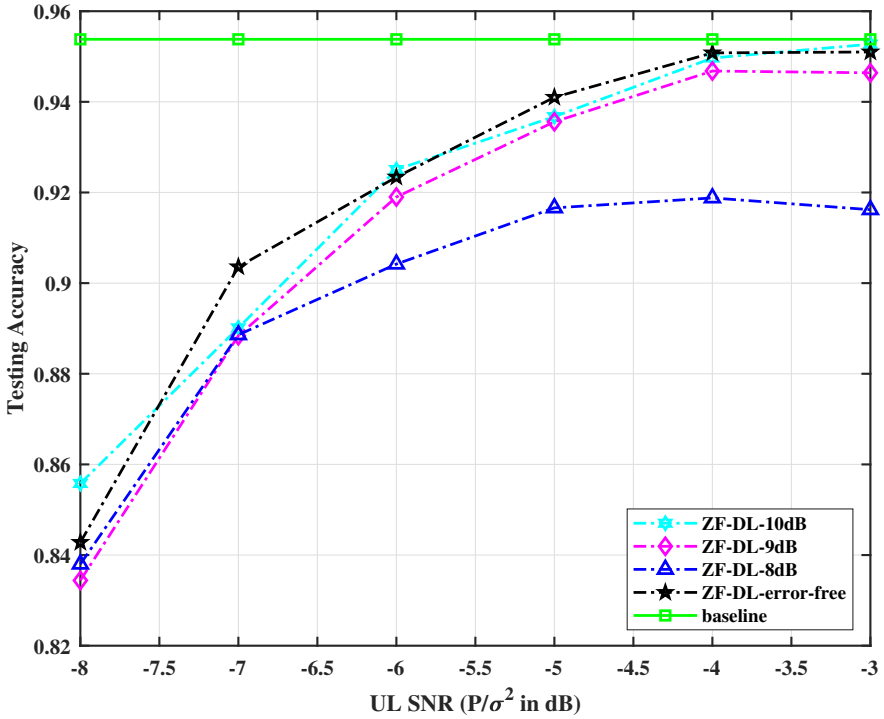


FIGURE 3.6: Converged local testing accuracy versus SNRs after $R = 100$ global communication rounds.

using this assumption, we ensure that the simulation captures the essential dynamics of FL performance while abstracting away user-specific SNR variability, which can be addressed in more detailed studies.

3.5.2.2 For different DL-SNRs and a fixed UL-SNR

Fig. 3.5 depicts the plots of testing accuracy progress when UL-SNR is fixed to -3 dB, and downlink SNR is varied between 7 dB to 10 dB. Similar to Fig. 3.4, the accuracy at the server side has more fluctuations compared with that at the users' side, and the accuracy is increased close to that of baseline approach for positive increments in DL-SNRs. The notable fact here is that the DL-SNR has a direct impact on the accuracies at both sides, as inferred in Lemma 3.1 and Theorem 3.3. This can be observed by comparing Fig. 3.5 with Fig. 3.4. At the server side, the fluctuations become less in Fig. 3.5a as SNR is increased, as compared to Fig. 3.4a. Similar observations can be seen for Fig. 3.5b and Fig. 3.4b. In this figure, the decline and rise of the accuracy at the first several rounds is due to random initialization of weights. When DL channel quality is relatively bad, e.g., DL-SNR equals 6 dB, such trend is more obvious and thus again, verifying our claim that DL channel quality is more important than that of UL. Similar to Fig. 3.4, green curves in Figs. 3.5a and 3.5b denote the baseline with perfect communication links introduced in [31].

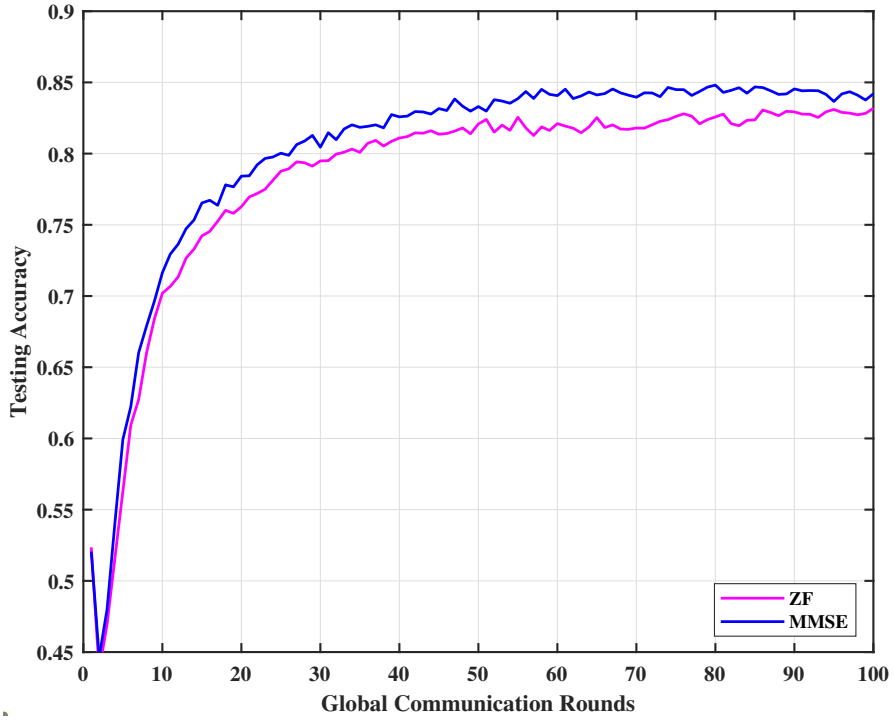


FIGURE 3.7: Testing accuracy at local user side on MNIST i.i.d. dataset when different precoding methods are implemented. $N_{BS} = 64$. Values of DL-SNR and UL-SNR are 8 dB and -8 dB, respectively; that is, $P_{DL,0}/\sigma^2 = 8$ dB, $P_{UL,0}/\sigma^2 = -8$ dB.

Fig. 3.6 shows the converged local testing accuracy versus UL-SNRs with different plots depicting different DL-SNRs. It can be seen that the testing accuracy increases with increase in any of UL or DL SNRs, and for higher SNR, it reaches close to the perfect case, i.e., the baseline approach. In this figure, apart from comparing our system's performance with baseline (green curve) introduced by [31], we also plot the simulation results based on the assumption that DL links are considered to be error-free (black curve) and this assumption has been made in many existing work [57–60].

3.5.2.3 For different precoding methods

In Fig. 3.7, we plot the average testing accuracy on i.i.d. MNIST dataset with ZF and MMSE precoding/combining methods. It can be witnessed that as compared with ZF, MMSE is able to achieve a slightly better performance and converges faster. In high SNR regime, this difference becomes negligible, therefore, for the rest of simulations, we adopt the ZF strategy. It can also be noted that the convergence bound depends on the statistical error variance which is approximately similar in high SNR regime for both the precoding strategies.

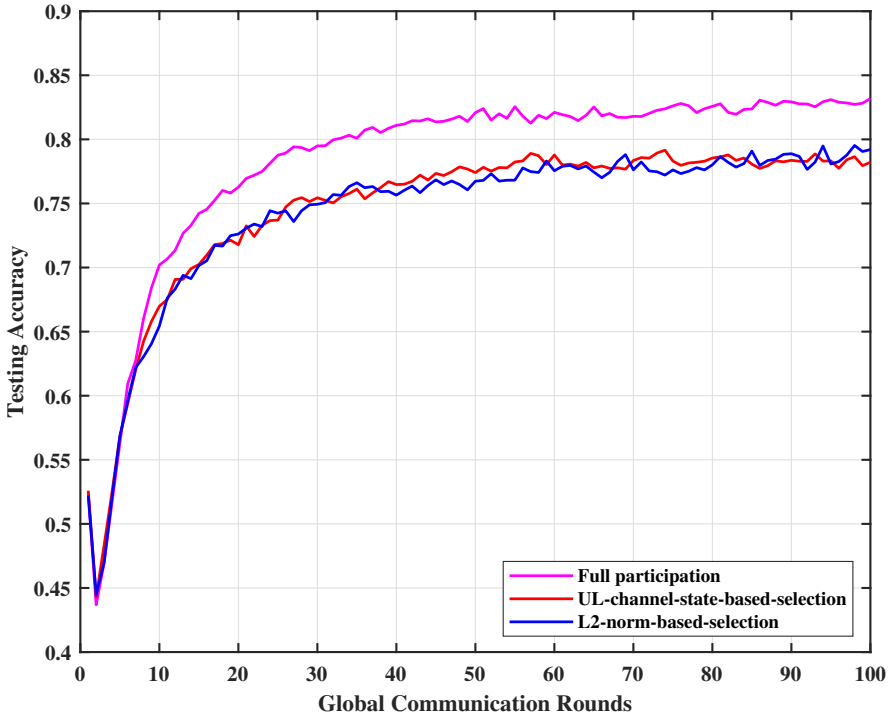


FIGURE 3.8: Testing accuracy at local user side on MNIST i.i.d. dataset when full participation, UL-channel-state-based scheduling policy and l_2-norm-based scheduling policy are considered. Values of DL-SNR and UL-SNR are 8 dB and -8 dB, respectively; that is, $P_{DL,0}/\sigma^2 = 8$ dB, $P_{UL,0}/\sigma^2 = -8$ dB.

3.5.2.4 Scheduling policy

In Fig. 3.8, the average testing accuracies at local user side are compared under full participation (magenta curve), UL-channel-state-based scheduling policy (red curve) and l_2-norm-based scheduling policy (blue curve). In both partial participation cases, within each global communication round, 15 users out of 25 are selected. In UL channel state scheduling policy, users with better UL communication links quality are selected, while in the l_2 scheduling policy, such selection is based on the criteria explained in Algorithm 1. Thus, UL channel state and l_2 scheduling policy can reduce the communication cost at the expense of testing performance reduction, that is, local models converge slower with less participations.

3.5.3 Testing accuracy for varying SNR and learning rate

In this subsection, we vary learning rate η_t, UL-SNR and DL-SNR through the training process. The value of η_t is decreased according to Theorem 3.3, i.e., $\eta_t = \frac{\beta_0}{t+\beta_1}$ with $\beta_0 = 2, \beta_1 = 100$, and both UL and DL SNRs are increased as $P_{UL,r} = P_{UL,0}(1 - e^{-\gamma r})$ and $P_{DL,r} = P_{DL,0}(1 - e^{-\gamma r})$ with $\gamma = 0.2$, $P_{UL,0}/\sigma^2 = -7$ dB, $P_{DL,0}/\sigma^2 = 7$ dB. Number of antennas at BS $N_{BS} = 128$.

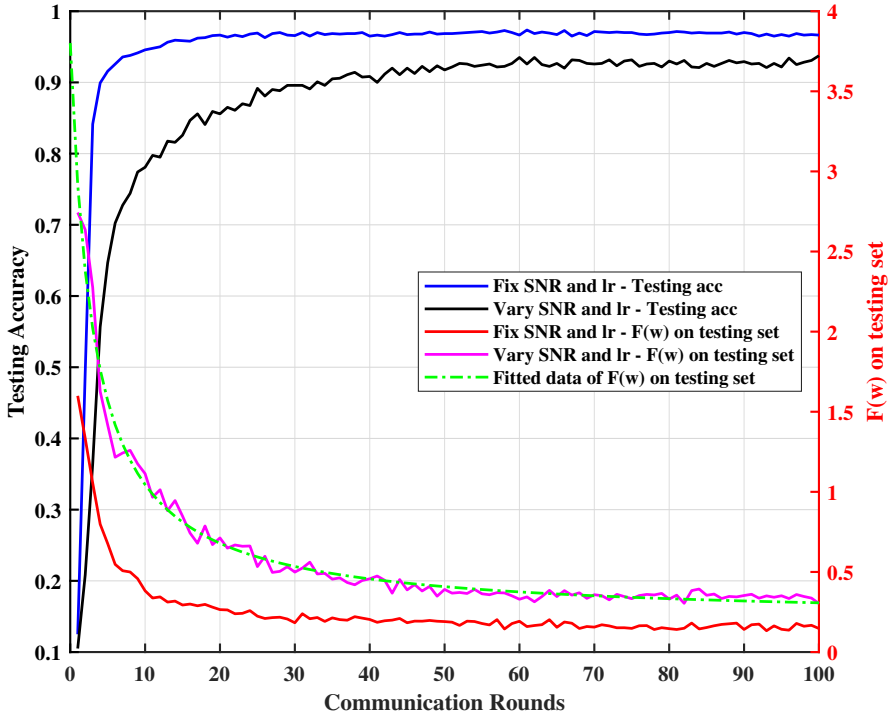


FIGURE 3.9: Progresses of testing accuracy (left-axis) and testing loss $F(\mathbf{w})$ (right axis) at the server side to compare the effects of varying and fixed SNRs through training process. The fitted line for varying SNR shows $\mathcal{O}\left(t_\tau^{-1}\sigma_z^2\right)$ behavior as proved in Theorem 3.3.

Fig. 3.9 plots the testing accuracy on i.i.d. MNIST (left-axis) and the values of the loss function $F(\mathbf{w})$ for the testing set (right-axis) versus the communication rounds. By comparing the performance for varying SNRs with the scenario where SNRs are fixed, it can be observed that in the former, models converge slower than that in the latter. However, varying SNR provides the advantage of energy efficiency improvement. The value of γ can be adjusted to trade-off energy efficiency with the convergence rate. To show the loss function decreases by $\mathcal{O}\left(t_\tau^{-1}\sigma_z^2\right)$, we fit the curve with the function $a+\frac{b}{t+c}$, which verifies the Theorem 3.3.

3.5.4 Simulations on non-i.i.d. MNIST

3.5.4.1 Effects of varying local steps and transmit power

In Fig. 3.10, we plot the averaged local testing accuracy versus the system running time as training progresses under different schemes with non-i.i.d. training data. Number of communication rounds is set to be $R = 60$ and non-i.i.d. factor is 70%. We average the accuracy of each scenario for 10 instances. It can be observed that under same system running time, varying local step scheme with fixed transmit power (green curve) gives best model performance at the end of training. While magenta and blue curves

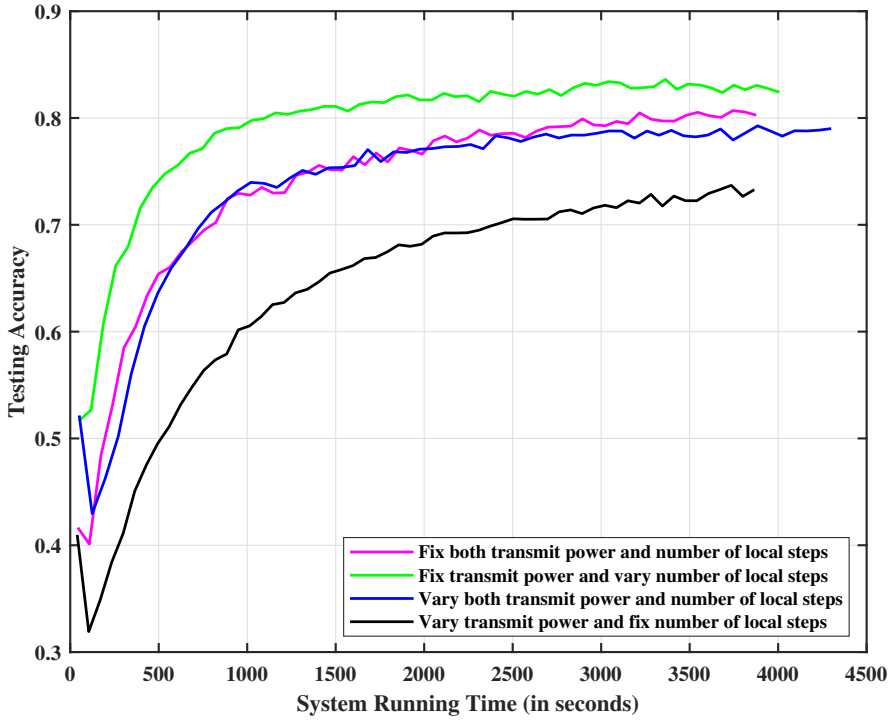


FIGURE 3.10: Testing accuracy on non-i.i.d. MNIST dataset versus system running time under different system schemes. Number of quantization bit is 16. $N_{BS} = 128$. Non-i.i.d. factor = 70%. Values of DL-SNR and UL-SNR are 7 dB and -7 dB, respectively; that is, $P_{DL,0}/\sigma^2 = 7$ dB, $P_{UL,0}/\sigma^2 = -7$ dB.

TABLE 3.2: Averaged testing accuracy of local user models at end of training when i.i.d. training dataset is used.

Schemes \ Quantization bits		16	14	12	10
Vary S_r	Vary P_r	95.52%	94.28%	95.44%	92.82%
	Fix P_r	95.32%	96.30%	95.90%	92.60%
Fix S_r	Vary P_r	94.14%	93.68%	93.62%	91.70%
	Fix P_r	94.86%	94.26%	94.66%	91.28%

show similar testing accuracy through training process, the difference is that the scheme corresponding to blue curve incurs less communication cost since both transmit power and number of local steps are varied in this scheme. Black curve represents the scheme where transmit power is varied and number of local steps is fixed, it can be witnessed that it returns worst result among all schemes under consideration, implying that the learning algorithm requires more iterations to converge, based on the transmit power variation model.

TABLE 3.3: Averaged testing accuracy of local user models at end of training when non-i.i.d. training dataset is used.

Schemes \ Quantization bits		16	14	12	10
Vary S_r	Vary P_r	79.01%	77.36%	76.82%	70.16%
	Fix P_r	82.42%	82.80%	81.32%	75.50%
Fix S_r	Vary P_r	73.29%	73.06%	71.56%	60.04%
	Fix P_r	80.27%	80.12%	79.00%	70.36%

3.5.4.2 Effects of varying quantization bits

In Table 3.2 and Table 3.3, we summarize the averaged testing accuracy of local user models at the end of training under different system schemes and quantization bits conditions on both i.i.d. and non-i.i.d. training datasets. For the same simulation setup in the previous subsubsection with $N_{BS} = 128$, ${}^{P_{DL,0}}/_{\sigma^2} = 7$ dB, ${}^{P_{UL,0}}/_{\sigma^2} = -7$ dB and 70% non-i.i.d. factor, we can observe that in both i.i.d. and non-i.i.d. cases, as number of quantization bits is increased, better testing performance can be seen at the expense of more communication cost. Since the difference introduced by varying number of local steps and transmit power can hardly be witnessed in the i.i.d. case, in the non-i.i.d. scenario, this difference stands out and becomes more obvious as shown in Fig. 3.10 and Table 3.3. Fixing transmit power through the whole training process gives better performance than "varying power" policy, this is due to the fact that on average, higher SNR values introduce fewer transmission errors.

Another interesting finding is that, in non-i.i.d. case, varying number of local steps not only accelerates the convergence but also leads to apparently better testing results at the end of training compared with its "fix local steps" counterpart. For instance, if we compare first and third rows of Table 3.3 (where transmit power is varied), we can observe 5% and 10% better testing performance in the former case when the constrains for quantization bits are 12 and 10, respectively. Similar trend can also be witnessed when the second and fourth rows are compared in Table 3.3, where the transmit power is fixed.

3.5.4.3 Comparison of communication cost

In Fig. 3.11, communication cost incurred by different system schemes in Table 3.3 is visualized and compared by utilizing a bar chart. Since all local users' models have same architecture, when plotting the bar chart, we do not consider "# of model parameters" as shown in Eqn. (3.18), leaving the remaining three factors under consideration:

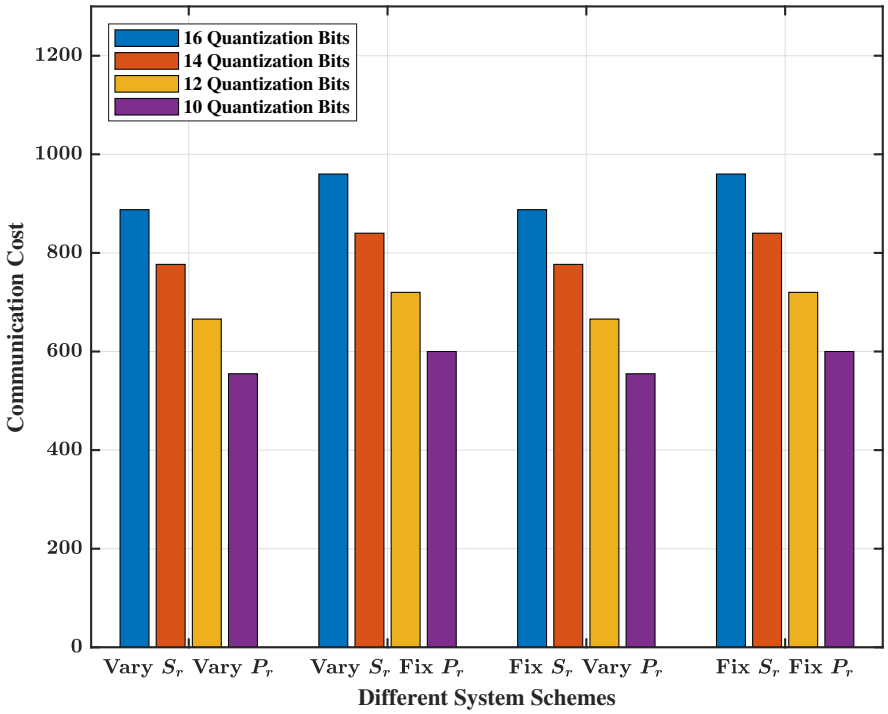


FIGURE 3.11: Visualization of communication cost under different systems schemes. Non-i.i.d. MNIST dataset is considered.

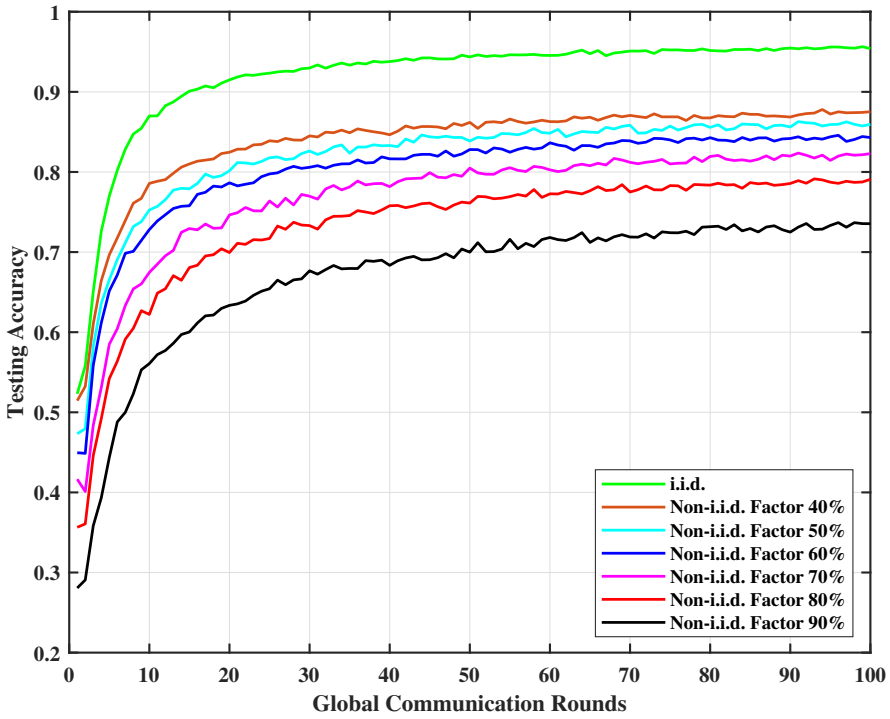


FIGURE 3.12: Testing accuracy on MNIST dataset with different non-i.i.d. factors. Number of quantization bit is 16. $N_{BS} = 128$. Values of DL-SNR and UL-SNR are 7 dB and -7 dB, respectively; that is, ${}^{P_{DL,0}}/_{\sigma^2} = 7$ dB, ${}^{P_{UL,0}}/_{\sigma^2} = -7$ dB.

1. Communication rounds: As stated in subsubsection 3.5.4.1, number of communication rounds R is set to be 60 for all system schemes.

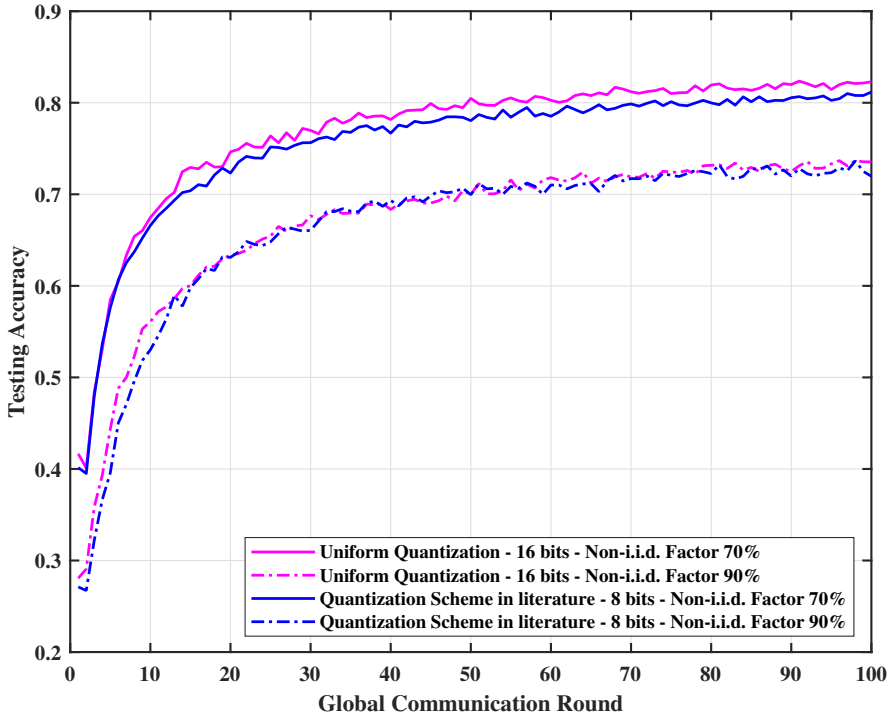


FIGURE 3.13: Testing accuracy on non-i.i.d. MNIST dataset when two quantization schemes are used. $N_{BS} = 128$. Values of DL-SNR and UL-SNR are 7 dB and -7 dB, respectively; that is, $P_{DL,0}/\sigma^2 = 7$ dB, $P_{UL,0}/\sigma^2 = -7$ dB.

2. Normalized transmit power: For $R = 60$ and $\gamma = \frac{1}{5}$, $\hat{P} = 0.9247$. When "fix P_r" strategy is used, normalized power is 1.

3. Quantization bits: We use the same values $\mathcal{B}$ as the ones we present in Table 3.3, that is, $16, 14, 12$ and 10 bits, respectively.

As shown in Fig. 3.11, for a given system scheme, decreasing number of quantization bits leads to less communication cost. While "vary S_r vary P_r" and "fix S_r vary P_r" yield same communication cost, the former scheme is able to provide better performance in terms of testing accuracy (as shown in Fig. 3.10). At the same time, even though system schemes with "fix P_r" characteristic incur more communication cost, higher testing accuracy can be witnessed at the end of training, compared with their "vary P_r" counterparts.

3.5.4.4 Different values of non-i.i.d. factors

In Fig. 3.12, we vary the non-i.i.d. factor and results are averaged over 10 instances. With 16 quantization bits, $N_{BS} = 128$, $P_{DL,0}/\sigma^2 = 7$ dB, and $P_{UL,0}/\sigma^2 = -7$ dB, performance degradation can be witnessed as the value of non-i.i.d. factor is increased.

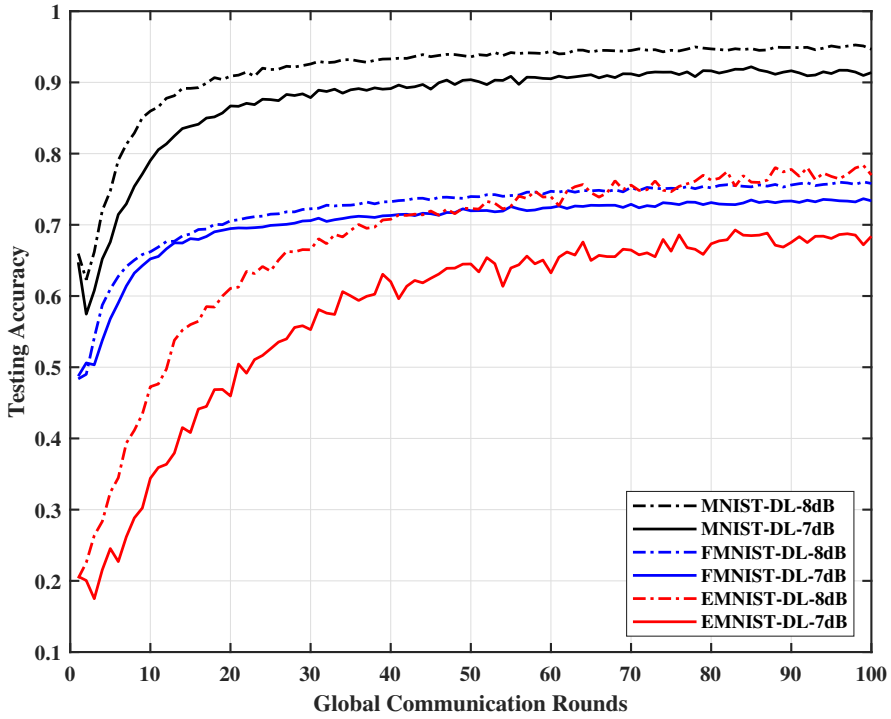


FIGURE 3.14: Testing accuracy at local user side on MNIST, EMNIST and FMNIST datasets. $N_{BS} = 64$ and number of local epochs is 7. Value of UL-SNR is fixed to -3 dB; that is, $P_{UL,0}/\sigma^2 = -3$ dB.

3.5.4.5 Different quantization schemes

In this subsubsection, we consider another quantization scheme based on stochastic rounding [40]. The corresponding result for 8 bits quantization is plotted in Fig. 3.13 and compared with the uniform quantization (16 bits). We utilize two different values (70% and 90%) of non-i.i.d. factor and it can be seen from Fig. 3.13 that for both quantization schemes, the system performances in terms of testing accuracy are approximately same; that is, saving half of the communication cost.

3.5.5 Simulations on EMNIST and FMNIST datasets

In Fig. 3.14, testing accuracies are plotted for i.i.d. MNIST, FMNIST and EMNIST datasets with different DL-SNRs and a fixed UL SNR -3 dB. Number of local epochs is set to be 7. Compared with MNIST and FMNIST, training history for EMNIST dataset has more fluctuations, since EMNIST dataset has more classes (26).

Similarly, FMNIST is also more challenging to classify as compared with MNIST, despite the same number of classes (10) in these two datasets. This is due to the fact that when dimension reduction techniques, e.g., uniform manifold approximation and projection (UMAP) [83] and minimum-distortion embedding (MDE) [84] are used, FMNIST

samples with different labels are more densely distributed than that of MNIST, leading to overlapped classes and reduced testing accuracy.

3.6 Conclusion

In this chapter, we have integrated the federated learning framework with a multi-user massive MIMO 6G network, considering the imperfections of both UL and DL links. At the massive MIMO BS, ZF and MMSE schemes have been used to null the co-channel interference. The tighter convergence bound on the modeling error has been derived in a simpler manner incorporating these imperfections, than the one in the literature. This $\mathcal{O}\left(t_\tau^{-1}\sigma_z^2\right)$ bound also has revealed that DL-SNR and UL-SNR can be increased as training goes on, motivated by which we improve the communication-efficiency of the system by adjusting local training steps, transmit power and quantization bits. Moreover, it has been shown that compared with the reliability of UL links, the quality of DL links plays a more significant role in this system. Simulation results for image recognition task with MNIST, EMNIST, and FMNIST datasets have verified the theoretical insights for fixed and varying SNRs. The testing accuracy has been compared with the scenario with the perfect links. Value of the loss function has also been plotted to verify the $\mathcal{O}\left(t_\tau^{-1}\sigma_z^2\right)$ performance.

Appendix 3.7 Proof of Lemma 3.1

The left hand side of lemma can be simplified as

$$\begin{aligned}
&\mathbb{E}\left\|\bar{\mathbf{w}}_{t+1}-\mathbf{w}^*\right\|_2^2 \\
&=\begin{cases}\mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}-\mathbf{w}^*\right\|_2^2, & \text{if } t+1\notin\mathcal{T},\\ \mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}+\sum_{k=1}^N\alpha_k\bar{\mathbf{z}}_t^k-\mathbf{w}^*\right\|_2^2, & \text{if } t+1\in\mathcal{T},\end{cases}\\
&=\mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}-\mathbf{w}^*\right\|_2^2+\mathbb{E}\left\|\sum_{k=1}^N\alpha_k\bar{\mathbf{z}}_t^k\right\|_2^2\delta_{t+1\in\mathcal{T}}\\
&=\mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}-\mathbf{w}^*\right\|_2^2+W\sigma_z^2\delta_{t+1\in\mathcal{T}},
\end{aligned}$$

where the variance of the effective noise $\sum_{n=1}^{N} \alpha_n \bar{\mathbf{z}}_t^n$ is obtained as

$$
\begin{aligned}
W\sigma_z^2 &= \mathbb{E}\left\| \sum_{n=1}^{N} \alpha_n \bar{\mathbf{z}}_t^n \right\|_2^2 \\
&= \mathbb{E}\left\| \sum_{n=1}^{N} \alpha_n \left(\bar{\mathbf{z}}_{t,UL} + \bar{\mathbf{z}}_{t,DL}^n \right) \right\|_2^2 \\
&= \sum_{k,j=1}^{N} \alpha_k \alpha_j tr\mathbb{E}\left(\bar{\mathbf{z}}_{t,UL} + \bar{\mathbf{z}}_{t,DL}^k \right) \left(\bar{\mathbf{z}}_{t,UL}^H + \bar{\mathbf{z}}_{t,DL}^{jH} \right) \\
&= \sum_{k,j=1}^{N} \alpha_k \alpha_j tr\mathbb{E}\left\{ \bar{\mathbf{z}}_{t,UL} \bar{\mathbf{z}}_{t,UL}^H + \bar{\mathbf{z}}_{t,DL}^k \bar{\mathbf{z}}_{t,DL}^{jH} \right\} \\
&= \frac{\|\boldsymbol{\alpha}\|_2^2 \sigma_w^2}{2} \frac{\sigma^2 N_D}{P_{UL} N_{BS}} tr\mathbf{I}_W + \sum_{n}^{N} \alpha_n^2 tr\mathbb{E}\left\{ \bar{\mathbf{z}}_{t,DL}^n \bar{\mathbf{z}}_{t,DL}^{nH} \right\} \\
&= \frac{\|\boldsymbol{\alpha}\|_2^2 \sigma_w^2}{2} \frac{\sigma^2 N_D}{P_{UL} N_{BS}} W + \frac{\|\boldsymbol{\alpha}\|_2^2 \hat{\sigma}_w^2}{2} \frac{\sigma^2}{P_{DL}} W \\
&= \frac{\|\boldsymbol{\alpha}\|_2^2 \sigma_w^2}{2} \frac{\sigma^2}{P_{DL}} \left[\frac{P_{DL} N_D}{P_{UL} N_{BS}} + \frac{\hat{\sigma}_w^2}{\sigma_w^2} \right] W \\
&= \frac{\|\boldsymbol{\alpha}\|_2^2 \sigma_w^2}{2} \frac{\sigma^2}{P_{DL}} \left[\frac{P_{DL} N_D}{P_{UL} N_{BS}} + \|\boldsymbol{\alpha}\|_2^2 \left(1 + \frac{\sigma^2 N_D}{P_{UL} N_{BS}} \right) \right] W \\
&= \frac{\|\boldsymbol{\alpha}\|_2^2 \sigma_w^2}{2} \frac{\sigma^2}{P_{DL}} \left[\left(\frac{P_{DL}}{\sigma^2} + \|\boldsymbol{\alpha}\|_2^2 \right) \frac{\sigma^2 N_D}{P_{UL} N_{BS}} + \|\boldsymbol{\alpha}\|_2^2 \right] W.
\end{aligned}
$$

Appendix 3.8 Proof of Lemma 3.2

The left-hand side of lemma can be simplified as

$$
\begin{aligned}
&\mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}-\mathbf{w}^*\right\|_2^2 \\
&=\mathbb{E}\left\|\bar{\mathbf{w}}_t-\eta_t\mathbf{g}_t-\mathbf{w}^*\pm\eta_t\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2 \\
&\overset{(a)}{=}\mathbb{E}\left\|\bar{\mathbf{w}}_t-\eta_t\nabla F(\bar{\mathbf{w}}_t)-\mathbf{w}^*\right\|_2^2+\eta_t^2\mathbb{E}\left\|\mathbf{g}_t-\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2 \\
&=\mathbb{E}\left\|\bar{\mathbf{w}}_t-\mathbf{w}^*\right\|_2^2+\eta_t^2\mathbb{E}\left\|\mathbf{g}_t-\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2+\eta_t^2\mathbb{E}\left\|\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2 \\
&\qquad-2\eta_t\mathbb{E}\left(\bar{\mathbf{w}}_t-\mathbf{w}^*\right)^T\nabla F(\bar{\mathbf{w}}_t) \\
&\overset{(b)}{\leq}\mathbb{E}\left\|\bar{\mathbf{w}}_t-\mathbf{w}^*\right\|_2^2+\eta_t^2\mathbb{E}\left\|\mathbf{g}_t-\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2+\eta_t^2\mathbb{E}\left\|\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2 \\
&+2\eta_t\mathbb{E}\left[F(\mathbf{w}^*)-F(\bar{\mathbf{w}}_t)-\frac{\mu}{2}\|\bar{\mathbf{w}}_t-\mathbf{w}^*\|_2^2\right] \\
&\overset{(c)}{\leq}(1-\eta_t\mu)\,\mathbb{E}\left\|\bar{\mathbf{w}}_t-\mathbf{w}^*\right\|_2^2+\eta_t^2\sum_n\alpha_n^2\left(\sigma_n^2+G^2\right) \\
&+2\eta_t\mathbb{E}\left[F(\mathbf{w}^*)-F(\bar{\mathbf{w}}_t)\right] \\
&\overset{(d)}{\leq}(1-\eta_t\mu)\,\mathbb{E}\left\|\bar{\mathbf{w}}_t-\mathbf{w}^*\right\|_2^2+\eta_t^2\sum_n\alpha_n^2\left(\sigma_n^2+G^2\right),
\end{aligned}
$$

where in (a), $\mathbf{g}_t=\sum_{n=1}^N\alpha_n\nabla F_n(\mathbf{w}_t^n,\xi_t^n)$ with $\mathbb{E}\left\{\mathbf{g}_t-\bar{\mathbf{g}}_t\right\}=\mathbf{0}$ is used; (b) is obtained from μ-strong convexity, i.e., $-2(\mathbf{w}-\mathbf{v})^T\nabla F(\mathbf{w})\leq 2F(\mathbf{v})-2F(\mathbf{w})-\mu\|\mathbf{v}-\mathbf{w}\|_2^2$; in (c), the results from Assumptions 3 and 4 are used; in (d), the fact $F(\mathbf{w}^*)\leq F(\bar{\mathbf{w}}_t)$ is used.

Appendix 3.9 Proof of Theorem 3.3

For the L-smooth function, we write

$$
\begin{aligned}
&\mathbb{E}\left\{F\left(\bar{\mathbf{w}}_{t+1}\right)\right\}-F^* && (3.23)\\
&\leq\mathbb{E}\left\{\left(\bar{\mathbf{w}}_{t+1}-\mathbf{w}^*\right)^T\nabla F(\mathbf{w}^*)\right\}+\frac{L}{2}\Delta_{t+1} && (3.24)\\
&\overset{(a)}{=}\frac{L}{2}\Delta_{t+1} && (3.25)\\
&\overset{(b)}{\leq}\frac{L}{2}\cdot\frac{\nu}{t+1+\beta_1}, && (3.26)
\end{aligned}
$$

where in (a), zero mean result $\mathbb{E}\left\{\bar{\mathbf{w}}_t-\mathbf{w}^*\right\}=\mathbf{0}$ is used, and $\Delta_t=\mathbb{E}\|\bar{\mathbf{w}}_t-\mathbf{w}^*\|_2^2$; the inequality, $\Delta_t\leq\frac{\nu}{t+\beta_1}$ in (b) is proved via induction as follows. Let $C=\sum_n\alpha_n^2\left(\sigma_n^2+G^2\right)+W\frac{\sigma_z^2}{\eta_t^2}\delta_{t+1\in\mathcal{S}},\nu\geq\frac{\beta_0^2C}{\beta_0\mu-1},\eta_t=\frac{\beta_0}{t+\beta_1}$, where $\beta_0>\frac{1}{\mu},\beta_1>0$. Then, definition of ν guarantees that the inequality in (b) holds for $t=0$, assume the conclusion holds for some

$t > 0$, by using the results of Lemma 3.1 and Lemma 3.2, we write

$$\Delta_{t+1} \leq (1 - \eta_t \mu)\, \Delta_t + \eta_t^2 C \tag{3.27}$$

$$\leq \frac{\nu}{t + \beta_1} - \frac{\beta_0 \mu \nu}{(t + \beta_1)^2} + \frac{\beta_0^2 C}{(t + \beta_1)^2} \tag{3.28}$$

$$= \frac{\nu(t + \beta_1) - \nu}{(t + \beta_1)^2} + \frac{\nu - \beta_0 \mu \nu + \beta_0^2 C}{(t + \beta_1)^2} \tag{3.29}$$

$$\leq \frac{\nu}{t + 1 + \beta_1} + 0. \tag{3.30}$$

Appendix 3.10 Proof of Corollary 3.4

When learning rate η is a constant, the model also converges, however, with fixed gap from the optimal values. From Eqn. (3.27) in the proof of Theorem 3.3 above, we replace $\eta_t = \eta$ and prove $\Delta_t \leq \frac{\eta C}{\mu}$ via induction as follows

$$\Delta_{t+1} \leq (1 - \eta\mu)\, \Delta_t + \eta^2 C \tag{3.31}$$

$$\leq (1 - \eta\mu)\, \frac{\eta C}{\mu} + \eta^2 C = \frac{\eta C}{\mu}, \tag{3.32}$$

Then, from the inequality in Eqn. (3.26), it follows that using the L-smooth function, we have

$$\mathbb{E}\left\{F\left(\bar{\mathbf{w}}_{t+1}\right)\right\} - F^* \leq \frac{L}{2} \Delta_{t+1} \leq \frac{L}{2} \frac{\eta C}{\mu}.$$

Chapter 4

Federated Distillation in Massive MIMO Networks

4.1 Introduction

In Chapter 3, we focused on the FL algorithm and recognized the ability to decouple the processes of data acquisition and computation of this promising distributed learning algorithm. Such privacy-preserving nature has been investigated and analyzed in existing literature [27–30]. One significant research direction in this topic is integrating FL frameworks with appropriate communication-efficient systems and fully empowering the potentials provided by next-generation communication systems. Meanwhile, all the clients participating in the FL training process must retain local models with the same architecture [39]; FL lacks model heterogeneity among local users.

Recently, by leveraging the knowledge distillation (KD) technique [87], Federated Distillation (FD) [88] has been attracting ever-increasing attention from both industry and academia. Unlike FL, FD allows local users to train nonidentical local models. Instead of model parameters, output logits on the public dataset are utilized as the information exchanged between BS and local users. This not only grants local users more flexibility to trade off their local computing capability and training performance but also reduces the incurred communication cost remarkably; achieving both heterogeneity and efficiency.

Work in this chapter has been published in IEEE Transactions on Cognitive Communications and Networking, March 2024 [85], with a preliminary version presented at IEEE SPAWC September 2023 [86].

To fully leverage the potential of FD, it is crucial to integrate it with the advancements in next-generation air interface (NGAI) and communication technologies. These technologies offer significant benefits that can address the unique challenges posed by FD frameworks, ensuring their effective implementation in real-world scenarios. The advantages provided by the NGAI, such as high-performance networking, hyper-flexible architecture and AI-enhanced security [35], are also vital for accomplishing the requirements of FD frameworks in future communication networks. In particular, the advantages inherent in NGAI massive MIMO systems, such as spatial multiplexing, diversity [89], interference management [90], and energy efficiency [91], are poised to reciprocally benefit the FD algorithm, especially in resource-constrained and heterogeneous environments.

The rest of this chapter is organized as follows. Section 4.2 provides a literature review of the existing research work. Section 4.3 and 4.4 detail the proposed system model and communication process included in the proposed framework, respectively. Convergence analysis is conducted in Section 4.5, followed by the explanation of the proposed communication channel-aware FD scheme in Section 4.6. Simulation results are presented in Section 4.7, and finally Section 4.8 concludes this chapter.

4.2 Related Work

4.2.1 Knowledge distillation-assisted FL/FD frameworks

Some of the existing works have endeavored to incorporate the KD technique in the vanilla FL algorithm and improve corresponding system performance by introducing new flexible aggregation schemes [92–94] or proposing enhanced local distillation/training algorithm [95–99]. Specifically, work in [92] proposes a weighted ensemble distillation algorithm based on private certainty scoring. In [93], in order to mitigate privacy risk, a flexible aggregation scheme is designed and unlabeled datasets are utilized for model fusion. Likewise, the aggregation methodology introduced in [94] leverages the statistical characteristics of uploaded logits and stays robust to the local model corruption issue, where the slow convergence of the local model belonging to one individual user might disseminate its effects to other local users. On the other hand, the work in [95] exploits the transfer learning technique and distillation was only performed at local user side. In [96], a generative model is learnt at server side and broadcasted to local users such that user information is ensembled and aggregated in a data-free manner. In [97], a divergence-based loss function is proposed for the distillation process, and the corresponding convergence analysis is also provided. Similarly, a feature distillation loss is introduced in [98] employing new trainable parameters, where classification and feature-similarity losses are combined and optimized jointly. The work in [99] designs a new variant of vanilla FD algorithm where each user is required to maintain two local

TABLE 4.1: Comparison with FD systems. Abbreviations in the table are defined as: "C": Convergence Analysis, "MH": Model Heterogeneity, $\mathcal{X}$ is the unlabeled public dataset that both local users and server have access to, "SD": Serve side Distillation, "LD": Local user side Distillation, "IE": Information Exchanged, $\mathcal{W}$ denotes that model weights are transmitted, $\mathcal{L}$ denotes that model output logits are transmitted, "N": Receiver Side Noise, "F": Channel Fading, "I": Interference from other users.

Refs.	C/MH/$\mathcal{X}$	SD/LD	IE-UL	IE-DL	UL-N/F/I	DL-N/F/I
[92]	× ✓ ✓	✓ ✓	$\mathcal{W}$	$\mathcal{W}$	× × ×	× × ×
[93]	× ✓ ✓	✓ ×	$\mathcal{W}$	$\mathcal{W}$	× × ×	× × ×
[94]	× ✓ ×	× ✓	$\mathcal{L}$	$\mathcal{L}$	× × ×	× × ×
[95]	× ✓ ✓	× ✓	$\mathcal{L}$	$\mathcal{L}$	× × ×	× × ×
[96]	✓ × ✓	✓ ✓	$\mathcal{W}$	$\mathcal{W}$	× × ×	× × ×
[97]	✓ ✓ ×	× ✓	$\mathcal{W}$	$\mathcal{W}$	× × ×	× × ×
[98]	× ✓ ×	✓ ✓	$\mathcal{L}$	$\mathcal{W}$	× × ×	× × ×
[99]	× ✓ ×	× ✓	$\mathcal{W}$	$\mathcal{W}$	× × ×	× × ×
[100]	× × ✓	✓ ×	$\mathcal{L}$	$\mathcal{W}$	✓ ✓ ×	✓ ✓ ×
[101]	× × ×	× ✓	$\mathcal{L}$	$\mathcal{L}$	✓ ✓ ×	× × ×
[102]	× × ×	× ✓	$\mathcal{L}$	$\mathcal{L}$	✓ ✓ ×	✓ ✓ ×
[103]	× × ✓	✓ ×	$\mathcal{L}$	$\mathcal{W}$	× × ×	× × ×
[104]	× × ✓	× ✓	$\mathcal{L}$	$\mathcal{L}$	× × ×	× × ×
[105]	× ✓ ✓	✓ ✓	$\mathcal{L}$	$\mathcal{L}$	✓ × ×	✓ × ×
[106]	× × ✓	× ✓	$\mathcal{W}$	$\mathcal{W}$	✓ × ×	✓ × ×
Prop.	✓ ✓ ✓	✓ ✓	$\mathcal{L}$	$\mathcal{L}$	✓ ✓ ✓	✓ ✓ ✓

models; that is, teacher model and student model, and only the student models will participate in the training process collaboratively and be shared.

In these works, practical communication aspects are completely ignored (also see in Table 4.1), which typically include quantization, modulation and transmission. At the same time, model parameters are still the information to be exchanged between BS and local users; that is, the communication-efficiency offered by FD paradigm is not exploited in these works (the works in [94] and [95] utilize the logits as the exchanged information for both UL and DL-links, but both of the proposed algorithms require a pre-training phase at local user side, which will also introduce extra communication overhead and latency).

4.2.2 Communication-integrated FD frameworks

Meanwhile, in the direction to integrate vanilla FD paradigm with communication-efficient systems, existing studies are summarized as follows. In [100], in order to

establish communication-efficiency promotion, uplink-downlink capacity asymmetry issue is investigated, where model parameters and model outputs are considered as the exchanged information for DL and UL transmission, respectively. Besides, a mixup technique is employed before local users upload the latest model outputs to further preserve privacy, and model distillation is only performed at server side. Similarly, the work in [101] considers both digital and analog transmission schemes, where Gaussian multiple-access channels are utilized as UL communication link model, leaving DL to be error-free. At the same time, [101] assumes no global model at server side; that is, distillation is only performed at local user side and the BS is only responsible for carrying out logits aggregation. The work in [101] was extended by the same authors later in [102] via considering imperfect DL channels. The works in [103–106] manage to improve communication efficiency by reducing the total UL transmission overhead. Specifically, the works in [103, 104] endeavor to design new schemes such that only the logits with low entropy will be selected, transmitted and participate in the aggregation procedure within each global communication round. Similarly, with same design idea, a soft-label quantization and source coding schemes are proposed and implemented in [105] to reduce the entropy of logits. In [106], a dynamic quantization scheme, where appropriate quantization levels are utilized by different users, is proposed and utilized to carry out model compression and reduce overall communication cost for the UL links.

In Table 4.1, we summarize and compare the algorithms discussed above, the comparison is made for convergence analysis, model heterogeneity, information exchanged via channels, location that distillation is performed, categories of noises considered in the communication process, etc. Specifically, for the communication-integrated frameworks discussed in this subsection [100–106], it can be witnessed that none of them are designed and investigated under a systematic communication scenario with the presence of all types of noises. To name a few, the works in [100] and [102] only consider channel fading and receiver side noise, leaving interference from other users uninvestigated. In the same way, the work in [101] only includes the receiver side noise and channel fading in the UL transmission, leaving DL to be error-free. Besides, the works in [100–104, 106] assume all the users to have a local model with identical structure; that is, the model heterogeneity offered by the FD algorithm is not exploited. On the other hand, even though the work in [105] empowers the local users to maintain models with different architectures, it only takes the imperfections introduced by quantization noise into consideration and corresponding convergence analysis is not conducted. Further, although the works in [96] and [97] provide corresponding convergence analysis, model weights are still treated as the exchanged information between local users and server; that is, the convergence bounds derived are for FL algorithm, rather than FD algorithm.

4.2.3 Contributions

In this chapter, we propose a communication-efficient FD algorithm under a practical multi-user massive multiple-input multiple-output (mMIMO) communication system. To provide communication efficiency, the proposed algorithm focuses on regulating the power for uplink and downlink, intelligently changing the number of training steps per communication round, and varying the quantization and modulation. Unlike the existing works listed in [92–99], where communication channels are assumed perfect and interference-free, we consider both UL and DL communication links to be error-prone while considering modulation and quantization. To the best of our knowledge, we are the first to evaluate FD algorithm under a mMIMO communication system and prove the corresponding convergence. The main contributions of this chapter can be summarized as follows.

- *Massive MIMO-integrated FD algorithm (FedTSKD):* We design and implement a FD algorithm with massive MIMO communication network, accounting imperfections in both UL and DL links. To mitigate the multi-user interference, we utilize zero-forcing (ZF) and minimum mean square error (MMSE) approaches for both uplink and downlink. In the proposed work, logits are treated as the only information exchanged over the imperfect communication links.

  We carry out extensive simulations to demonstrate the effects of various parameters on the FD convergence, testing accuracy, and communication efficiency; these parameters include number of antennas, number of users, UL and DL SNRs, channel fading, quantization, modulation, noise, intra-cell interference, different degrees of non-i.i.d.-ness of available datasets.

  Further, we compare simulation results of the proposed algorithm with two baselines in the existing literature: *1)* Both communication links (DL and UL) are assumed to be error-free [92–99]; *2)* Only uplinks are error-prone, and downlinks error-free [101]. Moreover, we also compare two popular learning algorithms, viz, FL and FD algorithms, where both are integrated with massive MIMO system. Simulations show the FD algorithm incurs 1% of the communication overhead of the FL algorithm, which is significant.

- *Convergence analysis for the proposed FD algorithm:* We provide the convergence analysis for the proposed communication efficient FD algorithm, accounting for imperfect links. The proposed analysis shows that the modeling error of the FD algorithm reduces in $\mathcal{O}\left(t_{\tau}^{-1}\right)$ manner, given the parameters of the learning algorithm satisfy certain conditions. We also show that given the conditions on the learning rate, the number of distillation steps can be adjusted to obtain better convergence bound of the modeling error.

- *Group-based FD algorithm (FedTSKD-G):* Motivated by the heterogeneity of wireless channels, we propose a group-split scheme, which allows the server side model to be aware of the effects resulting from different channel imperfections while making predictions on the public dataset. In addition to that, by transferring the design ideas on adapting the number of local steps, we reduce the overall communication cost incurred during the training process and accelerate the convergence. Simulation results on image classification problem with ImageNette/STL-10, CIFAR-10/STL-10 and MNIST/FMNIST dataset combinations demonstrate both the effectiveness of the proposed framework.

Remark: (Difference to Chapter 3 and our recent work [24]): The work in Chapter 3 and [24] is based on Federated Learning (FL), whereas the work in this chapter focuses on Federated Distillation (FD). We summarize the differences between these two works as follows.

- *Mathematical analysis perspective:* Two different algorithms (FD and FL) have different convergence analyses, which are based on logits and weights respectively. Compared with FL setting, the convergence analysis for FD paradigm in mMIMO system is much more challenging. In both cases, wireless impairment such as fading, interference and noise introduced by the communication transmission process is considered. Intuitively, in FL scenario, the perturbations in model weights will have a discernible impact on the overall model performance. While in the FD setting, the wireless impairments are imposed on the model output (logits), as will be presented in Section 4.5, the underlying relationship between the logits and system performance is more complicated to derive, and this is because, unlike the FL algorithm, in FD, there is no "*direct*" connection between these two factors and this calls for joint consideration of both training and distillation processes.

- *Algorithm design perspective:* The design focus of our work in Chapter 3 is about improving the *communication efficiency*. In this chapter, apart from this, we also emphasize *communication-aware* mechanism. Specifically, let the system be conscious of the communication quality and learn the imperfections introduced by the wireless channels itself. This will be elaborated in detail in Section 4.6.

- *Simulation perspective:* For the simulations in Chapter 3, all local users' model architecture has to be the same, this neglects the heterogeneity in terms of local users' computing capability. In this chapter, different clients' models are considered to be nonidentical to fully exploit the heterogeneity provided by the FD paradigm. Further, more challenging datasets, have been considered in this chapter to ensure and verify the proposed FD algorithm and its convergence.

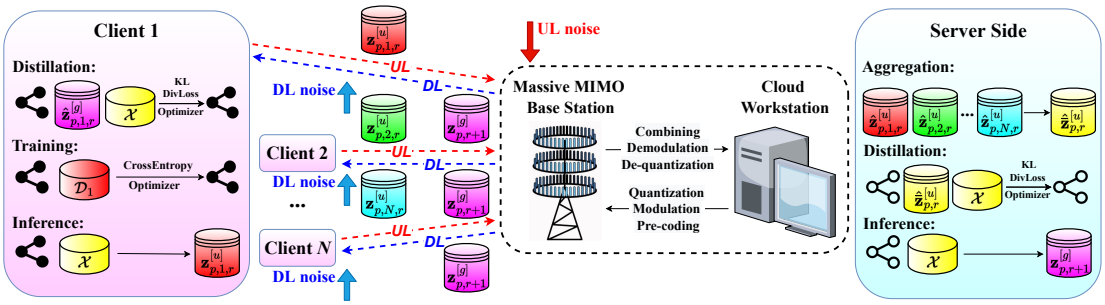


FIGURE 4.1: Architecture of the proposed massive MIMO-integrated FD framework with imperfect communication links, where quantization noise, channel fading, intra-cell interference and receiver side noise are considered. $\mathcal{D}_1$ and $\mathcal{X}$ represent the personal dataset of client 1 and public dataset, respectively. $\mathbf{z}_{p,r+1}^{[g]}$ denotes the predicted logits on the public dataset by server side model at $(r+1)^{th}$ round and $\hat{\mathbf{z}}_{p,n,r+1}^{[g]}$ stands for its corresponding noisy version, received by n^{th} user. Note that all the local users share similar procedures as shown for Client 1, detailed local procedures for Client 2 and Client N are omitted in this figure for better visualization.

4.3 System Model

We consider a wireless single-cell multi-user cellular network with N users and a server or base station (BS). The BS has N_{BS} number of antennas, while each user device is equipped with N_D antennas[1]. The n^{th} user stores a local private labeled dataset $\mathcal{D}_n = \{\mathbf{d}_i = (\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots, D_n\}$, that is, the total dataset across users $\mathcal{D} = \cup_{n=1}^{N} \mathcal{D}_n$ with size $D = \sum_{n=1}^{N} D_n$. Each data point $\mathbf{d}$ includes the input data $\mathbf{x}$ and the corresponding output label $\mathbf{y}$. We assume all users' datasets are not independent and identically distributed (non-i.i.d.). Similarly, $\mathcal{X} = \{\mathbf{x}_{p,i}, i = 1, \ldots, D_p\}$ denote the unlabeled public dataset known to both users and server.

The objective of users is to learn private heterogeneous models based on their local data. Let $\mathbf{z} = F_n(\mathbf{x}; \mathbf{w})$ denote the mapping with learning weights $\mathbf{w}$ from the input data point $\mathbf{x}$ to the output logits $\mathbf{z}$. The loss function between given labels ($\mathbf{y}$) and output logits of the model ($\mathbf{z}$) is assumed as $l(\mathbf{z}, \mathbf{y}) = l(F_n(\mathbf{x}; \mathbf{w}), \mathbf{y})$[2]. The overall loss function for the dataset $\mathcal{D}_n$ can be written as

$$L_n(\mathbf{w}) = \frac{1}{D_n} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_n} l(F_n(\mathbf{x}; \mathbf{w}), \mathbf{y}). \tag{4.1}$$

The aim of the federated distillation can be formulated as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} L_n(\mathbf{w}), \forall n, \tag{4.2}$$

[1] Considering its implementation complexity and resulting training latency [107, 108], in the designed system, we do not include the error correction coding schemes. It should be noted that increasing the number of antennas at BS and user will provide system performance gain in a similar way.

[2] In this chapter, we use cross entropy as local training loss function $l(\cdot)$.

When logits are available rather than labels such as in the distillation steps with the public dataset, Kullback-Leibler (KL) divergence is used for the distillation loss function as [92]

$$Q_n\left(\mathbf{w};\mathcal{Z}\right)=\frac{1}{|\mathcal{Z}|}\sum_{(\mathbf{x}_p,\mathbf{z}_p)\in\mathcal{X}\times\mathcal{Z}} l_{KL}\left(\mathrm{LSM}\left(F_n(\mathbf{x}_p;\mathbf{w})\right),\mathrm{SM}\left(\mathbf{z}_p\right)\right), \tag{4.3}$$

where $\mathbf{z}_p \in \mathcal{Z}$ are the set of logits corresponding to the unlabeled public dataset of inputs $\mathbf{x}_p$. $|\mathcal{Z}|$ denotes the cardinality of set $\mathcal{Z}$. LSM and SM stand for log-softmax and softmax functions, respectively. In the above (4.3), model output logits are utilized, rather than labels, which is due to the fact that labels, the output of softmax layer, lie within the range $(0, 1)$ and sum up to 1. Moreover, using model output logits provides more tolerance while communicating over imperfect links [101, 102, 104, 105].

4.3.1 Assumptions on the loss functions

Following assumptions for both training loss function $L_n(\cdot), n = 1, \ldots, N$ and distillation loss function $Q_n(\cdot), n = 1, \ldots, N$, are made for the convergence analysis. [24], [31, 69, 70].

Assumption 5. (Smoothness)*: $L_1, \ldots, L_N$ are all ψ-smooth, for $\forall \mathbf{v}, \mathbf{w}$, $L_n(\mathbf{v}) \leq L_n(\mathbf{w}) + (\mathbf{v}-\mathbf{w})^T\nabla L_n(\mathbf{w}) + \frac{\psi}{2}\|\mathbf{v}-\mathbf{w}\|_2^2$.*

Assumption 6. (Convexity)*: $L_1, \ldots, L_N$ are all μ-strongly convex, for $\forall \mathbf{v}, \mathbf{w}$, $L_n(\mathbf{v}) \geq L_n(\mathbf{w}) + (\mathbf{v}-\mathbf{w})^T\nabla L_n(\mathbf{w}) + \frac{\mu}{2}\|\mathbf{v}-\mathbf{w}\|_2^2$.*

$Q_1, \ldots, Q_N$ are all μ_q-strongly convex, for $\forall \mathbf{v}, \mathbf{w}$, $Q_n(\mathbf{v}) \geq Q_n(\mathbf{w}) + (\mathbf{v}-\mathbf{w})^T\nabla Q_n(\mathbf{w}) + \frac{\mu_q}{2}\|\mathbf{v}-\mathbf{w}\|_2^2$.

Assumption 7. (Bounded variance)*: The stochastic gradients for each client satisfy*

$$\mathbb{E}\left\|\nabla L_n\left(\mathbf{w}_{n,t},\mathcal{D}_{n,t}\right)-\nabla L_n\left(\mathbf{w}_{n,t},\mathcal{D}_n\right)\right\|_2^2 \leq \sigma_n^2, \tag{4.4}$$

$$\mathbb{E}\left\|\nabla Q_n\left(\mathbf{w}_{n,t},\mathcal{Z}_{n,t}\right)-\nabla Q_n\left(\mathbf{w}_{n,t},\mathcal{Z}_n\right)\right\|_2^2 \leq \sigma_{qn}^2, \tag{4.5}$$

where $\mathcal{D}_{n,t} \subset \mathcal{D}_n$ is the mini-batch (a subset chosen uniformly and independently from n^{th} user's local training dataset $\mathcal{D}_n$ at time step t).

Assumption 8. (Uniformly bounded gradient)*: The squared l_2-norm of gradients satisfies*

$$\mathbb{E}\left\|\nabla L_n\left(\mathbf{w}_{n,t},\mathcal{D}_n\right)\right\|_2^2 \leq G^2, \forall n, \tag{4.6}$$

$$\mathbb{E}\left\|\nabla Q_n\left(\mathbf{w}_{n,t},\mathcal{Z}_n\right)\right\|_2^2 \leq G_q^2, \forall n, \tag{4.7}$$

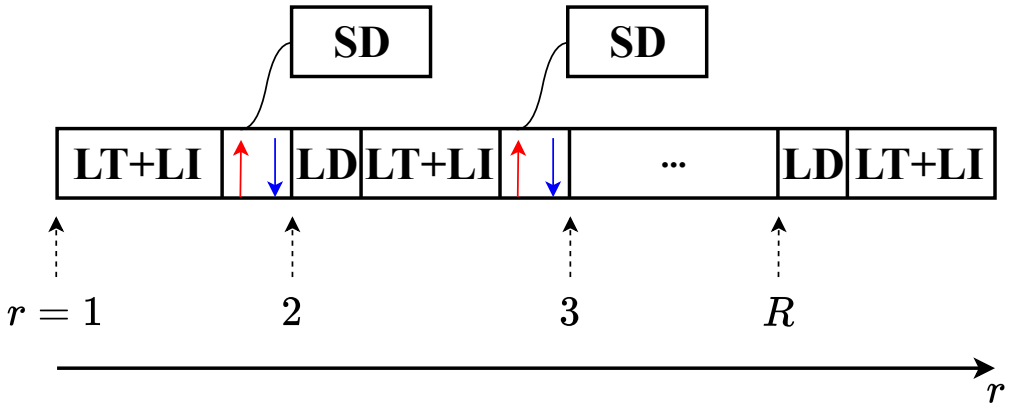


FIGURE 4.2: Illustration of the timeline of communication rounds (r) in federated distillation. Up and down arrows signify the uplink and downlink communications, respectively. Abbreviations are defined as: "LT": Local user side Training, "LI": Local user side Inference, "LD": Local user side Distillation, "SD": Server side Distillation.

Assumptions 5 and 6 are the standard ones for l_2-norm regularized linear regression, logistic regression, and softmax classifier. Assumptions 7 and 8 are based on bounded gradients and are helpful in bounding the model-error variance while analyzing the convergence. In Chapter 3, we made several assumptions that are revisited here due to differences in notation and slight modifications specific to this chapter. Notably, while Assumptions 2, 3, and 4 in Chapter 3 primarily concern the training loss function, this chapter extends these assumptions to include the distillation loss function as well. Rewriting these assumptions here ensures clarity and highlights the necessary changes relevant to the current context.

4.3.2 Learning dynamics

The process of FD learning is divided into three parts as shown in Fig. 4.1, viz, local training step, communication round, server and local distillation steps. The corresponding time slots are illustrated in Fig. 4.2. Let $\mathcal{T} = \{1, 2, \ldots, T\}$ denote the set for all time steps, that is,

$$\mathcal{T} = \mathcal{T}_{LT} \cup \mathcal{T}_{SD} \cup \mathcal{T}_{LD}, \tag{4.8}$$

where $\mathcal{T}_{LT}$, $\mathcal{T}_{SD}$ and $\mathcal{T}_{LD}$ denote the time indices for local training, server and local distillation, respectively[3]; K_r and S represent the numbers of local training and distillation steps in the r^{th} round, respectively. As depicted in Fig. 4.2, the learning process starts with local training with K_r steps using stochastic gradient descent (SGD). The update

[3]In the set definition at the top of next page, short notation $a + (1, \ldots, k)$ for $\{a + 1, \ldots, a + k\}$ is used.

$$\mathcal{T}_{LT} = \left\{ \{1, \ldots, K_1\}, \ldots, \left\{ \sum_{i=1}^{r} K_i + 2rS + (1, \ldots, K_{r+1}) \right\}, \ldots \right\}, \tag{4.12}$$

$$\mathcal{T}_{SD} = \left\{ \{K_1 + (1, \ldots, S)\}, \ldots, \left\{ \sum_{i=1}^{r} K_i + 2rS + K_{r+1} + (1, \ldots, S) \right\}, \ldots \right\}, \tag{4.13}$$

$$\mathcal{T}_{LD} = \left\{ \{K_1 + S + (1, \ldots, S)\}, \ldots, \left\{ \sum_{i=1}^{r} K_i + 2rS + K_{r+1} + S + (1, \ldots, S) \right\}, \ldots \right\}, \tag{4.14}$$

---

equation for weights for each user can be expressed as

$$\mathbf{w}_{n,t+1} = \mathbf{w}_{n,t} - \eta_t \nabla L_n(\mathbf{w}_{n,t}, \mathcal{D}_{n,t}), t \in \mathcal{T}_{LT} \tag{4.9}$$

where η_t is the learning rate. After the private training process, logits are generated based on the public dataset ($\mathcal{X}$) as

$$\mathbf{z}_n = F_n(\mathbf{x}_p; \mathbf{w}_{n,t}), \forall \mathbf{x}_p \in \mathcal{X}, \tag{4.10}$$

and collected in the set $\mathcal{Z}_n$ for each $\mathbf{x}_p \in \mathcal{X}$. These logits are transmitted from each user via uplink noisy channels (red dashed lines in Fig. 4.1) to the server. The server aggregates and distills the logits received from all users in S steps. Then, the server sends the processed logits to the downlink noisy channels (blue dashed lines in Fig. 4.1). Let $\hat{\mathcal{Z}}_n = \{\hat{\mathbf{z}}_n\}$ denote the set of logits received at the n^{th} user. The relation between $\mathcal{Z}$ and $\hat{\mathcal{Z}}_n$ is described in the next section. Subsequently, in the local distillation phase, the received logits are used to update the local models in S steps. Each distillation step can be given as

$$\mathbf{w}_{n,t+1} = \mathbf{w}_{n,t} - \eta_t \nabla Q_n(\mathbf{w}_{n,t}, \hat{\mathcal{Z}}_{n,t}), t \in \mathcal{T}_{LD}, \tag{4.11}$$

where $\hat{\mathcal{Z}}_{n,t} \subset \hat{\mathcal{Z}}_n$ is the mini-batch. This distillation step is followed by similar local training steps and so on.

4.4 Uplink and Downlink Communications

In the following subsections, we analyze the error introduced by the massive MIMO communication for the transmitted logits ($\mathbf{z}_{n,t}$), which are the information exchanged between local users and BS. Building on the analysis presented in Chapter 3, we assume a large number of antennas at the BS and employ the ZF precoding/combining approach at the BS. We use $\phi_n(\cdot)$ to represent the distortion arising due to uplink and downlink communications and this function can be decomposed into three components: uplink

operation $\phi_{n,UL}(\cdot)$, server aggregation and distillation $\phi_{av}(\cdot)$, and downlink broadcast operation $\phi_{n,DL}(\cdot)$, that is; $\phi_n\left(\mathbf{z}_{n,t}, \forall n\right) = \phi_{n,DL} \circ \phi_{av} \circ \phi_{n,UL}\left(\mathbf{z}_{n,t}, \forall n\right)$.

Although the UL and DL communication procedures for logits are similar to those discussed in Chapter 3, we will briefly review them here due to the different notions used in this chapter. Additionally, in the server aggregation mapping $(\phi_{av}(\cdot))$, we will introduce the additional distillation operation, highlighting its impacts on the received logits.

4.4.1 Uplink mapping

The uplink process first involves resizing $D_p \times C$ real-valued logits $\mathbf{z}_{n,t}$ into complex-valued data streams for transmission, that is; $\mathbf{Z}_{n,t} \in \mathbb{C}^{N_D \times \tau_{UL}}$, where $\tau_{UL} = \frac{D_p C}{2N_D}$. We assume entries of $\mathbf{Z}_{n,t}$ have zero mean and σ_z^2 variance. $\mathbf{\Omega}_t$ denotes zero-mean Gaussian noise with σ^2 variance. In this way, the received $(\mathbf{\Gamma}_t^{(UL)})$ and estimated $(\hat{\mathbf{Z}}_t)$ signals at the massive MIMO BS can be written as

$$\mathbf{\Gamma}_t^{(UL)} = \sum_{n=1}^{N} \mathbf{H}_t^n \mathbf{Z}_{n,t} \sqrt{\frac{P_{UL}}{N_D \sigma_z^2}} + \mathbf{\Omega}_t^{(UL)}, \tag{4.15}$$

$$\begin{aligned} \hat{\mathbf{Z}}_t &= \sqrt{\frac{N_D \sigma_z^2}{P_{UL}}} \left(\mathbf{H}_t^H \mathbf{H}_t\right)^{-1} \mathbf{H}_t^H \mathbf{\Gamma}_t^{(UL)} \\ &= \mathbf{Z}_t + \sqrt{\frac{N_D \sigma_z^2}{P_{UL}}} \left(\mathbf{H}_t^H \mathbf{H}_t\right)^{-1} \mathbf{H}_t^H \mathbf{\Omega}_t^{(UL)}, \end{aligned} \tag{4.16}$$

where P_{UL} is the transmit power constraint for each device such that $tr\mathbb{E}\left\{\mathbf{Z}_{n,t}\mathbf{Z}_{n,t}^H\right\} = \tau_{UL} N_D \sigma_z^2$. In Eqn. (4.16), $\mathbf{H}_t = \left[\mathbf{H}_t^1, \ldots, \mathbf{H}_t^N\right]$ and $\mathbf{Z}_t^T = \left[\mathbf{Z}_{1,t}^T, \ldots, \mathbf{Z}_{N,t}^T\right]$. Using the channel hardening that $\frac{1}{N_{BS}} \mathbf{H}_t^H \mathbf{H}_t \rightarrow \mathbf{I}$, , the variable $\hat{\mathbf{Z}}_t$ can be used to express the estimated model outputs logits for each user at the BS as $\hat{\mathbf{z}}_{n,t} = \mathbf{z}_{n,t} + \boldsymbol{\omega}_{n,t,UL}$, where $\boldsymbol{\omega}_{n,t,UL} \sim \mathcal{N}\left(\mathbf{0}, \frac{N_D \sigma_z^2 \sigma^2}{2 P_{UL} N_{BS}} \mathbf{I}\right)$.

Remark: In the analysis, we assume that the logits follow the Gaussian distribution, we made the following justifications for this:

1. Logits are the outputs of fully connected layers, which are the weighted sums of activations from previous layers. These activations are nonlinear transformations of weighted sums of input features. According to the central limit theorem [67], when a large number of independent variables are combined linearly, their sum tends to follow a Gaussian distribution, even if the individual variables do not. Therefore, assuming the logits follow a Gaussian distribution is reasonable, especially when the model architecture is complicated and has many neurons and layers.

2. According to [109], the distribution of activations and logits often approximates a Gaussian distribution, particularly after batch normalization (which are used in the models considered in this chapter). This approximation is valid when the model is well-trained and avoids overfitting problems. Further, the work in [110] also provides an empirical study for this by exploiting the random matrix theory.

3. Such assumption also helps to enhance analytical tractability, allowing us to derive a closed-form solution for the variance of the overall effective noise. This simplifies the theoretical analysis and offers valuable insights into the system's behavior.

4.4.2 Server aggregation and distillation mapping

We assume server distillation of logits from all users is similar to aggregation with an uncertainty $\boldsymbol{\omega}_D$, defined by a Gaussian distribution having zero mean and small variance σ_D^2. Therefore, having $\hat{\mathbf{Z}}_t$ at the BS, the averaging and distillation operation can be given as

$$\phi_{av}\left(\hat{\mathbf{Z}}_t\right) = \sum_{n=1}^{N} \alpha_n \hat{\mathbf{z}}_{n,t} + \boldsymbol{\omega}_D \tag{4.17}$$

$$= \bar{\mathbf{z}}_t + \bar{\boldsymbol{\omega}}_{t,UL} + \boldsymbol{\omega}_D \tag{4.18}$$

$$= \bar{\mathbf{z}}_t + \bar{\boldsymbol{\omega}}_{t,av}, \tag{4.19}$$

where $\bar{\mathbf{z}}_t = \sum_{n=1}^{N} \alpha_n \mathbf{z}_{n,t}$, $\alpha_n = \frac{D_n}{\sum_{n=1}^{N} D_n}$ is the weighting factor for n^{th} user and $\|\boldsymbol{\alpha}\|_2^2 = \sum_{n=1}^{N} \alpha_n^2$. $\bar{\boldsymbol{\omega}}_{t,UL} \sim \mathcal{N}\left(\mathbf{0}, \frac{\|\boldsymbol{\alpha}\|_2^2 \sigma_z^2}{2} \frac{\sigma^2 N_D}{P_{UL} N_{BS}} \mathbf{I}\right)$ and $\bar{\boldsymbol{\omega}}_{t,av} \sim \mathcal{N}\left(\mathbf{0}, \left[\frac{\|\boldsymbol{\alpha}\|_2^2 \sigma_z^2}{2} \frac{\sigma^2 N_D}{P_{UL} N_{BS}} + \sigma_D^2\right] \mathbf{I}\right)$

4.4.3 Downlink mapping

In DL transmission, the aggregated vector, say $\hat{\bar{\mathbf{z}}}_t = \phi_{av}\left(\hat{\mathbf{Z}}_t\right)$, is broadcasted to each user device. After resizing the vector $\hat{\bar{\mathbf{z}}}_t$ into a complex-valued matrix $\hat{\bar{\mathbf{Z}}}_t \in \mathbb{C}^{N_D \times \tau_{DL}}$ with $\tau_{DL} = \frac{D_p C}{2N_D}$, the received signal at the n^{th} device can be given as

$$\boldsymbol{\Gamma}_{n,t}^{(DL)} = \mathbf{H}_t^{nH} \mathbf{F}_t \hat{\bar{\mathbf{Z}}}_t p_{DL} + \boldsymbol{\Omega}_{n,t}^{(DL)}, \tag{4.20}$$

where $\mathbf{F}_t$ denotes the $N_{BS} \times N_D$ precoder. For ZF scheme, $\mathbf{F}_t = \frac{1}{\sqrt{N_{BS}}} \mathbf{H}_t^n \left(\mathbf{H}_t^{nH} \mathbf{H}_t^n\right)^{-1}$. The transmit power constraint (P_{DL}) is given as

$$p_{DL}^2 = \frac{P_{DL} \tau_{DL}}{tr\mathbb{E}\left\{\mathbf{F}_t \hat{\bar{\mathbf{Z}}}_t \hat{\bar{\mathbf{Z}}}_t^H \mathbf{F}_t^H\right\}} = \frac{P_{DL}}{\hat{\sigma}_z^2 N_D}, \tag{4.21}$$

where $\hat{\sigma}_z^2 = \|\boldsymbol{\alpha}\|_2^2 \sigma_z^2 \left(1 + \frac{\sigma^2 N_D}{P_{UL} N_{BS}}\right) + \sigma_D^2$. Then, the estimation at the n^{th} user can be obtained as $\left(\mathbf{H}_t^{nH} \mathbf{F}_t\right)^{-1} \boldsymbol{\Gamma}_{n,t}^{(DL)} p_{DL}^{-1}$. Similar to UL transmission, we can write $\phi_{n,DL}\left(\hat{\bar{\mathbf{z}}}_t\right) = \hat{\bar{\mathbf{z}}}_t + \bar{\boldsymbol{\omega}}_{n,t,DL}$, where $\bar{\boldsymbol{\omega}}_{n,t,DL} \sim \mathcal{N}\left(\mathbf{0}, \frac{\hat{\sigma}_z^2 \sigma^2}{2 P_{DL}} \mathbf{I}\right)$ is the Gaussian noise at the typical user.

Therefore, the composite uplink-downlink aggregation process can be expressed as

$$\phi_n\left(\mathbf{z}_{n,t}, \forall n\right) = \phi_{n,DL} \circ \phi_{av} \circ \phi_{n,UL}\left(\mathbf{z}_{n,t}, \forall n\right) \tag{4.22a}$$

$$= \phi_{n,DL} \circ \phi_{av}\left(\hat{\mathbf{Z}}_t\right) \tag{4.22b}$$

$$= \phi_{n,DL}\left(\bar{\mathbf{z}}_t + \bar{\boldsymbol{\omega}}_{t,av}\right) \tag{4.22c}$$

$$= \bar{\mathbf{z}}_t + \underbrace{\bar{\boldsymbol{\omega}}_{t,av} + \bar{\boldsymbol{\omega}}_{n,t,DL}}_{\bar{\boldsymbol{\omega}}_{n,t}}, \tag{4.22d}$$

where $\bar{\mathbf{z}}_t = \sum_{n=1}^{N} \alpha_n \mathbf{z}_{n,t}$, and proof of the variance for the effective noise $\bar{\boldsymbol{\omega}}_{n,t}$, $\sigma_{\bar{\omega}}^2$, can be seen in Appendix 4.9.

4.5 Convergence Analysis

In this section, we present our convergence analysis for the FD paradigm, where logits are treated as the only information exchanged between the server and local users. We consider the scenario where weights are updated at the end of each time slot.

Although the convergence analysis in [96] also has a similar objective function, the upper bound is derived with respect to data distribution, number of data samples and users, not for the time step t as we have done in this work. On the other hand, the convergence analysis provided in work [97] only includes the distillation module, not the whole algorithm.

The key idea of our proof is to consider both training and distillation processes. First, we upper-bound the next step error in Lemma 4.1 and Lemma 4.3 by utilizing the assumptions discussed in Section 4.3. Further, for both cases, with the help of induction, we bound the term of $\mathbb{E}\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\|_2^2$ and $\mathbb{E}\|\mathbf{w}_{n,t+2S} - \mathbf{w}_n^*\|_2^2$ in a unified format of $\mathcal{O}(t^{-1})$, as will be shown in Corollary 4.2 and Corollary 4.4, respectively. These two results will be used jointly to derive our main Theorem 4.5, which bounds the values of the global loss function with respect to the time steps.

We now begin to present our convergence analysis. First, identifying the fact that in the server side distillation process, local model weights are not updated, we have, $\mathbf{w}_{n,t} = \mathbf{w}_{n,t-S}, t = \tau(S), \tau \in \mathcal{T}_{SD}$.

Therefore, considering the unchanged local weights during the server distillation process, we can write the relation of the weights during server and local distillation rounds as

$$\mathbf{w}_{n,t+2S} = \mathbf{w}_{n,t} - \sum_{i=0}^{S-1} \eta_{t+S+i} \nabla Q_n(\mathbf{w}_{n,t+S+i}, \hat{\mathcal{Z}}_{n,t+S+i}),$$

where $t = \tau(K_r), \tau \in \mathcal{T}_{LT}$.

In the following lemma, bounded variance and bounded gradient form Assumptions 7 and 8 are utilized to bound the next step error during the local training process.

Lemma 4.1. *(Next step update for local training process): Let Assumptions 5-8 hold. Then, when $t+1 \in \mathcal{T}_{LT}$, with respect to the optimized model weights $\mathbf{w}_n^*$, the squared l_2-norm of the next step model update error can be bounded as*

$$\mathbb{E}\left\|\mathbf{w}_{n,t+1} - \mathbf{w}_n^*\right\|_2^2 \leq (1 - \eta_t \mu)\, \mathbb{E}\left\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\right\|_2^2 + \eta_t^2 \left(\sigma_n^2 + G^2\right). \tag{4.23}$$

Proof is provided in Appendix 4.10. In addition, we can further bound the term $\mathbb{E}\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\|_2^2$ with respect to time step t as follows (proof will be provided in the appendix), and this result will be used when we derive the upper bound of the modeling error for the proposed FD algorithm.

Corollary 4.2. *From Lemma 4.1, the term $\mathbb{E}\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\|_2^2$ can be further bounded as*

$$\mathbb{E}\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\|_2^2 \leq \frac{\nu_n}{t + \beta_1} \tag{4.24}$$

where $\nu_n \geq \frac{\beta_0^2 C_n}{\beta_0 \mu - 1}, C_n = \left(\sigma_n^2 + G^2\right), \eta_t = \frac{\beta_0}{t+\beta_1}$, $\beta_0 > \frac{1}{\mu}, \beta_1 > 0$.

Proof is derived in Appendix 4.11. Similarly, for the server and local distillation processes, we have the following bound.

Lemma 4.3. *(Next step update for server and local distillation processes): Let Assumptions 5-8 hold. Then, at the end of local training, when $t = \tau\left(K_r\right), \tau \in \mathcal{T}_{LT}$, with respect to the optimized model weights $\mathbf{w}_n^*$, the squared l_2-norm of the next step model update error can be bounded as*

$$\begin{aligned}\mathbb{E}\left\|\mathbf{w}_{n,t+2S} - \mathbf{w}_n^*\right\|_2^2 \leq & \left(1 - \mu_q \sum_{i=0}^{S-1} \eta_{t+S+i}\right) \mathbb{E}\left\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\right\|_2^2 \\ & + \left(\sum_{i=0}^{S-1} \eta_{t+S+i}^2\right) S\left(\sigma_{qn}^2 + G_q^2\right).\end{aligned} \tag{4.25}$$

Proof is provided in Appendix 4.12.

Corollary 4.4. *From Lemma 4.3, the term* $\mathbb{E}\|\mathbf{w}_{n,t+2S} - \mathbf{w}_n^*\|_2^2$ *can be further bounded as*

$$\mathbb{E}\|\mathbf{w}_{n,t+2S} - \mathbf{w}_n^*\|_2^2 \leq \frac{\nu_n}{t + 2S + \beta_1} \tag{4.26}$$

where $\nu_n \geq \frac{S^2\beta_0^2 C_{qn}}{\beta_0\mu_q - 2S}, C_{qn} = \left(\sigma_{qn}^2 + G_q^2\right), \eta_t = \frac{\beta_0}{t+\beta_1}, \beta_0 > \frac{2S}{\mu_q}, \beta_1 \geq 4.$

Proof is derived in Appendix 4.13.

The above results indicate that if μ, μ_q and η_t are chosen properly, one can show the model updates lead to non-increasing errors; that is, the convergence of the proposed FD algorithm is guaranteed. The results of the above lemmas are aggregated and used in the following theorem, which employs the smoothness of loss functions (Assumption 5).

Theorem 4.5. *Let Assumptions 5-8 hold. Then, the proposed FD algorithm satisfies*

$$\mathbb{E}\left\{L_n\left(\mathbf{w}_{n,t+1}\right)\right\} - L_n^* \leq \frac{\psi}{2} \cdot \frac{\nu_n}{t + 1 + \beta_1}, \tag{4.27}$$

where $\nu_n \geq \max\left(\frac{\beta_0^2 C_n}{\beta_0\mu - 1}, \frac{S^2\beta_0^2 C_{qn}}{\beta_0\mu_q - 2S}\right), C_{qn} = \left(\sigma_{qn}^2 + G_q^2\right), C_n = \left(\sigma_n^2 + G^2\right), \beta_0 > \max\left(\frac{1}{\mu}, \frac{2S}{\mu_q}\right),$ $\beta_1 \geq 4.$

Proof is provided in Appendix 4.14.

Theorem 4.5 tells that in the proposed FD algorithm, modeling error reduces in $\mathcal{O}\left(t_\tau^{-1}\right)$ manner, where $t_\tau = t - t_0$ for $t_0 < t$. The derived convergence bound relies on the values of ψ, μ, C_n and C_{qn}, which are from the statistical properties of L_n and Q_n; that is, the loss functions should be selected carefully. The convergence bound linearly depends on the ψ-smoothness, whereas it is inversely proportional to the μ-convexity; that is, the stronger the convexity is, the lower the convergence error is. Further, from the above theorem, it can be concluded that the loss value at the convergence (ν_n) depends on S, the number of distillation steps. When S, L_n and Q_n are selected such that $\nu_n \geq \frac{S^2\beta_0^2 C_{qn}}{\beta_0\mu_q - 2S}$ and $\beta_0 > \frac{2S}{\mu_q}$, for a given time step t, the modeling error convergence bound increases in $\mathcal{O}\left(S^3\right)$ manner. On the other hand, when $\nu_n \geq \frac{\beta_0^2 C_n}{\beta_0\mu - 1}$ and $\beta_0 > \frac{2S}{\mu_q}$, we get $\mathcal{O}\left(S\right)$ bound. For $\beta_0 > \frac{1}{\mu}$, if $\nu_n \geq \frac{\beta_0^2 C_n}{\beta_0\mu - 1}$, the bound is independent of S. This is an interesting finding since this implies that the loss functions for local training and distillation could be dynamically changed during the training process to accelerate the model convergence and performance. However, due to limited space, we would like to delegate further study to our future investigation.

In Algorithm 2, we summarize the proposed massive MIMO-integrated training steps varied FD algorithm (FedTSKD), where for each round (r), only the uplink transmission takes place after the local updates. Lines 3-5 denote the downlink transmission from

Algorithm 2 Dynamic Training Steps based FD Algorithm (FedTSKD)

1: **for** $r = 1, \ldots, R$ **do**
2: **if** $r \neq 1$ **then**
3: Downlink transmission:
4: 1. BS quantizes, modulates and precodes logits.
5: 2. BS broadcasts $\mathbf{z}_{p,r}^{[g]}$ to all users.
6: Local distillation:
7: 1. Local users apply softmax function to the received logits.
8: 2. Local users perform local distillation.
9: **end if**
10: **for** $n = 1, \ldots, N$ **do**
11: **for** $t = \mathcal{T}_{LT}(r)$ **do**
12: Update local weights as:
13: $\mathbf{w}_{n,t+1} = \mathbf{w}_{n,t} - \eta_t \nabla L_n(\mathbf{w}_{n,t}, \mathcal{D}_{n,t})$
14: **end for**
15: n^{th} user performs inference on the public dataset and gets corresponding logits $\mathbf{z}_{p,n,r}^{[u]}$.
16: **end for**
17: Uplink transmission:
18: 1. Local users quantize and modulate logits.
19: 2. Users upload logits to server.
20: 3. BS combines, demodulates and dequantizes the received logits.
21: Logits aggregation at BS:
22: 1. BS aggregates the logits sent from users to produce $\hat{\bar{\mathbf{z}}}_{p,r}^{[u]}$.
23: 2. $\hat{\bar{\mathbf{z}}}_{p,r}^{[u]}$ is passed through softmax function.
24: Server side model performs distillation.
25: Server side model predicts on the public dataset and obtains $\mathbf{z}_{p,r+1}^{[g]}$.
26: **end for**

BS to users, where quantization, modulation and precoding are performed for distilled logits. After getting logits at the user side, users process it locally using softmax-function and distillation as shown in lines 6-8. These logits are further updated via local training steps simultaneously for each user, where the number of steps, denoted by $\mathcal{T}_{LT}(r)$, can differ each round, and can be dynamically chosen based on the training progress history. Subsequently, to prepare for federated transmission, logits are obtained by operating on the public dataset. These logits are quantized and modulated. The received logits at the BS are aggregated and processed using softmax function. After BS's distillation, the public dataset's logits are obtained again for the downlink process, given in lines 3-5.

It can be noted that in order to obtain and improve communication-efficiency in the proposed FD algorithm, we decrease the number of local training steps as the rounds proceed (shown in lines 11-14), i.e., $K_1 \geq K_2 \geq \cdots \geq K_r \geq \cdots \geq K_R$. It can be seen

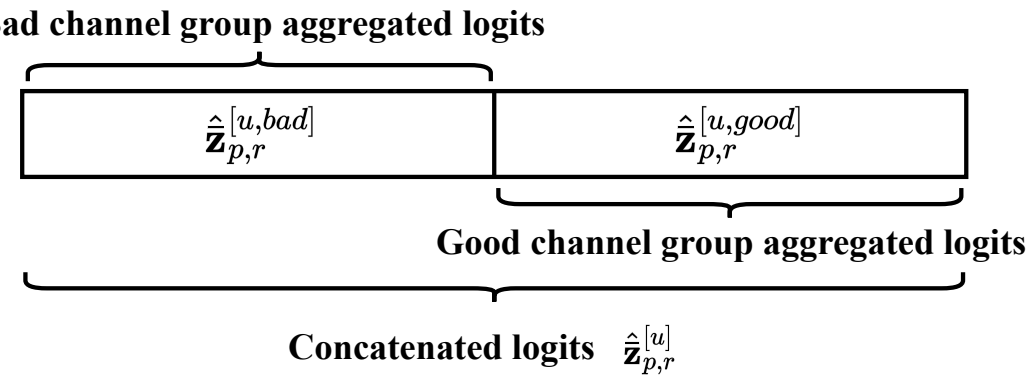


FIGURE 4.3: Concatenation of logits at server side.

from simulations that near convergence, reducing local iterations cannot further improve the performance, e.g., $K_r = K_{r+1}$.

4.6 Communication Channel-aware FD Algorithm

In this section, we propose group-based communication channel-aware FD algorithm, where the FD system is made adaptive to the different communication links. Further, Algorithm 1 is improvised including these changes.

For the group-based communication channel-aware FD framework (FedTSKD-G), the main design idea is to include the effect of channel quality within the distillation process and let the model learn different channel conditions intrinsically at the server side. The received logits information from users with different channel strengths are processed separately.

In wireless communications, due to the reflection, diffraction, and scattering effects caused by the obstacles, when transmitted through wireless medium, the attenuation experienced by different signals from various transmitters/receivers will vary, leading to inconsistent channel quality among users. Specifically, two channel qualities, namely "good" and "bad" are considered as shown in the Fig. 4.3.

These different channel strengths can degrade the logits communications, and in turn can deteriorate the performance of the FD system, since in the local training steps and in distillation, the UL and DL errors are propagated. Note that at the BS, logits from users of good and bad channel strengths are aggregated/merged together. It means users with relatively good channels will also suffer, leading to slower convergence with worse testing accuracy.

To tackle this unfair aggregation, received logits at BS from two groups of users (good and bad) are aggregated separately, providing $\hat{\tilde{\mathbf{z}}}_{p,r}^{[u,bad]}$ and $\hat{\tilde{\mathbf{z}}}_{p,r}^{[u,good]}$. These steps are included in the Algorithm 3, which is improvised over Algorithm 2. These aggregated logits

Algorithm 3 Group-based FD Algorithm (FedTSKD-G)

1: **for** $r = 1, \ldots, R$ **do**
2: **if** $r \neq 1$ **then**
3: Downlink transmission:
4: 1. BS splits the predicted logits into two groups.
5: 2. BS quantizes, modulates and precodes logits.
6: 3. BS broadcasts $\mathbf{z}_{p,r}^{[g,bad]}$ and $\mathbf{z}_{p,r}^{[g,good]}$ to target users.
7: Local distillation:
8: 1. Local users apply softmax function to the received logits.
9: 2. Local users perform local distillation.
10: **end if**
11: **for** $n = 1, \ldots, N$ **do**
12: **for** $t = \mathcal{T}_{LT}(r)$ **do**
13: Update local weights as:
14: $\mathbf{w}_{n,t+1} = \mathbf{w}_{n,t} - \eta_t \nabla L_n(\mathbf{w}_{n,t}, \mathcal{D}_{n,t})$
15: **end for**
16: n^{th} user performs inference on the public dataset and gets corresponding logits $\mathbf{z}_{p,n,r}^{[u]}$.
17: **end for**
18: Uplink transmission:
19: 1. Local users quantize and modulate logits.
20: 2. Users upload logits to server.
21: 3. BS combines, demodulates and dequantizes the received logits.
22: Logits manipulation at BS:
23: 1. BS aggregates the logits sent from bad and good channel groups, respectively.
24: 2. BS concatenates the aggregated logits from these two groups and obtains $\hat{\hat{\mathbf{z}}}_{p,r}^{[u]}$.
25: 3. $\hat{\hat{\mathbf{z}}}_{p,r}^{[u]}$ is passed through softmax function.
26: Server side model performs distillation.
27: Server side model predicts on the public dataset and obtains $\mathbf{z}_{p,r+1}^{[g]}$.
28: **end for**

are then concatenated for distillation (see lines 22-26), that is, $\hat{\hat{\mathbf{z}}}_{p,r}^{[u]} = [\hat{\hat{\mathbf{z}}}_{p,r}^{[u,bad]}, \hat{\hat{\mathbf{z}}}_{p,r}^{[u,good]}]$. Accordingly, the output dimension of the last fully connected layer of the server-side model is doubled. In this way, the server-side model learns the influences of two different channel qualities on received logits. After distillation, predicted logits $\mathbf{z}_{p,r}^{[g]}$ of the public dataset are split into two groups ($\mathbf{z}_{p,r}^{[g,bad]}$ and $\mathbf{z}_{p,r}^{[g,good]}$), and broadcasted to target users via downlink transmission. For example, compared with the logits from the good channel group, the logits from the bad channel group will experience more distortion. Accounting for the logits concatenation mechanism mentioned above and the distillation process at BS, the BS side model will attempt to mitigate and recover the distortion in the received logits from the bad channel group.

In practical wireless communication scenarios, accurately performing channel estimation and distinguishing between user groups based on their channel conditions becomes increasingly challenging as the number of channel quality classes grows. To simplify the analysis, in this chapter, we focus on a basic example with only two classes of channel conditions, i.e., good and bad channel groups.

4.7 Simulation Results

In the simulation setup, each user learns a convolutional neural network (CNN) model for image classification task on private datasets ImageNette [111], CIFAR-10 [112] and MNIST [75], where for federated distillation, unlabeled STL-10 [113] and FMNIST [77] are used as the public datasets. CIFAR-10 and STL-10 datasets contain 10 classes of natural color images with sizes 32×32 and 96×96, whereas MNIST dataset includes handwriting number images 0 to 9 of size 28×28, and fashion items in FMNIST dataset consists of gray-scale images of size 28×28 with 10 classes. ImageNette is a subset of ImageNet [114] with 10 classes, and the average image resolution is 469×387.

Number of users N is set to be 15, each with single antenna $N_D = 1$; number of global communication rounds R is 200; the BS has $N_{BS} = 64$ antennas. For i.i.d. multi-path Rayleigh fading model, Gaussian noise is assumed to have zero mean and unit variance at both local user and BS sides. Uniform quantization scheme with 8 bits is employed to quantize the predicted logits, followed by Quadrature Phase Shift Keying (QPSK) for modulation. To obtain two groups of users, a path loss factor (0.3) is multiplied with CSI strength to simulate the channels with relatively bad quality. Full participation is considered; that is, all local users join the training process for each global communication round, since the communication overhead required in the FD algorithm is much less than that in the FL scheme.

CIFAR-10 + STL-10 datasets: For private datasets of users, $50,000$ samples of CIFAR-10 dataset are split in non-i.i.d. manner using Dirichlet distribution [115] with the parameter α_d enabling us to control the degree of non-i.i.d.-ness. Default value of α_d is 0.5, unless stated otherwise. A lower value of α_d implies a higher degree of non-i.i.d.-ness. Note that the testing sets have i.i.d. data samples. For the unlabeled public dataset, which is common among users, a set with 2000 images from STL-10 dataset is taken. The Adam optimizer [116] is used with learning rate $\eta = 0.001$ for both distillation and training. Batch sizes for training and distillation are set to be 128 and 500, respectively. For data augmentation, each sample is randomly flipped horizontally with probability of 0.5. Number of epochs is set to 2 for both distillation and training. When the number of training steps is varied, for the first 100 global communication rounds, we set $K_r = \frac{D_n}{|\mathcal{D}_{n,t}|}\left(5 - \lfloor \frac{r-1}{25} \rfloor\right)$, where $\lfloor\rfloor$ denotes floor operation, and $\left|\mathcal{D}^{n,t}\right|$ is the cardinality of $\mathcal{D}_{n,t}$. To analyze the models' heterogeneity provided by FD, local

TABLE 4.2: Model architectures

	CNN_1	CNN_2	CNN_3
Layer 1	Conv 1-32	Conv 1-12	Conv 1-8
Layer 2	Maxpool	Maxpool	Maxpool
Layer 3	Conv 32-64	Conv 12-24	Conv 8-16
Layer 4	Maxpool	Maxpool	Maxpool
Layer 5	FC-128	FC-64	FC-64
Layer 6	FC-10	FC-10	FC-10
Layer 7	Dropout	Dropout	Dropout
# of parameters	1,199,882	78,714	25,034

model at each user is chosen from one of three architectures ResNet [117], MobileNet [118] and ShuffleNet [119] CNNs.

ImageNette + STL-10 datasets: For the ImageNette dataset, we consider i.i.d. private dataset distribution and 9469 training images are apportioned among users. All other parameters and configurations in our simulation align with those employed in the CIFAR-10 + STL-10 scenario.

MNIST + FMNIST datasets: In the MNIST dataset, 3000 sample images are divided among N users for non-i.i.d. private datasets using Dirichlet distribution. Uniform random rotation (ranging from -20 degree to 20 degree), random horizontal translation (ranging from -3 pixels to 3 pixels) and random vertical translation (ranging from -3 pixels to 3 pixels) are performed on each training image for data augmentation. Batch size is set to be 20 for training and 500 for distillation. For distillation, a public dataset is constructed with 2000 images from FMNIST dataset. Local user models follow a standard conventional CNN architecture which includes several stacks of convolutional layer, ReLU activation layer and max-pooling layer. For model heterogeneity, model structures are varied among users and detailed model configurations can be found in Table 4.2 [4], where the values in the last row denote the total number of trainable parameters for each model.

In the following simulation results, for better visualization of the results, moving average (with a window size of 10) is plotted along with the actual values using the corresponding transparent colors. We use CUDA version 11 and Pytorch 1.7 [11] as libraries for training the models. Simulations are run on the system with one Intel i9-10980XE CPU (3 GHz) and one NVIDIA GeForce RTX 3090 GPU (24 GB).

[4]Each user maintains one of these three CNN models locally. For all the CNNs, the convolutional layers contain kernels of size 3×3 and are followed by a ReLU activation layer.

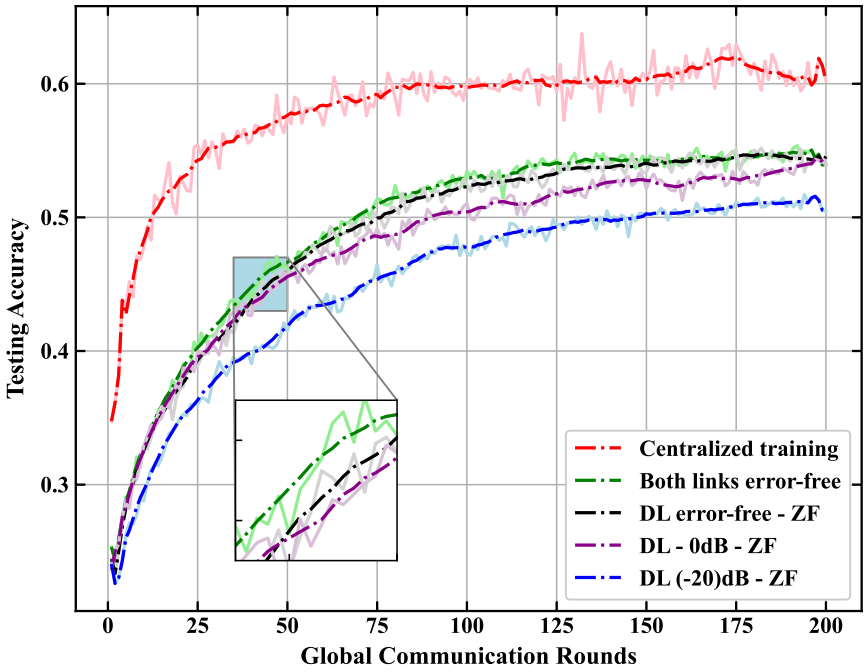


FIGURE 4.4: Averaged testing accuracy across users with CIFAR-10/STL-10 datasets with different DL SNRs and fixed UL SNR −8 dB.

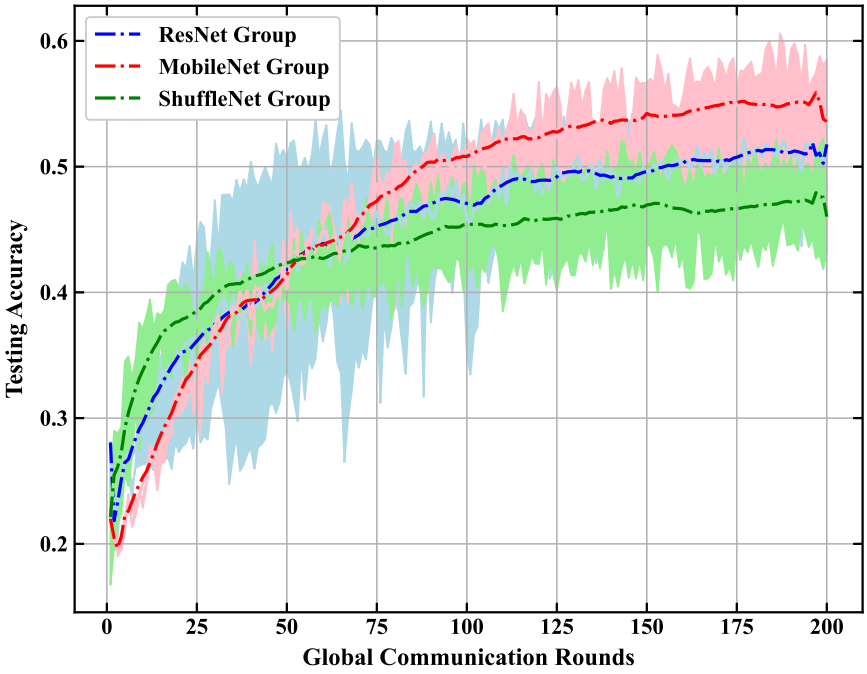


FIGURE 4.5: Testing accuracy across different users model architecture groups with CIFAR-10/STL-10 datasets with DL SNR −20 dB and UL SNR −8 dB.

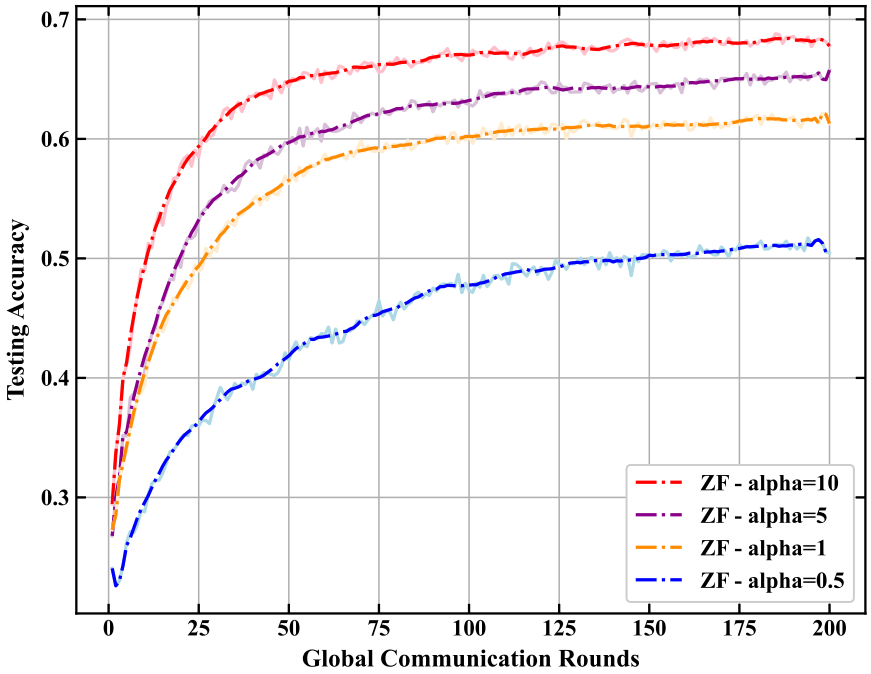


FIGURE 4.6: Averaged testing accuracy across users with CIFAR-10/STL-10 datasets for UL SNR -8 dB, DL SNR -20 dB, and different non-i.i.d.-ness $\alpha_d = 0.5, 1, 5, 10$.

4.7.1 Error-prone and error-free comparison

In Fig. 4.4, the testing accuracy with different values of DL SNR is plotted for a fixed UL SNR -8 dB. Green and black curves represent two baselines with error-free UL/DL links [92–99] and error-free DL link [101], respectively. It can be observed that for both error-free and error-prone links, the testing accuracies converge; however, with more error-prone links, the accuracy achieved is lower and takes longer to converge. With -20 dB DL SNR, the achieved accuracy is approximately 91% of the accuracy for error-free links. Red curve stands for the centralized training baseline, where a single node contains all the training samples. It can also be witnessed that, when DL SNR reaches 0 dB, the converged accuracy closely matches that of the baseline with error-free communication links. This suggests that at this SNR level, the communication is nearly fully accurate and reliable for the system under consideration, indicating that the system reaches saturation at this point.

In Fig. 4.5, we present the testing accuracies of different user groups across various model architectures. It is evident that the ResNet group exhibits a higher variance in testing accuracies compared to the MobileNet and ShuffleNet groups.

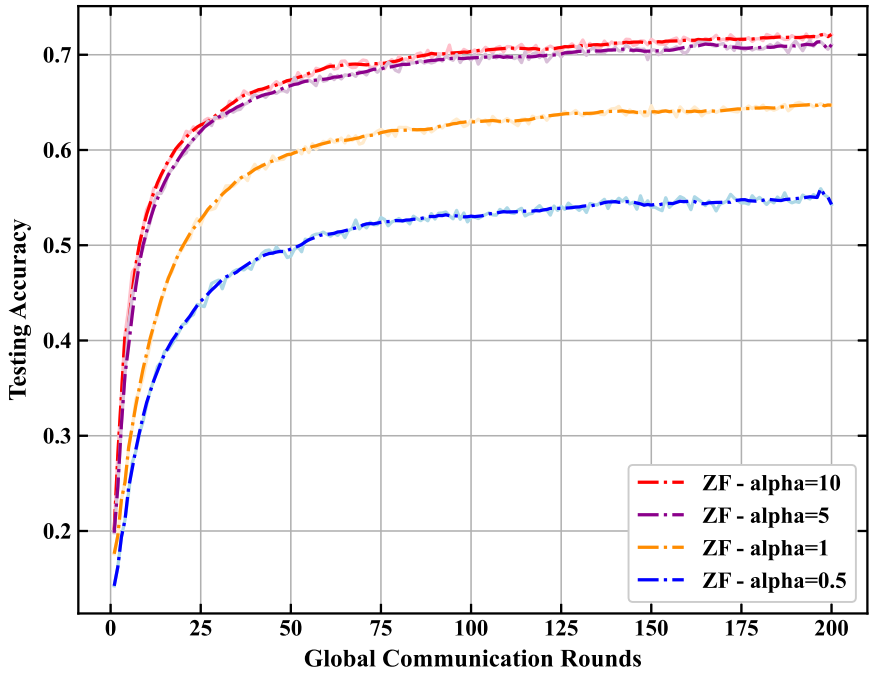


FIGURE 4.7: Averaged testing accuracy with MNIST/FMNIST datasets for UL SNR -8 dB, DL SNR -20 dB, and different non-i.i.d.-ness $\alpha_d = 0.5, 1, 5, 10$.

4.7.2 Different degrees of non-i.i.d.-ness (α_d)

In Fig. 4.6, the value of non-i.i.d. factor α_d is varied and the progress of averaged testing accuracy is plotted for UL SNR -8 dB and DL SNR -20 dB, respectively. It can be seen that the testing accuracy is higher as α_d is increased, that is, non-i.i.d.-ness is decreased. In other words, the closer samples are to i.i.d., the higher is the accuracy.

In Fig. 4.7, effects of non-i.i.d.-ness variations are presented for MNIST/FMNIST datasets for fixed UL and DL SNRs -8 and -20 dBs, respectively. Compared with the result presented in Fig. 4.6 for CIFAR-10/STL-10 datasets, with same degree of non-i.i.d.-ness, increments in the testing accuracy can be witnessed in Fig. 4.7, although in this scenario, each local user has much fewer training samples. This is due to the fact that MNIST dataset consists of single channel gray-scale image samples, whereas CIFAR-10 dataset has three-channel RGB images, making it more difficult to learn than the dataset with gray-scale images.

4.7.3 With different precoding methods

In Fig. 4.8, the averaged testing accuracy is plotted for non-i.i.d. CIFAR-10/STL-10 datasets with ZF and MMSE precoding/combining methods. It can be witnessed that, both ZF and MMSE precoding strategies achieve similar performance; however at lower

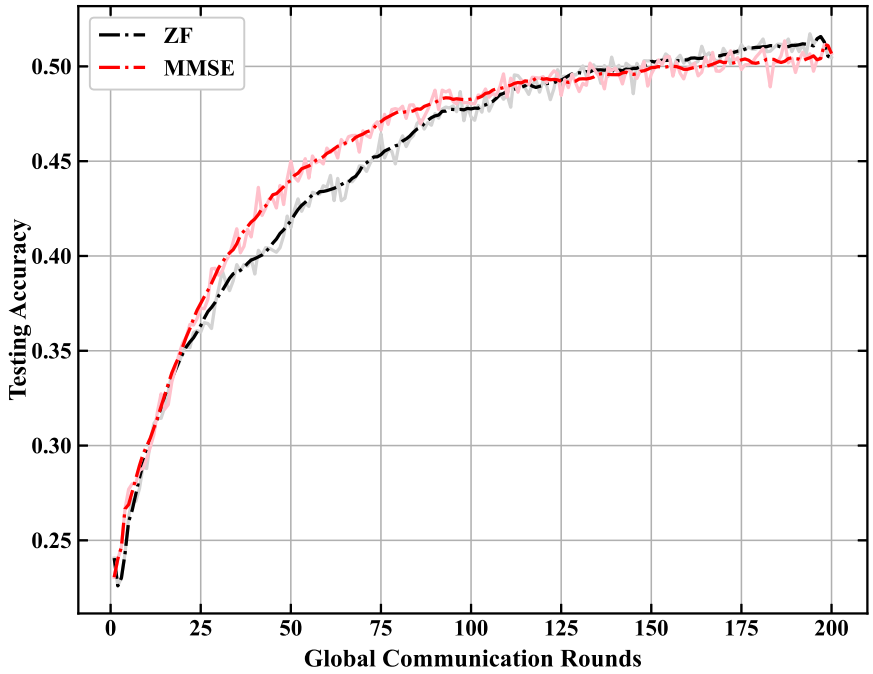


FIGURE 4.8: Averaged testing accuracy across users for different precoding methods at fixed UL-SNR −8 dB and DL-SNR −20 dB with CIFAR-10/STL-10 datasets.

rounds, MMSE provides better accuracy. In high SNR regime, this difference becomes negligible, therefore, for the rest of the simulations, we adopt the ZF strategy.

4.7.4 With training steps variations and group splitting

In Fig. 4.9, the averaged testing accuracy versus CPU running time for different variations of the FD algorithm with the same number of communication rounds is plotted. It can be seen that the converged accuracy is higher if links are error-free (green curve). The blue curve depicts the error-prone accuracy without any variations with finite DL/UL SNRs. To make the algorithm converge faster with error-prone UL/DL links, number of training steps per round is varied (TS-labeled purple curve). In this variation, the numbers of training steps at earlier rounds are higher than those in later rounds, leading to converging faster and requiring fewer communication resources. To reduce the effect of communication errors at users, two group split scheme (TG-labeled red curve, without TS) shows better testing accuracy at earlier rounds than that of the blue curve, and both incur similar running time.

4.7.5 Model heterogeneity

In Table 4.3 and 4.4, we summarize the averaged testing accuracies for user groups with nonidentical model structures when both error-free and error-prone channels (UL SNR

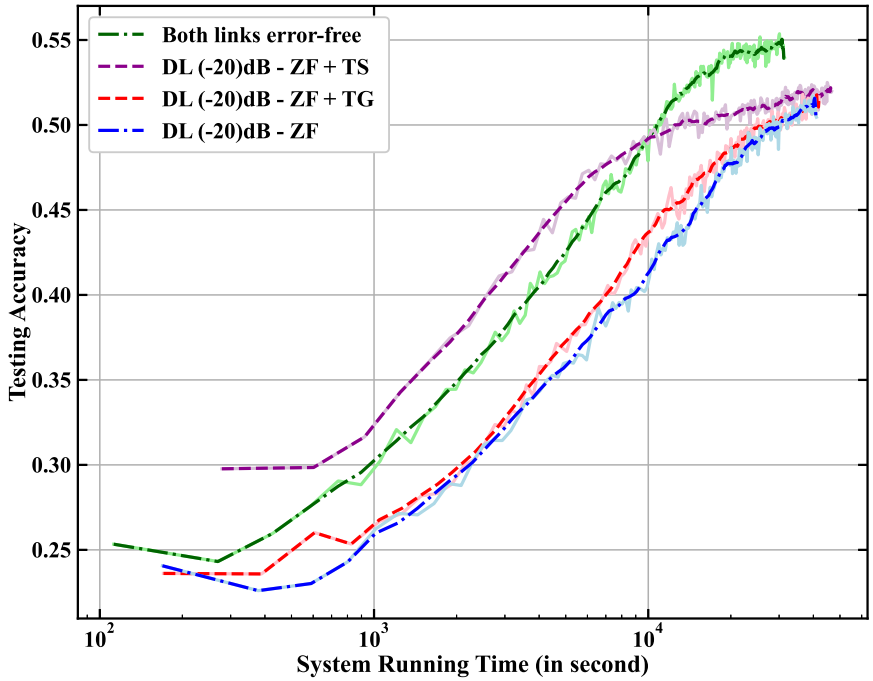


FIGURE 4.9: Averaged testing accuracy across users for CIFAR-10/STL-10 datasets with fixed UL-SNR −8 dB and DL-SNR −20 dB (TG: Two Group Split, TS: Training Steps Variations).

TABLE 4.3: Converged testing accuracies for user groups with nonidentical model architectures when error-free and error-prone channels are considered with different degrees of non-i.i.d.-ness (α_d). CIFAR-10 + STL-10 dataset combination is utilized.

		CIFAR-10+STL-10 datasets		
Settings	Model group	$\alpha_d = 0.5$	$\alpha_d = 1$	$\alpha_d = 5$
Error-free	ResNet	54.91%	65.73%	68.78%
	MobileNet	56.96%	62.73%	69.23%
	ShuffleNet	49.99%	58.85%	61.69%
	Overall	53.95%	62.44%	66.57%
Error-prone (mMIMO with ZF)	ResNet	51.77%	64.31%	66.73%
	MobileNet	53.58%	61.16%	71.06%
	ShuffleNet	46.02%	58.38%	59.47%
	Overall	50.46%	61.23%	65.76%

−8 dB and DL SNR −20 dB) are considered, and different degrees of non-i.i.d.-ness (α_d) are utilized. It is worth noting that, for the scenario with CIFAR-10/STL-10 datasets, compared with ResNet and MobileNet groups, it can be witnessed that users with the simplest architecture ShuffleNet have the most unsatisfactory performance (an approximate performance degradation of 6%, 5% and 8% when $\alpha_d = 0.5$, 1 and 5, respectively).

TABLE 4.4: Converged testing accuracies for user groups with nonidentical model architectures when error-free and error-prone channels are considered with different degrees of non-i.i.d.-ness (α_d). MNIST + FMNIST dataset combination is utilized.

		MNIST+FMNIST datasets		
Settings	Model group	$\alpha_d = 0.5$	$\alpha_d = 1$	$\alpha_d = 5$
Error-free	CNN_1	55.69%	65.45%	70.47%
	CNN_2	56.36%	64.99%	72.29%
	CNN_3	57.17%	67.38%	72.22%
	Overall	56.41%	65.94%	71.66%
Error-prone (mMIMO with ZF)	CNN_1	54.56%	63.77%	70.39%
	CNN_2	52.15%	64.18%	71.04%
	CNN_3	56.14%	66.18%	71.75%
	Overall	54.28%	64.71%	71.06%

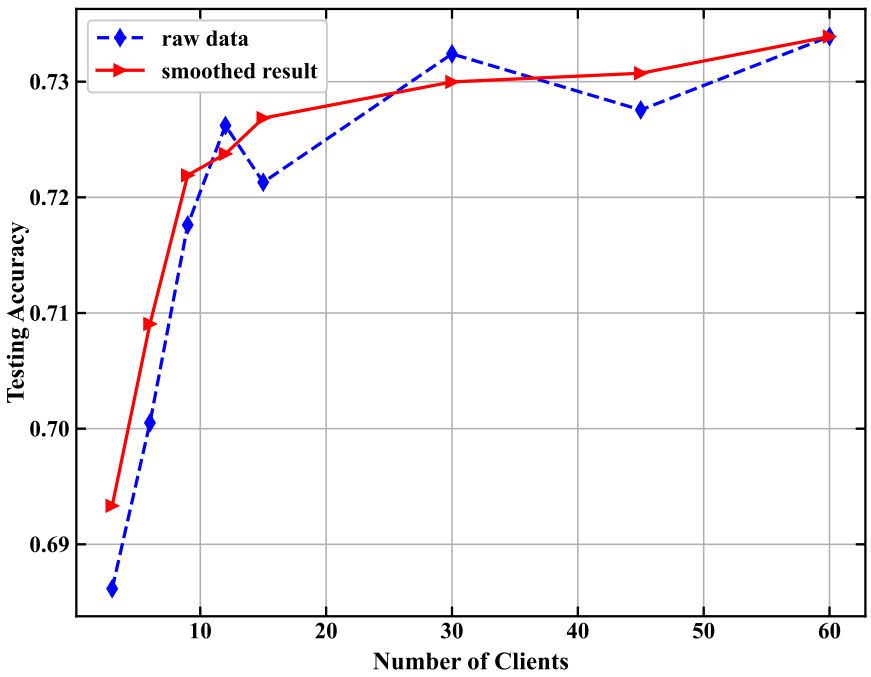


FIGURE 4.10: Converged testing accuracy with MNIST/FMNIST datasets when different numbers of clients are considered. UL SNR -8 dB, DL SNR -20 dB, $\alpha_d = 10$.

This implies that achieving successful convergence of the FD algorithm requires the simultaneous consideration of both data heterogeneity and model complexity.

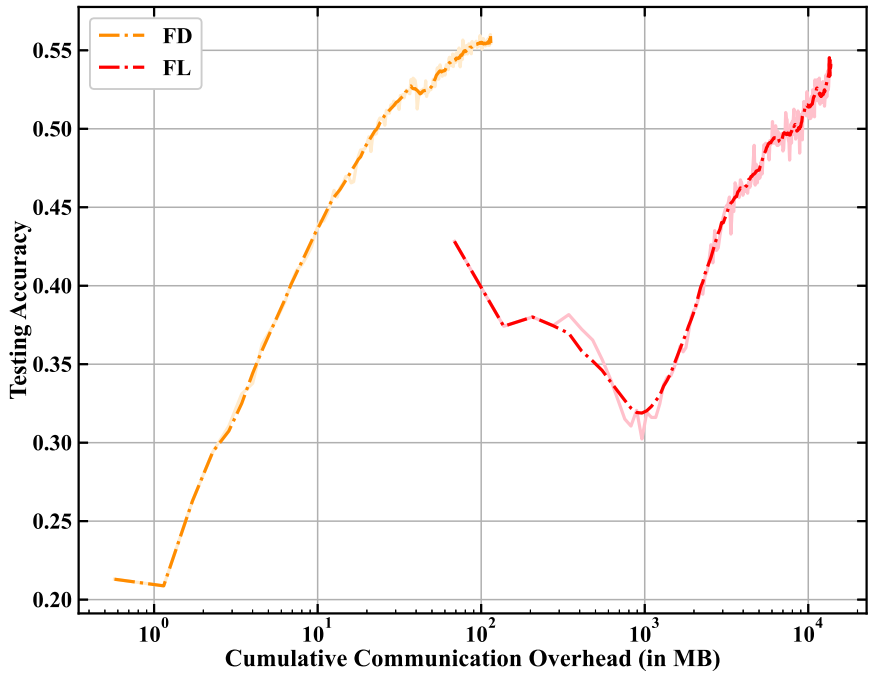


FIGURE 4.11: Averaged testing accuracy versus the cumulative overhead across communication rounds for FD and FL algorithms with fixed UL SNR −8 dB, DL SNR 3 dB and with MNIST/FMNIST datasets

4.7.6 With different numbers of clients

In Fig. 4.10, the averaged testing accuracies are plotted with respect to different numbers of clients. MNIST is used as local/private dataset and FMNIST is used for distillation. The smoothed moving average result with a window size of 3 is plotted in red. It can be seen that with the increase of the number of clients, the converged testing accuracy shows an increasing trend. However, such increment becomes smaller after $N = 15$, this is because in the massive MIMO communication system with the same number of antennas at BS and higher number of users, UL and DL SNRs decrease and consequently, the bit error rate (BER) will increase, leading to more communication errors in the received logits. Therefore, the gain from increasing the number of clients becomes negligible; that is, the system becomes saturated.

4.7.7 Communication overhead comparison between FD and FL

In Fig. 4.11, the testing accuracies of massive MIMO communication system-integrated FL [24] and FD algorithms are compared in terms of communication overhead for fixed UL SNR −8 dB and DL SNR 3 dB with MNIST/FMNIST datasets. In the FL algorithm, each of 15 users has the same CNN model with 1,199,882 learnable weights and 16-bit quantization is used for communication. In the FD algorithm, with 8-bit quantization, the number of logit values is 2000×10 where 10 is the number of classes. For

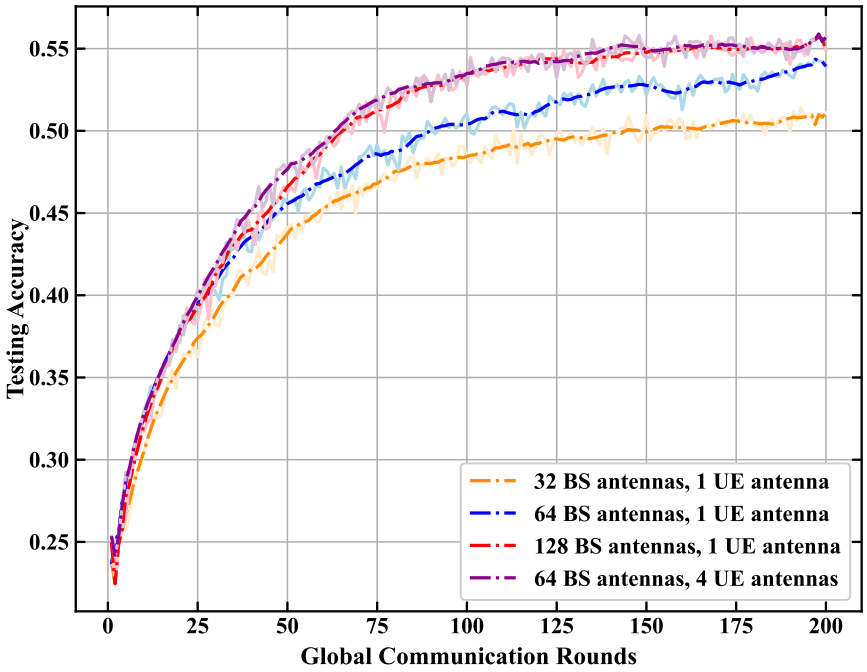


FIGURE 4.12: Averaged testing accuracy across users for different numbers of BS/UE antennas at fixed UL-SNR −8 dB and DL-SNR 0 dB with CIFAR-10/STL-10 datasets.

fair comparison, users in both frameworks are assumed to have the same CNN model architecture. Therefore, with much less communication overhead in FD, the plot for FD converges much earlier than that of FL. The variations in the FL plot are at earlier rounds, which occur due to random initializations and local minimia of weights. At the convergence, it can be witnessed that the difference in the cumulative overhead is approximately 14 GB, that is, FD overhead is around 1% of that of FL.

4.7.8 With different numbers of BS and UE antennas

In Fig. 4.12, for CIFAR-10/STL-10 datasets, the numbers of BS and UE antennas are varied. Comparing yellow (32 BS antennas), blue (64 BS antennas), and red (128 BS antennas) curves, the performance improvement can be witnessed as the number of BS antennas is increased. As an illustration, testing accuracy experiences improvement from approximate 50% to 55% by augmenting the number of BS antennas from 32 to 128. Similarly, with the increase of UE antenna numbers, an obvious ascent in testing accuracy can be observed by comparing blue and purple curves, where UE antenna numbers are set to be 1 and 4, respectively.

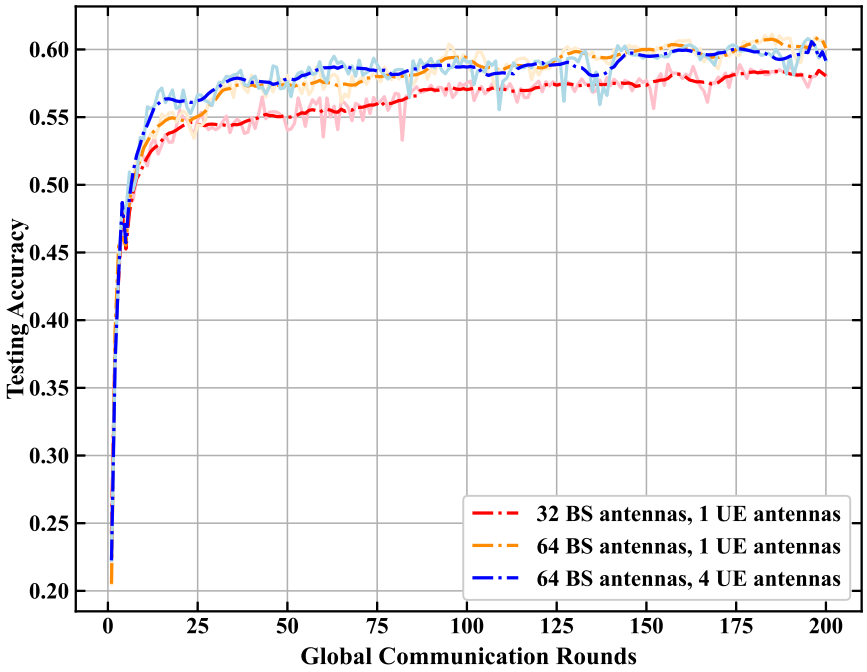


FIGURE 4.13: Averaged testing accuracy across users for different numbers of BS/UE antennas at fixed UL-SNR −8 dB and DL-SNR 0 dB with ImageNette/STL-10 datasets.

4.7.9 Simulations on ImageNette dataset

In Fig. 4.13, testing accuracies are plotted for ImageNette/STL-10 datasets with different numbers of BS/UE antennas. Due to the high-quality image in the ImageNette dataset, it can be witnessed that the model converges faster compared with CIFAR-10 or MNIST dataset scenarios.

4.8 Conclusion

In this chapter, we have integrated the federated distillation framework with multi-user massive MIMO system considering the imperfections of both UL and DL links. At the massive MIMO BS, ZF and MMSE schemes have been used to null the intra-cell interference. The convergence bound on the modeling error has been derived. By varying the number of local training steps in the proposed FedTSKD algorithm, we have improved the communication efficiency and accelerated the training process. Further, we have proposed a group-split scheme (FedTSKD-G) to let the model learn the imperfections introduced by the communication links intrinsically at the BS. The testing accuracy has been compared with the scenario with the perfect links. Comparison between FD and FL algorithms in terms of communication overhead cost has also been included to demonstrate the significantly less overhead of FD, specifically, FD needs 1% of the FL's communication overhead.

Appendix 4.9 Proof of variance for the total effective noise

Let $W = D_p \times C$. Then, the variance of the effective noise $\sum_{n=1}^{N} \alpha_n \bar{\boldsymbol{\omega}}_{n,t}$ is obtained as

$$
\begin{aligned}
W\sigma_{\omega}^2 &= \mathbb{E}\left\|\sum_{n=1}^{N} \alpha_n \bar{\boldsymbol{\omega}}_{n,t}\right\|_2^2 = \mathbb{E}\left\|\sum_{n=1}^{N} \alpha_n \left(\bar{\boldsymbol{\omega}}_{t,av} + \bar{\boldsymbol{\omega}}_{n,t,DL}\right)\right\|_2^2 \\
&= \sum_{k,j=1}^{N} \alpha_k \alpha_j tr\mathbb{E}\left(\bar{\boldsymbol{\omega}}_{t,av} + \bar{\boldsymbol{\omega}}_{k,t,DL}\right)\left(\bar{\boldsymbol{\omega}}_{t,av}^H + \bar{\boldsymbol{\omega}}_{j,t,DL}^H\right) \\
&= \sum_{k,j=1}^{N} \alpha_k \alpha_j tr\mathbb{E}\left\{\bar{\boldsymbol{\omega}}_{t,av}\bar{\boldsymbol{\omega}}_{t,av}^H + \bar{\boldsymbol{\omega}}_{k,t,DL}\bar{\boldsymbol{\omega}}_{j,t,DL}^H\right\} \\
&= \left[\frac{\|\boldsymbol{\alpha}\|_2^2\sigma_z^2}{2}\frac{\sigma^2 N_D}{P_{UL}N_{BS}} + \sigma_D^2\right] tr\mathbf{I}_W + \sum_{n=1}^{N} \alpha_n^2 tr\mathbb{E}\left\{\bar{\boldsymbol{\omega}}_{n,t,DL}\bar{\boldsymbol{\omega}}_{n,t,DL}^H\right\} \\
&= \left[\frac{\|\boldsymbol{\alpha}\|_2^2\sigma_z^2}{2}\frac{\sigma^2 N_D}{P_{UL}N_{BS}} + \sigma_D^2\right] W + \frac{\|\boldsymbol{\alpha}\|_2^2\hat{\sigma}_z^2}{2}\frac{\sigma^2}{P_{DL}} W.
\end{aligned}
$$

Appendix 4.10 Proof of Lemma 4.1

The equations are given as follows. In (a), the fact that $L_n\left(\mathbf{w}_n^*\right) \leq L_n\left(\mathbf{w}_{n,t}\right)$ is used. The inequality in (b) is from the results of Assumptions 7 and 8.

$$
\begin{aligned}
&\mathbb{E}\left\|\mathbf{w}_{n,t+1} - \mathbf{w}_n^*\right\|_2^2 \leq \mathbb{E}\left\|\mathbf{w}_{n,t} - \eta_t \nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_{n,t}\right) - \mathbf{w}_n^*\right\|_2^2 \\
&= \mathbb{E}\left\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\right\|_2^2 + \mathbb{E}\left\|-\eta_t \nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_{n,t}\right)\right\|_2^2 - 2\Re\left(\mathbf{w}_{n,t} - \mathbf{w}_n^*\right)^T \eta_t \nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_{n,t}\right) \\
&= \mathbb{E}\left\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\right\|_2^2 + \mathbb{E}\left\|-\eta_t \nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_{n,t}\right) + \eta_t \nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_n\right)\right\|_2^2 \\
&\quad + \mathbb{E}\left\|-\eta_t \nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_n\right)\right\|_2^2 - 2\Re\mathbb{E}\left(\mathbf{w}_{n,t} - \mathbf{w}_n^*\right)^T \eta_t \nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_{n,t}\right) \\
&\quad + 2\Re\left(-\eta_t \nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_n\right)\right)^T \underbrace{\mathbb{E}\left(-\eta_t \nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_{n,t}\right) + \eta_t \nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_n\right)\right)}_{=0} \\
&= \left(1 - \eta_t\mu\right)\mathbb{E}\left\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\right\|_2^2 + \eta_t^2\mathbb{E}\left\|\nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_n\right) - \nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_{n,t}\right)\right\|_2^2 \\
&\quad + \eta_t^2\mathbb{E}\left\|\nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_n\right)\right\|_2^2 + 2\eta_t\mathbb{E}\left[L_n(\mathbf{w}_n^*) - L_n(\mathbf{w}_{n,t})\right] \\
&\overset{(a)}{\leq} \left(1 - \eta_t\mu\right)\mathbb{E}\left\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\right\|_2^2 + \eta_t^2\mathbb{E}\left\|\nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_n\right) - \nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_{n,t}\right)\right\|_2^2 \\
&\quad + \eta_t^2\mathbb{E}\left\|\nabla L_n\left(\mathbf{w}_{n,t}, \mathcal{D}_n\right)\right\|_2^2 \\
&\overset{(b)}{\leq} \left(1 - \eta_t\mu\right)\mathbb{E}\left\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\right\|_2^2 + \eta_t^2\left(\sigma_n^2 + G^2\right)
\end{aligned}
$$

Appendix 4.11 Proof of Corollary 4.2

We conduct our proof via induction as follows. Let $C_n = \left(\sigma_n^2 + G^2\right), \nu_n \geq \frac{\beta_0^2 C_n}{\beta_0 \mu - 1}, \eta_t = \frac{\beta_0}{t+\beta_1}, \beta_0 > \frac{1}{\mu}, \beta_1 > 0$. By utilizing the results from Lemma 1, for $t \geq 0$, we can get the bound for $\mathbb{E}\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\|_2^2$ as follows

$$\mathbb{E}\|\mathbf{w}_{n,t+1} - \mathbf{w}_n^*\|_2^2 \leq (1 - \eta_t \mu)\, \mathbb{E}\|\mathbf{w}_{n,t} - \mathbf{w}_n^*\|_2^2 + \eta_t^2 C_n \tag{4.28}$$

$$\leq \frac{\nu_n}{t+\beta_1} - \frac{\beta_0 \mu \nu_n}{(t+\beta_1)^2} + \frac{\beta_0^2 C_n}{(t+\beta_1)^2} \tag{4.29}$$

$$= \frac{\nu_n (t+\beta_1) - \nu_n}{(t+\beta_1)^2} + \frac{\nu_n - \beta_0 \mu \nu_n + \beta_0^2 C_n}{(t+\beta_1)^2} \tag{4.30}$$

$$\leq \frac{\nu_n}{t+1+\beta_1} + 0. \tag{4.31}$$

Appendix 4.12 Proof of Lemma 4.3

The equations are given as follows. In (a), based on the SGD, the following result is used

$$\mathbb{E}\left(\nabla Q_n(\mathbf{w}_{n,t+S+i}, \hat{\mathcal{Z}}_{n,t+S+i}) - \nabla Q_n(\mathbf{w}_{n,t+S+i}, \hat{\mathcal{Z}}_n)\right) = 0 \tag{4.32}$$

In (b), μ_q strong convexity from Assumption 6 is used. In (c), results from Assumptions 7 and 8 are utilized. The inequality in (d) comes from the fact that $Q_n\left(\mathbf{w}_n^*\right) \leq Q_n\left(\mathbf{w}_n\right)$.

$$
\begin{aligned}
\mathbb{E}\left\|\mathbf{w}_{n,t+2S}-\mathbf{w}_n^*\right\|_2^2 &= \mathbb{E}\left\|\mathbf{w}_{n,t}-\sum_{i=0}^{S-1}\eta_{t+S+i}\nabla Q_n(\mathbf{w}_{n,t+S+i},\hat{\mathcal{Z}}_{n,t+S+i})-\mathbf{w}_n^*\right\|_2^2 \\
&= \mathbb{E}\left\|\mathbf{w}_{n,t}-\mathbf{w}_n^*\right\|_2^2+\mathbb{E}\left\|\sum_{i=0}^{S-1}\eta_{t+S+i}\left(\nabla Q_n(\mathbf{w}_{n,t+S+i},\hat{\mathcal{Z}}_{n,t+S+i})\pm\nabla Q_n(\mathbf{w}_{n,t+S+i},\hat{\mathcal{Z}}_n)\right)\right\|_2^2 \\
&\quad -2\Re\mathbb{E}\sum_{i=0}^{S-1}\eta_{t+S+i}\left(\mathbf{w}_{n,t}-\mathbf{w}_n^*\right)^T\nabla Q_n(\mathbf{w}_{n,t+S+i},\hat{\mathcal{Z}}_{n,t+S+i}) \\
&\overset{(a)}{=} \mathbb{E}\left\|\mathbf{w}_{n,t}-\mathbf{w}_n^*\right\|_2^2+\mathbb{E}\left\|\sum_{i=0}^{S-1}\eta_{t+S+i}\left(\nabla Q_n(\mathbf{w}_{n,t+S+i},\hat{\mathcal{Z}}_{n,t+S+i})-\nabla Q_n(\mathbf{w}_{n,t+S+i},\hat{\mathcal{Z}}_n)\right)\right\|_2^2 \\
&\quad +\mathbb{E}\left\|\sum_{i=0}^{S-1}\eta_{t+S+i}\nabla Q_n(\mathbf{w}_{n,t+S+i},\hat{\mathcal{Z}}_n)\right\|_2^2 \\
&\quad -2\Re\sum_{i=0}^{S-1}\eta_{t+S+i}\mathbb{E}\left(\mathbf{w}_{n,t}-\mathbf{w}_n^*\right)^T\nabla Q_n(\mathbf{w}_{n,t+S+i},\hat{\mathcal{Z}}_{n,t+S+i}) \\
&\overset{(b)}{\leq} \mathbb{E}\left\|\mathbf{w}_{n,t}-\mathbf{w}_n^*\right\|_2^2+\left(\sum_{i=0}^{S-1}\eta_{t+S+i}^2\right)S\sigma_{qn}^2+\mathbb{E}\left\|\sum_{i=0}^{S-1}\eta_{t+S+i}\nabla Q_n(\mathbf{w}_{n,t+S+i},\hat{\mathcal{Z}}_n)\right\|_2^2 \\
&\quad +2\Re\sum_{i=0}^{S-1}\eta_{t+S+i}\left[\mathbb{E}\left\{Q_n(\mathbf{w}_n^*,\hat{\mathcal{Z}}_{n,t+S+i})-Q_n(\mathbf{w}_{n,t},\hat{\mathcal{Z}}_{n,t+S+i})\right\}-\frac{\mu_q}{2}\mathbb{E}\left\|\mathbf{w}_{n,t}-\mathbf{w}_n^*\right\|_2^2\right], \\
&\overset{(c)}{\leq} \left(1-\mu_q\sum_{i=0}^{S-1}\eta_{t+S+i}\right)\mathbb{E}\left\|\mathbf{w}_{n,t}-\mathbf{w}_n^*\right\|_2^2+\left(\sum_{i=0}^{S-1}\eta_{t+S+i}^2\right)\left(S\sigma_{qn}^2+SG_q^2\right) \\
&\quad +2\Re\sum_{i=0}^{S-1}\eta_{t+S+i}\underbrace{\left[\mathbb{E}\left\{Q_n(\mathbf{w}_n^*,\hat{\mathcal{Z}}_{n,t+S+i})-Q_n(\mathbf{w}_{n,t},\hat{\mathcal{Z}}_{n,t+S+i})\right\}\right]}_{Q_n(\mathbf{w}_n^*)-Q_n(\mathbf{w})\leq 0} \\
&\overset{(d)}{\leq} \left(1-\mu_q\sum_{i=0}^{S-1}\eta_{t+S+i}\right)\mathbb{E}\left\|\mathbf{w}_{n,t}-\mathbf{w}_n^*\right\|_2^2+\left(\sum_{i=0}^{S-1}\eta_{t+S+i}^2\right)S\left(\sigma_{qn}^2+G_q^2\right)
\end{aligned}
$$

Appendix 4.13 Proof of Corollary 4.4

Similar to Corollary 4.2, we conduct our proof via induction as follows. Let $C_{qn}=\left(\sigma_{qn}^2+G_q^2\right), \nu_n \geq \frac{S^2\beta_0^2C_{qn}}{\beta_0\mu_q-2S}, \eta_t=\frac{\beta_0}{t+\beta_1}$, where $\beta_0>\frac{2S}{\mu_q}, \beta_1\geq 4$. By utilizing the results from Lemma 4.3, for $t\geq 0$, we write

$$
\begin{aligned}
\mathbb{E}\|\mathbf{w}_{n,t+2S}-\mathbf{w}_n^*\|_2^2 &\leq \left(1-\mu_q\sum_{i=0}^{S-1}\eta_{t+S+i}\right)\mathbb{E}\|\mathbf{w}_{n,t}-\mathbf{w}_n^*\|_2^2+\left(\sum_{i=0}^{S-1}\eta_{t+S+i}^2\right)SC_{qn}\\
&\leq \left(1-\mu_q\sum_{i=0}^{S-1}\eta_{t+S+i}\right)\frac{\nu_n}{t+\beta_1}+\left(\sum_{i=0}^{S-1}\eta_{t+S+i}^2\right)SC_{qn}\\
&\leq \frac{\nu_n}{t+\beta_1}-\frac{\nu_n}{t+\beta_1}\mu_q\sum_{i=0}^{S-1}\eta_{t+S+i}+\left(\sum_{i=0}^{S-1}\eta_{t+S+i}^2\right)SC_{qn}\\
&\overset{(a)}{\leq} \frac{\nu_n}{t+\beta_1}\times\frac{t+\beta_1}{t+\beta_1}-\frac{\nu_n}{t+\beta_1}\frac{\beta_0}{t+\beta_1}\mu_q+\frac{S\beta_0^2}{(t+\beta_1)^2}SC_{qn}\\
&= \frac{\nu_n(t+\beta_1)-2S\nu_n}{(t+\beta_1)^2}+\frac{2S\nu_n}{(t+\beta_1)^2}-\frac{\nu_n}{t+\beta_1}\frac{\beta_0}{t+\beta_1}\mu_q+\frac{S\beta_0^2}{(t+\beta_1)^2}SC_{qn}\\
&\leq \frac{\nu_n}{t+2S+\beta_1}+\frac{2S\nu_n-\nu_n\beta_0\mu_q+S^2\beta_0^2C_{qn}}{(t+\beta_1)^2}\\
&\leq \frac{\nu_n}{t+2S+\beta_1}+0,
\end{aligned}
$$

where in (a), the following results are used

$$
\begin{aligned}
\sum_{i=0}^{S-1}\eta_{t+S+i}&=\sum_{i=0}^{S-1}\frac{\beta_0}{t+S+i+\beta_1}\overset{(b)}{>}\frac{\beta_0}{t+\beta_1}\\
\sum_{i=0}^{S-1}\eta_{t+S+i}^2&=\sum_{i=0}^{S-1}\frac{\beta_0^2}{(t+S+i+\beta_1)^2}<\frac{S\beta_0^2}{(t+\beta_1)^2},
\end{aligned}
$$

in inequality (b), the condition that $\beta_1\geq 4$ is used.

Appendix 4.14 Proof of Theorem 4.5

With the ψ-smoothness in Assumption 5, we write

$$
\begin{aligned}
&\mathbb{E}\left\{L_n\left(\mathbf{w}_{n,t+1}\right)\right\}-L_n^*\\
&\leq \mathbb{E}\left\{\left(\mathbf{w}_{n,t+1}-\mathbf{w}_n^*\right)^T\nabla L_n(\mathbf{w}_n^*)\right\}+\frac{\psi}{2}\mathbb{E}\|\mathbf{w}_{n,t+1}-\mathbf{w}_n^*\|_2^2\\
&\overset{(a)}{=}\frac{\psi}{2}\mathbb{E}\|\mathbf{w}_{n,t+1}-\mathbf{w}_n^*\|_2^2\\
&\overset{(b)}{\leq}\frac{\psi}{2}\cdot\frac{\nu_n}{t+1+\beta_1},
\end{aligned}
$$

where in (a), zero mean result $\mathbb{E}\left\{\mathbf{w}_{n,t}-\mathbf{w}_n^*\right\}=\mathbf{0}$ is used. In (b), the inequality, $\mathbb{E}\|\mathbf{w}_{n,t}-\mathbf{w}_n^*\|_2^2\leq\frac{\nu_n}{t+\beta_1}$ is from Corollary 4.2 and Corollary 4.4, where $\nu_n\geq\max\left(\frac{\beta_0^2C_n}{\beta_0\mu-1},\frac{S^2\beta_0^2C_{qn}}{\beta_0\mu_q-2S}\right)$, $C_{qn}=\left(\sigma_{qn}^2+G_q^2\right),C_n=\left(\sigma_n^2+G^2\right),\eta_t=\frac{\beta_0}{t+\beta_1},\beta_0>\max\left(\frac{1}{\mu},\frac{2S}{\mu_q}\right),\beta_1\geq 4$.

Chapter 5

Adaptive Model Pruning in Distributed Federated Learning

5.1 Introduction

In the previous chapters, we have discussed the FL algorithm as well as its FD variant. For the proper working of these algorithms, a BS is required to aggregate the received model weights from users, coordinate the training process, and make training-related decisions. We demonstrated how the distributed nature provided by these algorithms is able to address the challenges provided by traditional "one-node learning", such as potential privacy leakage, huge communication overhead, and slow convergence. In this way, FL is able to harness collective intelligence while respecting data ownership and privacy [27], and such capability renders it particularly appealing for real-world scenarios where traditional approaches fall short [121], such as healthcare [122], smart home [123] and industrial engineering [124].

Despite the effectiveness of the aforementioned star-topology FL paradigm, where a central server is included, in wireless communications, there are many important scenarios where no BS is available due to the time-varying characteristic of network topology. In parallel, the advent of Device-to-Device (D2D) communication networks [125] has triggered an evolution in how devices interact and exchange information. In D2D communications, instead of relaying data via a central BS, communication links are directly established between proximate devices for direct data sharing and transmission, thus enhancing the scalability of network and offering an intriguing framework to address the challenges of traditional star-topology approaches [126].

Work in this chapter has been accepted at IEEE SPAWC 2024 [120] and submitted to IEEE Transactions on Vehicular Technology 2024.

At the same time, in FL, when the size of local model is very large, the communication and computation latency will be very high, especially when the local device has limited computing capability and wireless resources. To solve the problem, model pruning is considered. The model pruning technique enables the deep neural network (DNN) to be represented in a more simplified and compact architecture such that total communication overhead during the FL training process is reduced, rendering it a promising research topic in many interdisciplinary fields.

The rest of this chapter is organized as follows. A summary of the related work is given in Section 5.2. Section 5.3 details the system model and problem formulation. Convergence analysis and problem transformation are provided in Section 5.4. Section 5.5 presents the optimization of pruning ratio and wireless bandwidth allocation. Simulation results are presented in Section 5.6. Finally, Section 5.7 concludes this chapter.

5.2 Related Work

5.2.1 FL in wireless communication systems

As explained in Chapter 3 and demonstrated in many works [24, 39, 46, 59], the advanced signal processing and communication techniques can be utilized and applied to speed up the convergence of FL model and help to train a global model with better performance accounting the reliability of wireless links. Meanwhile, the privacy-preserving nature of FL paradigm has been recognized as one of the most promising solutions to achieve ubiquitous AI [2] in future wireless communication, such as beyond fifth-generation (5G) and sixth-generation (6G) systems [32, 35, 36]; that is, wireless communication and FL can benefit each other mutually. Specifically, research trend in this direction to design and optimize star-topology FL algorithm in the wireless system mainly falls into four categories, quantization schemes [41, 42], imperfect communication links for weights transmission [57, 60], resource allocation [45, 47], and scheduling policy [73, 127].

5.2.2 Fully distributed FL

On the other hand, serverless federated learning, i.e., fully distributed FL, has progressively emerged as another prominent area of research due to its flexibility and scalability over conventional star-topology FL. In [128], for a fully distributed FL system, it was assumed that communication links were established between any pair of local devices. An approximation of the global gradient was computed by collecting the gradients of neighbor devices, and the divergence between the local and global gradients was measured and evaluated via the inner product. Subsequently, based on the evaluation, an adaptive mechanism for weight aggregation was introduced. Tao *et al.* considered an undirected

graph for network topology in [129] and each device employed stochastic gradient descent (SGD) with momentum for local training. Corresponding convergence analysis was also provided. Similarly, in [130], the considered wireless network was characterized by an undirected graph, and an event-triggering scheduling policy was proposed to deal with the straggler problem caused by the heterogeneity among local devices. In [131], for a decentralized federated learning system in wireless D2D networks, a graph-coloring-based scheduling policy was proposed to guarantee interference-free communication at a common neighbor device. Both digital and analog implementations were analyzed in this work. The convergence bound for the proposed algorithm in [131] was conducted by the authors in [132]. In [133], the impact of D2D transmission failures was considered as a factor in assessing the efficacy of the decentralized FL algorithm and physical layer resources, such as data rate and bandwidth allocation, were jointly optimized under a given latency condition.

5.2.3 Clustered FL

In addition, there are some works focusing on clustered federated learning (CFL). In [134], devices are partitioned into overlapping clusters. In this way, the local information of local devices can be shared among multiple clusters simultaneously and utilized efficiently. The work in [135] proposed a clustering algorithm based on the geometric property of the global loss function to deal with the multi-task learning problem under privacy constraints. An iterative federated clustering algorithm was proposed in [136], where the central server maintained multiple global models, with devices grouped into clusters. The assignment of devices to specific global models was determined through an allocation process predicated on the statistical distribution characteristics of their respective local training datasets. In [137], Christopher *et al.* suggested employing a hierarchical clustering mechanism to categorize devices according to the resemblance between their local updates and the global joint model. Although the above-mentioned clustering algorithms could be viewed as postprocessing methods of conventional FL and no modifications to the communication protocol were required, they neglected the dynamic movement of devices inside the communication system during the training process.

5.2.4 Model pruning-enabled FL

The research on investigating the integration of model pruning technique and FL systems, in contrast, still remains quite limited. In [138], Saeed *et al.* investigated a personalized FL algorithm under a data heterogeneity setting. For each device, a small subnetwork was established at each local iteration via model pruning and server-side aggregation was performed on the sub-models to guarantee the generalization of the

global model. Pavana *et al.* [139] proposed a compression-based FL algorithm, where the model pruning technique was utilized before the local training process to enable efficient model aggregation such that power-constrained IoT devices can participate effectively in the FL process. However, in this work, no communication factors were considered, such as channel fading and noise. In [140], for a star-topology FL system, an adaptive model pruning algorithm was proposed to adjust the local model size and reduce communication overhead. But still, the practical communication process was ignored.

5.2.5 Contributions

In this chapter, we investigate a fully distributed federated learning framework with dynamic model pruning in a wireless D2D communication network. In each global communication round, K-means clustering algorithm is employed to group devices based on their geographical positions, and devices only communicate with their neighbors via fading channels in the same group to share model weights. Meanwhile, model pruning technique is considered to decrease computation and communication latency while guaranteeing learning accuracy. Furthermore, we mathematically model the computation and communication latency and analyze the convergence of the proposed FL framework. Finally, KKT conditions are deployed to derive the optimal solutions for the pruning ratio and bandwidth allocation under latency and bandwidth thresholds. The main contributions of this chapter are summarized as follows.

- We propose and implement a fully distributed federated learning system in D2D networks. Adaptive model pruning is employed to accelerate the training process, reduce communication overhead, and accommodate devices with limited computing capability.
- For the proposed system, we mathematically model the computation and communication latency and conduct the convergence analysis, accounting for the D2D communication and model pruning processes. The analysis shows that the modeling error of the proposed FL algorithm reduces in $\mathcal{O}\left(t_{\tau}^{-1}\right)$ manner. Then, we jointly optimize the pruning ratio and bandwidth allocation to maximize the convergence rate while ensuring learning accuracy.
- Given the constraints of latency and bandwidth thresholds, we divide the optimization problem into two sub-problems. Leveraging Karush-Kuhn-Tucker (KKT) conditions, these sub-problems are resolved to derive optimal solutions for both the pruning ratio and bandwidth allocation.
- The simulation results demonstrate that the proposed FL algorithm necessitates only 88% of the training time required by the system employing equal pruning and

80% of the training time needed by the system without pruning, while achieving a testing accuracy comparable to that of the latter.

It is important to note that, while the proposed system is based on established techniques, the key innovations in our work stem from the unique integration and optimization of these methods to address the specific challenges of fully distributed FL over wireless networks. Our approach requires the joint consideration of wireless communication impairments, dynamic network topology, FL training procedures, and the model pruning process, all within a carefully formulated problem framework.

5.3 System Model

We consider a fully distributed federated learning system over wireless D2D networks. The network has N devices. Each device is allocated to a specific cluster according to its geographical position in each global communication round by using the K-means clustering algorithm. Each device only communicates with other devices within the same cluster. The neighbor set of device n at the r^{th} global communication round is denoted by $\mathcal{V}_r^n$. In particular, a logical control node [130, 141] exists for global scheduling. It should be noted that this control node is entirely different from the parameter server responsible for local model aggregation. Its primary responsibility is to collect global information within the system, such as wireless channel conditions and device locations, and to make decisions regarding network topology construction in each global round.

5.3.1 Fully distributed federated learning process

In the considered system, the n^{th} device has a localized training dataset denoted as $\mathcal{D}_n = \{\mathbf{d}_i\}_{i=1}^{D_n}$, where D_n represents its dataset size. Each data point $\mathbf{d}$ within $\mathcal{D}_n$ is structured as an input-output pair, $\{\mathbf{x}, y\}$, where $\mathbf{x}$ signifies the input to the model, and y signifies the corresponding output. In this way, the total dataset across devices is denoted by $\mathcal{D} = \cup_{n=1}^{N} \mathcal{D}_n$ with size $D = \sum_{n=1}^{N} D_n$. The training optimization problem is to find the optimal model parameters $\mathbf{w}^*$ to minimize the global loss function, denoted as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} F(\mathbf{w}), \tag{5.1}$$

where $F(\mathbf{w}) = \sum_{n=1}^{N} \alpha_n F_n(\mathbf{w})$ denotes the global loss function and $\alpha_n = {}^{D_n}/_{D}$ is the weighting factor of the n^{th} device such that $\alpha_n \geq 0, \forall k$ and $\sum_{n=1}^{N} \alpha_n = 1$. $F_n(\mathbf{w})$ is the local loss function for n^{th} device, and it is given as

$$F_n(\mathbf{w}) \coloneqq \frac{1}{D_n} \sum_{\mathbf{d} \in \mathcal{D}_n} f(\mathbf{w}, \mathbf{d}). \tag{5.2}$$

In Eqn. (5.2), $f(\mathbf{w}, \mathbf{d})$ represents the loss function, which quantifies the performance of the model parameterized by $\mathbf{w}$, given the labeled data point $\mathbf{d}$.

Building on the notations used in Chapter 3, we use S to denote the number of steps in one communication round, and let $\mathcal{T} = \left\{S_1, S_1 + S_2, \ldots, \sum_{r=1}^{R} S_r\right\}$ represent the time instances for each *global* round. In each global round, these time instances encapsulate events including model pruning, D2D communication, model aggregation, and device relocation. Consequently, the instances for local updates are represented as $\mathcal{S} = \left\{0, 1, 2, \ldots, S_1, \ldots, S_1 + S_2, \ldots, \sum_{r=1}^{R} S_r\right\} \setminus \mathcal{T}$, where R is the number of total global communication rounds. To this end, the major procedures of the proposed distributed FL algorithm for a typical (say t^{th}) communication round are described as follows.

5.3.1.1 Device movement and clustering

At the start of the t^{th} round, each device will undergo a relocation of its spatial coordinates, and the K-means algorithm is employed to group devices based on their updated geographical proximity. Most modern smartphones, tablets, and IoT devices are equipped with Global Positioning System (GPS) modules as a standard feature. Given that our proposed system involves mobile devices, it is practical to assume that these devices have the ability to provide their geographical location via GPS.

K-means [142] is an unsupervised learning algorithm that categorizes data points into distinct clusters based on their proximity to centroids. Through an iterative optimization process, the algorithm progressively improves cluster assignments, ultimately converging towards a configuration that mitigates the variance within the individual clusters, which is expressed as

$$\min \sum_{i=1}^{K} \sum_{\mathbf{z} \in \mathcal{C}_i} \|\mathbf{z} - \boldsymbol{\mu}_i\|^2, \tag{5.3}$$

where $\mathcal{C}_i$ and $\boldsymbol{\mu}_i$ are the i^{th} cluster and its corresponding centroid, $\mathbf{z}$ is the data point and $\boldsymbol{\mu}_i$ is updated in each round as

$$\boldsymbol{\mu}_i = \frac{1}{|\mathcal{C}_i|} \sum_{\mathbf{z} \in \mathcal{C}_i} \mathbf{z}. \tag{5.4}$$

For the proposed system in this chapter, the input to the clustering algorithm is the geographical location of devices, and the number of clusters K should be significantly smaller than the number of devices N.

5.3.1.2 Local training

Following the steps of device movement and clustering, the n^{th} device initializes the local weights by the aggregated model from the $(t-1)^{th}$ round and performs local stochastic gradient descent (SGD) steps using its local dataset as

$$\mathbf{w}_{t+1}^n = \mathbf{w}_t^n - \eta_t \nabla F_n(\mathbf{w}_t^n, \xi_t^n), t \in \mathcal{S}, \tag{5.5}$$

where η_t denotes the learning rate; ξ_t^n signifies a mini-batch, which is a subset chosen uniformly and independently from the n^{th} device's local training dataset $\mathcal{D}_n$. At the end of S steps, the locally trained model, $\mathbf{w}_t^n$, is evaluated on the local testing dataset and the averaged testing accuracy over all devices is defined as the local testing accuracy for the t^{th} round.

With a slight abuse of notation, we adopt floating point operations (FLOP) f to evaluate the complexity of the local neural network [130]. For the n^{th} device, let E_n denote the number of training epochs in one global round and D_n denote training samples. Moreover, let f_n represent the computing capacity of the n^{th} device in terms of FLOPs (floating point operations per second). To this end, the latency incurred by local training for one global round on the n^{th} device is [1]

$$l_{n,t}^{train} = \frac{f D_n E_n}{f_n}, \tag{5.6}$$

5.3.1.3 D2D communication

Each device broadcasts its trained weights to its neighbors inside the assigned cluster. We consider orthogonal frequency-division multiple access (OFDMA) as the communication protocol for D2D transmission [72, 130, 143] to ensure that no interference exists between each communication link. Then, at time instance t, the received signal at device n from device j, denoted by y_t^{nj}, is expressed as

$$y_t^{nj} = \sqrt{A_0} \left(\frac{d_{nj}}{d_0} \right)^{-\frac{\gamma}{2}} h_t^{nj} x_t^j + n_t^n, \tag{5.7}$$

where $h_t^{nj} \sim \mathcal{CN}(0,1)$ is the Rayleigh fading channel coefficient between device n and j. $n_t^n \sim \mathcal{CN}\left(0, \sigma^2\right)$ stands for the additive white Gaussian noise (AWGN) at device n. A_0 is the average channel power gain at reference distance d_0 and d_{nj} denotes the distance between device n and j; γ is the path loss exponent factor. Then, the throughput

[1] For the consistency of notation, we still use the subscript of n here. In practice, we consider the j^{th} neighbors of device n such that $j \in \mathcal{V}_t^n$, as will be seen later in Eqn. (5.13).

between device n and j is given by

$$R_{nj}^{t} = a_{nj}^{t} B \log_2 \left(1 + \frac{p_{nj} \left| h_{nj,t}^{'} \right|^2}{\sigma^2} \right), \tag{5.8}$$

where $h_{nj,t}^{'} = \sqrt{A_0} \left(\frac{d_{nj}}{d_0} \right)^{-\frac{\gamma}{2}} h_t^{nj}$ denotes the channel gain between device n and j at $t \in \mathcal{T}$. a_{nj}^{t} denotes the bandwidth fraction allocated to the link between device n and j, and B represents the bandwidth of the wireless channel, To this end, the communication latency between device n and j is written as

$$l_{n,j,t}^{comm} = \frac{q_j^t W}{R_{nj}^t}, \tag{5.9}$$

where q_j^t is the number of quantization bits, W represents the number of model weights.

5.3.1.4 Model aggregation

After receiving the weight transmitted by neighboring nodes within the cluster, each device proceeds to update its local weights by aggregating these noisy weight values. Let $\hat{\mathbf{w}}_t^{j,p}$ denote the received model weights from device j and $|\mathcal{V}_t^n|$ denotes the cardinality of the set $\mathcal{V}_t^n$, then, the model aggregation process is written as

$$\hat{\mathbf{w}}_t^n = \frac{1}{|\mathcal{V}_t^n|} \sum_{j \in \mathcal{V}_t^n} \hat{\mathbf{w}}_t^j. \tag{5.10}$$

For the n^{th} device, the latency incurred by the model aggregation process is calculated as

$$l_{n,t}^{aggre} = \frac{\psi W}{f_n}, \tag{5.11}$$

where $\psi > 0$ is the normalizing factor for characterizing the model aggregation process. Therefore, the local computation latency, denoted by l_n^{comp}, is calculated as

$$l_{n,t}^{comp} = l_{n,t}^{aggre} + l_{n,t}^{train} = \frac{\psi W}{f_n} + \frac{f D_n E_n}{f_n}, \tag{5.12}$$

5.3.2 Overall computation and communication latency

As discussed above, consider both local computation and model weights transmission latency, for a specific device j, the overall latency is calculated as

$$l_{j,t}^{total} = l_{j,t}^{comp} + l_{n,j,t}^{comm} = \frac{\psi W}{f_j} + \frac{f D_j E_j}{f_j} + \frac{q_j^t W}{R_{nj}^t}. \tag{5.13}$$

For the n^{th} device, to be simplicity, synchronous transmission of the weights from its neighbors in the assigned cluster is considered. To this end, its overall related latency is bottlenecked by its slowest neighbor that completes the local model aggregation, local model training, and weights transmission, that is

$$l_{n,t} = \max_{j \in \mathcal{V}_t^n} \{l_{j,t}^{total}\}. \tag{5.14}$$

5.3.3 Optimization problem formulation

Given the abovementioned system model, the objective of this work is to minimize the global loss function $F(\mathbf{w})$, under the thresholds of bandwidth and latency. The optimization problem is formulated as

$$\min_{a_{nj}^t} F(\mathbf{w}), \tag{5.15}$$

subject to

$$l_{n,t} \leq l_{th}, \tag{5.16}$$

$$\sum_{j=1}^{|\mathcal{V}_t^n|} a_{nj}^t \leq 1, \tag{5.17}$$

$$0 \leq a_{nj}^t \leq 1, \quad j \in \mathcal{V}_t^n. \tag{5.18}$$

l_{th} in Eqn. (5.16) denotes the threshold for the overall latency of local computation and weights transmission. Constraints in Eqn. (5.17) and Eqn. (5.18) represent the bandwidth allocation thresholds. In the following section, we first consider post-training model pruning to decrease the local computation and weight transmission latency. Then, the convergence bound of the proposed system is proved. Finally, based on the derived convergence analysis, the optimization problem in Eqn. (5.15) is transformed into optimizing the convergence bound.

5.4 Convergence Analysis and Problem Transformation

In this section, we start by elaborating on the concepts of model pruning, identifying its potential to decrease the latency in the proposed system. Then, we conduct the convergence analysis for the proposed FL framework and the original optimization problem shown in the previous section is transformed into a joint optimization of pruning ratio and bandwidth allocation.

5.4.1 Model pruning

In the proposed FL system, the process of updating learning models on devices with restricted computational resources and transmitting them between neighbors with sub-optimal wireless channel conditions could result in high computation and communication latency. Therefore, following the process of local training, we consider a post-training model pruning process on each device's local trained model to eliminate non-critical parameters. The goal of neural network pruning is to identify and remove redundant connections to reduce the size of the network without sacrificing model performance. In a pruning algorithm, each model parameter in the network is assigned a score, which indicates and ranks the importance of the corresponding neuron connection.

In this work, we exploit the iterative pruning strategy [144], where the pruning process is carried out iteratively over multiple rounds or epochs. During each iteration, a portion of connections with low importance is pruned and scores are reevaluated after the removal of these connections. Such iterations are repeated until the desired level of sparsity is achieved. In this way, the model is gradually refined and the issue caused by layer collapse can be alleviated. We consider two different pruning criteria, magnitude score [145] and synaptic flow conservation score [146] to evaluate the pruning performance. For the n^{th} device's locally trained model $\mathbf{w}_t^n$, we denote its corresponding pruned model as $\mathbf{w}_t^{n,p}$. We also define the pruning ratio or sparsity ρ_n as the percentage of the weights in the model with a value of zero. That is,

$$\rho_n = \frac{W_n^p}{W}, \tag{5.19}$$

where W_n^p is the number of zero-valued parameters in the pruned model and W denotes the overall size of the local model. Increasing ρ_n yields a more compact model, thereby reducing communication overhead. However, it may compromise model performance.

For the iterative pruning algorithm, during each iteration, a binary mask is employed to nullify weights characterized by the lowest scores. Taking the iterative synaptic flow pruning [146] as an example, the pruning process can be summarized in Algorithm 4.

Accounting for this post-training model pruning process in mind, the proposed FL framework is shown in Fig. 5.1 and the computation and communication latency discussed in the previous section is updated as follows:

5.4.1.1 Local computation latency

For the n^{th} device, given the pruning ratio ρ_n^t, in each global communication round, $W(1-\rho_n^t)$ weights need to be aggregated, and the total computation demand is expressed

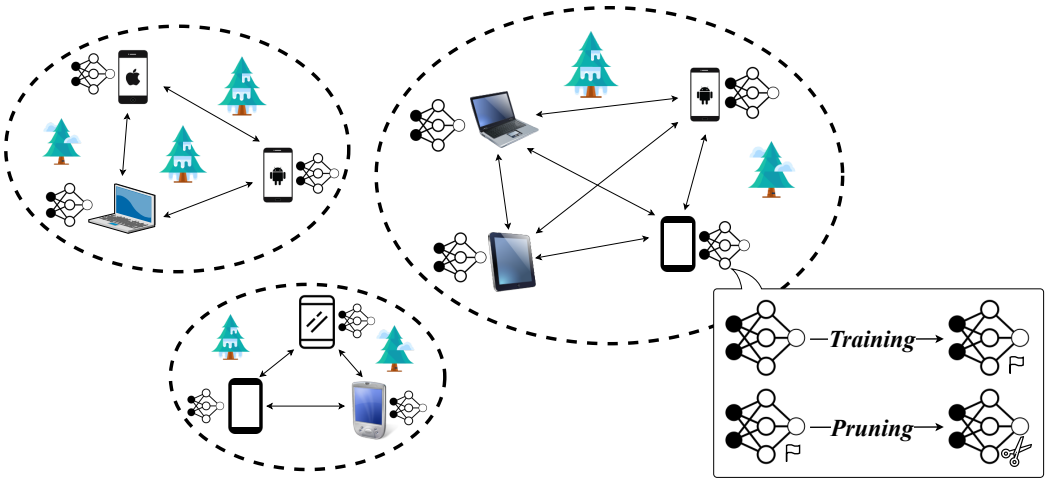


FIGURE 5.1: Architecture of the fully distributed federated learning framework with clustering and model pruning, where each device can only communicate with its neighbors inside its assigned cluster.

Algorithm 4 Iterative Synaptic Flow Pruning

Input:
  parameters $\mathbf{w}$, target pruning ratio ρ, number of pruning iterations s_p.
for $s = 1, \ldots, s_p$ **do**
  $\mathbf{w}^p = \mathbf{w} \odot \mathbf{m}$
  $\mathcal{L} = \mathbf{1}^T \left(\prod_{u=1}^{U} \left| \mathbf{w}^p_{[u]} \right| \right) \mathbf{1}$
  $\mathcal{S} = \frac{\partial \mathcal{L}}{\partial \mathbf{w}^p} \odot \mathbf{w}^p$
  $\zeta \leftarrow (1 - \rho^{-s/s_p})$ percentile of S
  $\mathbf{m} \leftarrow (\zeta < \mathcal{S})$
end for
Output:
  $\mathbf{w}^p = \mathbf{w} \odot \mathbf{m}$

as

$$\theta_n = \underbrace{\psi W(1-\rho_n^t)}_{\text{aggregation}} + \underbrace{f D_n E_n}_{\text{training}}. \tag{5.20}$$

Then, the expression of total computation latency in Eqn. (5.12) is rewritten as

$$\hat{l}_{n,t}^{comp} = \hat{l}_{n,t}^{aggre} + l_{n,t}^{train} = \frac{\psi W(1-\rho_n^t)}{f_n} + \frac{f D_n E_n}{f_n}. \tag{5.21}$$

In (5.21), the computational complexity related to the model pruning process is ignored, this is because compared with forward and backward propagation during local model training, the computational demand required by the model pruning process, such as calculating importance scores and applying binary masks, is significantly smaller [139, 147].

5.4.1.2 Communication latency

In the proposed FL framework, only the pruned model will be transmitted via D2D links, therefore, accounting for the transmission rate calculated in Eqn. (5.8) and model pruning process, the communication latency in Eqn. (5.9) is updated as

$$\hat{l}_{n,j,t}^{comm} = \frac{q_j^t(1-\rho_j^t)W}{R_{nj}^t}. \tag{5.22}$$

Furthermore, according to the above updated computation and communication latency (Eqn. (5.21) and Eqn. (5.22)), the overall latency in Eqn. (5.13) is rewritten as

$$\hat{l}_{j,t}^{total} = \hat{l}_{j,t}^{comp} + \hat{l}_{n,j,t}^{comm} \tag{5.23}$$

$$= \frac{\psi W(1-\rho_j^t)}{f_j} + \frac{fD_jE_j}{f_j} + \frac{q_j^t(1-\rho_j^t)W}{R_{nj}^t}. \tag{5.24}$$

Similarly, the bottleneck latency of the n^{th} device derived in Eqn. (5.14) is updated as

$$\hat{l}_{n,t} = \max_{j\in\mathcal{V}_t^n}\{\hat{l}_{j,t}^{total}\}. \tag{5.25}$$

5.4.2 Convergence analysis

In this subsection, we conduct the convergence analysis for the proposed system. We start by justifying the essential assumptions needed for the proof, followed by identifying the overall effective noise incurred during model pruning, D2D communications, and aggregation processes. Finally, the convergence bound is derived, and corresponding useful insights are provided.

5.4.2.1 Assumptions on the loss functions

In this chapter, we borrow the Assumptions 1, 2, 3, and 4 from Chapter 3 for our convergence analysis [24, 31, 72, 85]. Further, we make the following assumption, which is helpful in characterizing the impact of the model pruning procedure.

Assumption 9. (Bounded variance for model pruning [148])*: the squared l_2-norm between the original and pruned model weights satisfies*

$$\mathbb{E}\left\|\mathbf{w}_t^n - \mathbf{w}_t^{n,p}\right\|_2^2 \leq \rho_n A^2, \tag{5.26}$$

which leads to $\mathbb{E}\left\|\bar{\mathbf{w}}_t - \bar{\mathbf{w}}_t^p\right\|_2^2 \leq \sum_n \alpha_n^2 \rho_n A^2$, where A is a given constant.

Remark: To show that the results Assumption 9 is satisfied by the magnitude pruning scheme used in this chapter, we make the justification as follows.

We divide both sides of the Eqn. (5.26) by A^2, we have

$$\frac{\mathbb{E}\left\|\mathbf{w}_t^n - \mathbf{w}_t^{n,p}\right\|_2^2}{A^2} \leq \rho_n \tag{5.27}$$

In the magnitude pruning scheme, a higher pruning ratio, ρ_n, results in more weights being set to zero. This means that more elements in the vector $\mathbf{w}_t^{n,p}$ will be zero, increasing the distance between $\mathbf{w}_t^n$ and $\mathbf{w}_t^{n,p}$ (equivalent to a higher value of $\mathbb{E}\left\|\mathbf{w}_t^n - \mathbf{w}_t^{n,p}\right\|_2^2$), as explained in Eqn. (5.27), where A is a given constant.

5.4.2.2 D2D communications process

Without loss of generalization, we consider the scenario where within each cluster, one device is selected as the small base station (SBS) for performing the aggregation[2]. Similar to the derivation in Chapter 3, first, let the variable $\mathbf{v}_t$ denote the immediate result of one step SGD. Then, the update step of FL algorithm in a global round is expressed as

$$\mathbf{v}_{t+1}^n = \mathbf{w}_t^n - \eta_t \nabla F_n(\mathbf{w}_t^n, \xi_t^n), \tag{5.28}$$

$$\mathbf{w}_{t+1}^n = \begin{cases} \mathbf{v}_{t+1}^n, & \text{if } t+1 \notin \mathcal{T}, \\ \phi_n\left(\mathbf{v}_{t+1}^j, \forall j \in \mathcal{V}_{t+1}^n\right), & \text{if } t+1 \in \mathcal{T}. \end{cases} \tag{5.29}$$

For the D2D communication process, we start by considering transmitting one weight ($v_t^{j,p}$). Consider device m_k as the SBS for an arbitrary cluster k, as indicated in the previous section, the received weight from its neighbor j within this cluster is represented as

$$y_t^{m_k,j} = \sqrt{A_0}\left(\frac{d_{m_k j}}{d_0}\right)^{-\frac{\gamma}{2}} h_t^{m_k j} v_t^{j,p} \sqrt{\frac{P}{\sigma_w^2}} + n_t^{m_k}, \tag{5.30}$$

where $v_t^{j,p}$ is assumed to have zero mean and σ_w^2 variance. P is the transmit power constraint for each device. Considering the zero-forcing at the receiver side, the estimated weight is derived as

$$\begin{aligned} \hat{w}_t^{m_k,j,p} &= \sqrt{\frac{\sigma_w^2}{A_0 d_0^{\gamma} P}} \left(d_{m_k j}\right)^{\frac{\gamma}{2}} \left(h_t^{m_k jH} h_t^{m_k j}\right)^{-1} h_t^{m_k jH} y_t^{m_k,j} \\ &= v_t^{j,p} + \sqrt{\frac{\sigma_w^2}{A_0 d_0^{\gamma} P}} \left(d_{m_k j}\right)^{\frac{\gamma}{2}} \left(h_t^{m_k jH} h_t^{m_k j}\right)^{-1} h_t^{m_k jH} n_t^{m_k}, \end{aligned} \tag{5.31}$$

[2]In the simulation, for each global communication round, each device transmits the trained model to all its neighbors in the assigned cluster. In other words, all the devices serve as the SBS.

where the second term on the right-hand side of Eqn. (5.31) is the estimation error due to the D2D communication. Owing to channel hardening, we have $h_t^{m_k jH} h_t^{m_k j} \to 1$, the random variable for the estimation error can be simplified as $\sqrt{\frac{\sigma^2\sigma_w^2}{A_0 d_0^\gamma P}}\,(d_{m_k j})^{\frac{\gamma}{2}}$. We consider the case where the number of devices in a cluster is finite and they form a binomial point process. Then, the distance distribution between devices is well-approximated by the Gamma distribution. For a circle area, this approximation is good since the symmetry of the circle ensures that the distribution of distances has a consistent radial structure, which makes the Gamma distribution a good fit. In this way, the PDF for pairwise distance is denoted by $f(d) = \frac{d^{k_\tau - 1} e^{\frac{-d}{\theta_\tau}}}{\Gamma(k_\tau)\theta_\tau^{k_\tau}}$, where k_τ is the shape parameter, θ_τ is the scale parameter and $\Gamma(\cdot)$ denotes the Gamma function. Further, we consider the maximum distance between devices is d_{max} that is $d \in [0, d_{max}]$. This results in the truncated Gammas distribution PDF as $f(d|d_{max}) = \frac{d^{k_\tau - 1} e^{\frac{-d}{\theta_\tau}}}{\theta_\tau^{k_\tau}\gamma_\tau(k_\tau, \frac{d_{max}}{\theta_\tau})}, 0 < d < d_{max}$, where $\gamma_\tau(\cdot)$ denotes the lower incomplete Gamma function. Thus, the mean and variance of $d^{\frac{\gamma}{2}}$ are derived as

$$\mathbb{E}\left\{d^{\frac{\gamma}{2}}\right\} = \int_0^{d_{max}} d^{\frac{\gamma}{2}} f(d|d_{max}) d(d) = \frac{\theta_\tau^{\frac{\gamma}{2}} \gamma_\tau(\frac{\gamma}{2} + k_\tau, \frac{d_{max}}{\theta_\tau})}{\gamma_\tau(k_\tau, \frac{d_{max}}{\theta_\tau})},$$

$$\mathrm{Var}(d^{\frac{\gamma}{2}}) = \mathbb{E}\left\{(d^{\frac{\gamma}{2}})^2\right\} - \mathbb{E}\left\{d^{\frac{\gamma}{2}}\right\}^2 = \frac{\theta_\tau^{\gamma} \gamma_\tau(\gamma + k_\tau, \frac{d_{max}}{\theta_\tau})}{\gamma_\tau(k_\tau, \frac{d_{max}}{\theta_\tau})} - \left(\frac{\theta_\tau^{\frac{\gamma}{2}} \gamma_\tau(\frac{\gamma}{2} + k_\tau, \frac{d_{max}}{\theta_\tau})}{\gamma_\tau(k_\tau, \frac{d_{max}}{\theta_\tau})}\right)^2.$$

Therefore, the mean and variance of the error estimation term $\sqrt{\frac{\sigma^2\sigma_w^2}{A_0 d_0^\gamma P}}\,(d_{m_k j})^{\frac{\gamma}{2}}$ are represented as

$$\mu_e = \sqrt{\frac{\sigma^2\sigma_w^2}{A_0 d_0^\gamma P}} \frac{\theta_\tau^{\frac{\gamma}{2}} \gamma_\tau(\frac{\gamma}{2} + k_\tau, \frac{d_{max}}{\theta_\tau})}{\gamma_\tau(k_\tau, \frac{d_{max}}{\theta_\tau})}, \tag{5.32}$$

$$\sigma_e^2 = \frac{\sigma^2\sigma_w^2}{A_0 d_0^\gamma P} \left(\frac{\theta_\tau^{\gamma} \gamma_\tau(\gamma + k_\tau, \frac{d_{max}}{\theta_\tau})}{\gamma_\tau(k_\tau, \frac{d_{max}}{\theta_\tau})} - \left(\frac{\theta_\tau^{\frac{\gamma}{2}} \gamma_\tau(\frac{\gamma}{2} + k_\tau, \frac{d_{max}}{\theta_\tau})}{\gamma_\tau(k_\tau, \frac{d_{max}}{\theta_\tau})}\right)^2\right). \tag{5.33}$$

To this end, consider the vector $\hat{\mathbf{w}}_t^{m_k, j, p}$ containing all $W_g = (1 - \rho_n)W$ estimated weights, we have $\hat{\mathbf{w}}_t^{m_k, j, p} = \mathbf{v}_t^{j,p} + \mathbf{z}_t^{m_k j}$, where the covariance matrix for $\mathbf{z}_t^{m_k j}$ is $\sigma_e^2 \mathbf{I}$. Then, for $j \in \mathcal{V}_t^{m_k}$, the aggregated weights in this cluster is denoted as

$$\hat{\mathbf{w}}_t^{m_k, p} = \frac{1}{|\mathcal{V}_t^{m_k}|} \sum_{j \in \mathcal{V}_t^{m_k}} \mathbf{v}_t^{j,p} + \mathbf{z}_t^{m_k j}. \tag{5.34}$$

If we neglect the DL communication process[3] and all devices within the cluster will be broadcasted with $\hat{\mathbf{w}}_t^{m_k,p}$ for further training; that is, $\mathbf{w}_t^n = \hat{\mathbf{w}}_t^{m_k,p}$ for $n \in \mathcal{V}_t^{m_k}$.

Let $\bar{\mathbf{w}}_{t+1} = \sum_{n=1}^{N} \alpha_n \mathbf{w}_{t+1}^n$ denote the ensemble of all devices' weights at time instance $t+1$, and similarly, we define $\bar{\mathbf{v}}_{t+1} = \sum_{n=1}^{N} \alpha_n \mathbf{v}_{t+1}^n$ for the immediate results of SGD, then we have

$$\bar{\mathbf{w}}_{t+1} = \begin{cases} \bar{\mathbf{v}}_{t+1}, & \text{if } t+1 \notin \mathcal{T}, \\ \sum\limits_{n,k} \alpha_n \hat{\mathbf{w}}_{t+1}^{m_k,p}, & \text{if } t+1 \in \mathcal{T}. \end{cases} \tag{5.35}$$

Specifically, for the second case where $t+1 \in \mathcal{T}$, we expand it as

$$\begin{aligned} &\sum_{n,k} \alpha_n \hat{\mathbf{w}}_{t+1}^{m_k,p} = \sum_{n} \alpha_n \frac{1}{\left|\mathcal{V}_{t+1}^{m_k}\right|} \sum_{j \in \mathcal{V}_{t+1}^{m_k}} \mathbf{v}_t^{j,p} + \mathbf{z}_t^{m_k,j} \\ &= \left(\alpha_1 + \ldots + \alpha_{\tau_1}\right) \hat{\mathbf{w}}_{t+1}^{m_1,p} + \left(\alpha_{\tau_1+1} + \ldots + \alpha_{\tau_1+\tau_2}\right) \hat{\mathbf{w}}_{t+1}^{m_2,p} \\ &+ \ldots + \left(\alpha_{N-\tau_K+1} + \ldots + \alpha_N\right) \hat{\mathbf{w}}_{t+1}^{m_K,p}, \end{aligned}$$

where $\tau_k = \left|\mathcal{V}_{t+1}^{m_k}\right|$ denotes the number of devices in the k^{th} cluster at time instance $t+1$. Specially, for the case where $\alpha_1 = \alpha_2 = \ldots = \alpha_N = 1/N$, the above summation is rewritten as

$$\begin{aligned} &\sum_{n,k} \alpha_n \hat{\mathbf{w}}_{t+1}^{m_k,p} = \frac{\tau_1}{N} \hat{\mathbf{w}}_{t+1}^{m_1,p} + \frac{\tau_2}{N} \hat{\mathbf{w}}_{t+1}^{m_2,p} + \ldots + \frac{\tau_K}{N} \hat{\mathbf{w}}_{t+1}^{m_K,p} \\ &= \frac{\tau_1}{N} \frac{1}{\tau_1} \sum_{j \in \mathcal{V}_{t+1}^{m_1}} \left(\mathbf{v}_{t+1}^{j,p} + \mathbf{z}_{t+1}^{m_1 j}\right) + \frac{\tau_2}{N} \frac{1}{\tau_2} \sum_{j \in \mathcal{V}_{t+1}^{m_2}} \left(\mathbf{v}_{t+1}^{j,p} + \mathbf{z}_{t+1}^{m_2 j}\right) + \ldots + \frac{\tau_K}{N} \frac{1}{\tau_K} \sum_{j \in \mathcal{V}_{t+1}^{m_K}} \left(\mathbf{v}_{t+1}^{j,p} + \mathbf{z}_{t+1}^{m_K j}\right) \\ &= \frac{1}{N} \left(\sum_{j} \mathbf{v}_{t+1}^{j,p} + \sum_{k} \sum_{j \in \mathcal{V}_{t+1}^{m_k}} \mathbf{z}_{t+1}^{m_k,j} \right) = \bar{\mathbf{v}}_{t+1}^{p} + \frac{1}{N} \underbrace{\sum_{\lambda_{t+1}^{nj}=1} \mathbf{z}_{t+1}^{nj}}_{(N-K) \text{ terms}}, \end{aligned} \tag{5.36}$$

where λ_{t+1}^{nj} is the clustering indicator for identifying whether device j and n are in the same cluster for time instance $t+1$; that is, $\lambda_{t+1}^{nj} = 1$ if $j \in \mathcal{V}_{t+1}^{n}$ and $\lambda_{t+1}^{nj} = 0$ otherwise. When $t+1 \in \mathcal{T}$, the device operating as an SBS can employ its locally trained model weights directly for the aggregation process[4]. In this context, these weights bypass the necessity for D2D communications. Therefore, in Eqn. (5.36), it is noticed that there are $N-K$ noise terms involved, where K represents the total number of clusters. Overall,

[3] In the fully distributed FL procedure, during each global communication round, the D2D communication occurs only once between each pair of neighboring nodes, which is equivalent to neglecting the DL communication.

[4] $d_{m_k m_k} = 0$, thus $\mathbf{z}_{t+1}^{m_k,m_k} = 0, \forall k$

we have

$$\bar{\mathbf{w}}_{t+1} = \begin{cases} \bar{\mathbf{v}}_{t+1}, & \text{if } t+1 \notin \mathcal{T}, \\ \bar{\mathbf{v}}^p_{t+1} + \frac{1}{N} \sum\limits_{\lambda^{nj}_{t+1}=1} \mathbf{z}^{nj}_{t+1}, & \text{if } t+1 \in \mathcal{T}. \end{cases} \tag{5.37}$$

5.4.2.3 Convergence bound

Accounting for the overall effective noise incurred during the post-training model pruning and D2D communication procedures, we first bound the squared l_2-norm of the difference between weights ensemble and the optimal solution with respect to the immediate ensemble result of one-step SGD ($\bar{\mathbf{v}}_t$).

Lemma 5.1. *For the considered model pruning-enabled distributed FL system in D2D networks, given the optimal model weights* $\mathbf{w}^*$*, the squared* l_2*-norm of the model error is bounded as*

$$\mathbb{E}\left\|\bar{\mathbf{w}}_{t+1} - \mathbf{w}^*\right\|_2^2 \leq \mathbb{E}\left\|\bar{\mathbf{v}}_{t+1} - \mathbf{w}^*\right\|_2^2 + \Bigg(\underbrace{\frac{1}{N^2}\sum_n \rho_n^t A^2}_{\substack{\text{noise arisen from} \\ \text{model pruning}}} + \underbrace{\frac{W_g(N-K)}{N^2}\sigma_z^2}_{\substack{\text{noise arisen from} \\ \text{D2D communications}}} \Bigg) \delta_{t+1\in\mathcal{T}},$$

where $\sigma_z^2 = \mu_e^2 + \sigma_e^2 = \frac{\sigma^2\sigma_w^2}{A_0 d_0^\gamma P}\left(\frac{\theta_\tau^\gamma \gamma_\tau(\gamma + k_\tau, \frac{d_{max}}{\theta_\tau})}{\gamma_\tau(k_\tau, \frac{d_{max}}{\theta_\tau})}\right)$*, A is a constant, ρ_n denotes the pruning ratio for device n, K represents the number of clusters in the system, and $\delta_{t+1\in\mathcal{T}}$ is an indicator function, taking value 1, when $t+1 \in \mathcal{T}$, and 0 otherwise.*

Proof: The left-hand side of the Lemma 5.1 is rewritten as

$$\begin{aligned} &\mathbb{E}\left\|\bar{\mathbf{w}}_{t+1} - \mathbf{w}^*\right\|_2^2 \\ &= \begin{cases} \mathbb{E}\left\|\bar{\mathbf{v}}_{t+1} - \mathbf{w}^*\right\|_2^2, & \text{if } t+1 \notin \mathcal{T}, \\ \mathbb{E}\left\| \bar{\mathbf{v}}^p_{t+1} + \frac{1}{N} \sum\limits_{\lambda^{nj}_{t+1}=1} \mathbf{z}^{nj}_{t+1} - \mathbf{w}^* \right\|_2^2, & \text{if } t+1 \in \mathcal{T}. \end{cases} \end{aligned} \tag{5.38}$$

For the second case where $t+1 \in \mathcal{T}$, we expand it as

$$\begin{aligned}
&\mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}^{p}+\bar{\mathbf{v}}_{t+1}-\bar{\mathbf{v}}_{t+1}+\frac{1}{N}\sum_{\lambda_{t+1}^{nj}=1}\mathbf{z}_{t+1}^{nj}-\mathbf{w}^{*}\right\|_2^2 \\
&=\mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}-\mathbf{w}^{*}\right\|_2^2+\mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}^{p}-\bar{\mathbf{v}}_{t+1}\right\|_2^2+\frac{1}{N^2}\mathbb{E}\left\|\sum_{\lambda_{t+1}^{nj}=1}\mathbf{z}_{t+1}^{nj}\right\|_2^2 \\
&\overset{(a)}{\leq}\mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}-\mathbf{w}^{*}\right\|_2^2+\frac{1}{N^2}\sum_{n}\rho_n^t A^2+\frac{W_g(N-K)}{N^2}\sigma_z^2,
\end{aligned}$$

where $\sigma_z^2=\mu_e^2+\sigma_e^2=\frac{\sigma^2\sigma_w^2}{A_0 d_0^{\gamma} P}\left(\frac{\theta_\tau^{\gamma}\gamma_\tau(\gamma+k_\tau,\frac{d_{max}}{\theta_\tau})}{\gamma_\tau(k_\tau,\frac{d_{max}}{\theta_\tau})}\right)$ denotes the variance of the overall effective noise introduced during clustered D2D communications process. In (a), Assumption 9 and the following results are used

$$\begin{aligned}
\mathbb{E}\left\|\mathbf{z}_{t+1}^{nj}\right\|_2^2&=\sum_{i=1}^{W_g}\mathbb{E}\left\{\left(z_{t+1,i}^{nj}\right)^2\right\} \\
&=\sum_{i=1}^{W_g}\mathrm{Var}\left(z_{t+1,i}^{nj}\right)+\left(\mathbb{E}\left\{z_{t+1,i}^{nj}\right\}\right)^2 \\
&=W_g\left(\sigma_e^2+\mu_e^2\right),
\end{aligned}$$

where the expression for μ_e and σ_e^2 can be found in Eqn. (5.32) and (5.33), given the Gamma distribution of the distance between device n and j. Therefore, considering both cases in Eqn. (5.38), we have

$$\mathbb{E}\left\|\bar{\mathbf{w}}_{t+1}-\mathbf{w}^{*}\right\|_2^2\leq\mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}-\mathbf{w}^{*}\right\|_2^2+\left(\frac{1}{N^2}\sum_{n}\rho_n^t A^2+\frac{W_g(N-K)}{N^2}\sigma_z^2\right)\delta_{t+1\in\mathcal{T}},$$

and this concludes our proof for Lemma 5.1.

The above result shows that if no clustering and D2D communications take place, i.e., $\sigma_z^2=0$, the aggregated model weights only suffer from the noise introduced by the post-training model pruning. Further, when $t+1\notin\mathcal{T}$; that is, $\delta_{t+1}=0$, one can witness that $\bar{\mathbf{w}}_{t+1}=\bar{\mathbf{v}}_{t+1}$, and this aligns with the definition of immediate one-step SGD result of $\bar{\mathbf{v}}_t$.

The following result utilizes μ-strong convexity of loss functions (Assumption 2) and bounded SGD (Assumptions 3 and 4) to upper-bound the next step error.

Lemma 5.2. *Let Assumptions 1-4 hold. Then, with respect to the optimized model weights* $\mathbf{w}^*$*, the squared* l_2*-norm of the next step model update error can be bounded as*

$$\mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}-\mathbf{w}^*\right\|_2^2 \leq (1-\eta_t\mu)\,\mathbb{E}\left\|\bar{\mathbf{w}}_t-\mathbf{w}^*\right\|_2^2+\frac{\eta_t^2}{N^2}\left(\sum_n\left(\sigma_n^2+G^2\right)\right). \tag{5.39}$$

Proof: The left-hand side of the lemma 5.2 is simplified as

$$\begin{aligned}
&\mathbb{E}\left\|\bar{\mathbf{v}}_{t+1}-\mathbf{w}^*\right\|_2^2\\
&=\mathbb{E}\left\|\bar{\mathbf{w}}_t-\eta_t\mathbf{g}_t-\mathbf{w}^*+\eta_t\nabla F(\bar{\mathbf{w}}_t)-\eta_t\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2\\
&\overset{(a)}{=}\mathbb{E}\left\|\bar{\mathbf{w}}_t-\eta_t\nabla F(\bar{\mathbf{w}}_t)-\mathbf{w}^*\right\|_2^2+\eta_t^2\mathbb{E}\left\|\mathbf{g}_t-\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2\\
&=\mathbb{E}\left\|\bar{\mathbf{w}}_t-\mathbf{w}^*\right\|_2^2-2\eta_t\mathbb{E}\left(\bar{\mathbf{w}}_t-\mathbf{w}^*\right)^T\nabla F(\bar{\mathbf{w}}_t)+\eta_t^2\mathbb{E}\left\|\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2+\eta_t^2\mathbb{E}\left\|\mathbf{g}_t-\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2\\
&\overset{(b)}{\leq}\mathbb{E}\left\|\bar{\mathbf{w}}_t-\mathbf{w}^*\right\|_2^2+\eta_t^2\mathbb{E}\left\|\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2+\eta_t^2\mathbb{E}\left\|\mathbf{g}_t-\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2\\
&+2\eta_t\mathbb{E}\left[F(\mathbf{w}^*)-F(\bar{\mathbf{w}}_t)-\frac{\mu}{2}\|\bar{\mathbf{w}}_t-\mathbf{w}^*\|_2^2\right]\\
&\overset{(c)}{\leq}(1-\eta_t\mu)\,\mathbb{E}\left\|\bar{\mathbf{w}}_t-\mathbf{w}^*\right\|_2^2+\frac{\eta_t^2}{N^2}\sum_n\left(\sigma_n^2+G^2\right)+2\eta_t\mathbb{E}\left[F(\mathbf{w}^*)-F(\bar{\mathbf{w}}_t)\right]\\
&\overset{(d)}{\leq}(1-\eta_t\mu)\,\mathbb{E}\left\|\bar{\mathbf{w}}_t-\mathbf{w}^*\right\|_2^2+\frac{\eta_t^2}{N^2}\sum_n\left(\sigma_n^2+G^2\right),
\end{aligned}$$

where in (a), the results (from Chapter 3) that $\mathbf{g}_t=\sum_{n=1}^N\alpha_n\nabla F_n(\mathbf{w}_t^n,\xi_t^n)$, $\bar{\mathbf{g}}_t=\sum_{n=1}^N\alpha_n\nabla F_n(\mathbf{w}_t^n)$, and $\mathbb{E}\left\{\mathbf{g}_t-\bar{\mathbf{g}}_t\right\}=\mathbf{0}$ are used; (b) is obtained from μ-strong convexity as specified in Assumption 2, i.e., $-2(\mathbf{w}-\mathbf{v})^T\nabla F(\mathbf{w})\leq 2F(\mathbf{v})-2F(\mathbf{w})-\mu\|\mathbf{v}-\mathbf{w}\|_2^2$; in (c), the results from Assumptions 3 and 4 are employed to substitute the terms $\mathbb{E}\left\|\mathbf{g}_t-\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2$ and $\mathbb{E}\left\|\nabla F(\bar{\mathbf{w}}_t)\right\|_2^2$, respectively; and the result in (d) comes from the fact that $F(\mathbf{w}^*)\leq F(\bar{\mathbf{w}}_t)$.

The above result shows that if η_t and μ are chosen properly, the model updates lead to non-increasing errors, in other words, the convergence of the proposed algorithm is guaranteed. With Lemma 5.1 and 5.2, we manage to get rid of the immediate result $\bar{\mathbf{v}}_t$ when bounding $\mathbb{E}\left\|\bar{\mathbf{w}}_{t+1}-\mathbf{w}^*\right\|_2^2$, and this enables us to derive the convergence bound on overall loss function in the following theorem, which employs the smoothness of F_n (Assumption 1).

Theorem 5.3. *Let Assumptions 1-4 and 9 hold, and* σ_z^2 *be defined therein. Then, the proposed FL algorithm satisfies*

$$\mathbb{E}\left\{F\left(\bar{\mathbf{w}}_t\right)\right\}-F^*\leq\frac{L}{2}\cdot\frac{\nu}{t+\beta_1}, \tag{5.40}$$

where $\eta_t=\frac{\beta_0}{t+\beta_1}$, $\nu\geq\frac{\beta_0^2Q}{\beta_0\mu-1}$, $\beta_1>0$, $\beta_0>\frac{1}{\mu}$,

and $Q = \frac{1}{N^2}\left(\sum_n \left(\sigma_n^2 + G^2\right) + \left(\sum_n \rho_n^t A^2 + W_g(N-K)\sigma_z^2\right)\frac{\delta_{t+1\in\mathcal{T}}}{\eta_t^2}\right)$.

Proof: For the L-smooth function, we write

$$\mathbb{E}\left\{F\left(\bar{\mathbf{w}}_{t+1}\right)\right\} - F^* \tag{5.41}$$

$$\leq \mathbb{E}\left\{\left(\bar{\mathbf{w}}_{t+1} - \mathbf{w}^*\right)^T \nabla F(\mathbf{w}^*)\right\} + \frac{L}{2}\Delta_{t+1} \tag{5.42}$$

$$\overset{(a)}{=} \frac{L}{2}\Delta_{t+1} \tag{5.43}$$

$$\overset{(b)}{\leq} \frac{L}{2} \cdot \frac{\nu}{t+1+\beta_1}, \tag{5.44}$$

where in (a), zero mean result $\mathbb{E}\left\{\bar{\mathbf{w}}_t - \mathbf{w}^*\right\} = \mathbf{0}$ is used, and $\Delta_t = \mathbb{E}\|\bar{\mathbf{w}}_t - \mathbf{w}^*\|_2^2$; the inequality, $\Delta_t \leq \frac{\nu}{t+\beta_1}$ in (b) is proved via induction as follows. Let Q be defined therein, $\nu \geq \frac{\beta_0^2 Q}{\beta_0\mu - 1}, \eta_t = \frac{\beta_0}{t+\beta_1}$, where $\beta_0 > \frac{1}{\mu}, \beta_1 > 0$. Then, the definition of ν guarantees that the inequality in (b) holds for $t = 0$, assume the conclusion holds for some $t > 0$, by substituting the result of Lemma 5.2 into the right-hand side of Lemma 5.1, we have

$$\Delta_{t+1} \leq \left(1 - \eta_t\mu\right)\Delta_t + \eta_t^2 Q \tag{5.45}$$

$$\leq \frac{\nu}{t+\beta_1} - \frac{\beta_0\mu\nu}{\left(t+\beta_1\right)^2} + \frac{\beta_0^2 Q}{\left(t+\beta_1\right)^2} \tag{5.46}$$

$$= \frac{\nu(t+\beta_1) - \nu}{\left(t+\beta_1\right)^2} + \frac{\nu - \beta_0\mu\nu + \beta_0^2 Q}{\left(t+\beta_1\right)^2} \tag{5.47}$$

$$\overset{(a)}{\leq} \frac{\nu}{t+1+\beta_1} + 0, \tag{5.48}$$

where in inequality (a), the fact that $(t+\beta_1)^2 - 1 < (t+\beta_1)^2$ and condition $\nu \geq \frac{\beta_0^2 Q}{\beta_0\mu - 1}$ are used and this concludes our proof for the Theorem 5.3.

Theorem 5.3 tells that the modeling error for the proposed FL algorithm reduces in an $\mathcal{O}\left(t_\tau^{-1}\sigma_z^2\right)$ manner, where $t_\tau = t - t_0$ for $t_0 < t$. This means that the algorithm converges in time, and the accuracy is inversely proportional to effective error variance σ_z^2, whose expression accounts for and explains the effects of the communication and model pruning constraints on the convergence. The convergence bound exhibits a linear dependence on L-smoothness while being inversely proportional to μ-convexity, indicating that stronger convexity results in lower convergence error.

5.4.3 Optimization problem transformation

Motivated by Theorem 5.3, we need to maximize the convergence rate, which is equivalent to minimizing the convergence bound (ν). Furthermore, it is witnessed that ν is in a linear relationship with pruning ratio ρ_n^t. Therefore, the optimization problem is

reformulated as

$$\min_{a_{nj}^t, \rho_j^t} \sum_{j=1}^{|\mathcal{V}_t^n|} \rho_j^t, \tag{5.49}$$

subject to

$$\hat{l}_{n,t} \leq l_{th}, \tag{5.50}$$

$$\sum_{j=1}^{|\mathcal{V}_t^n|} a_{nj}^t \leq 1, \tag{5.51}$$

$$0 \leq a_{nj}^t \leq 1, \quad j \in \mathcal{V}_t^n, \tag{5.52}$$

$$0 \leq \rho_j^t \leq 1, \quad j \in \mathcal{V}_t^n. \tag{5.53}$$

Eqn. (5.53) represents the constraints of the pruning ratio threshold. It is noticed that, to minimize the global loss function, the values of pruning ratio and bandwidth allocation should be carefully selected, based on the given constraints.

5.5 Optimization of Pruning Ratio and Bandwidth Allocation

In this section, we derive the closed-form solution of the transformed optimization problem discussed in the previous section. We divide the problem in Eqn. (5.49) into two sub-problems.

5.5.1 Optimization of pruning ratio

From the constraint in Eqn. (5.50), for n^{th} device, the overall latency of all its neighbors ($j \in \mathcal{V}_t^n$) should not exceed the given latency constraint, which is written as

$$\frac{\psi W(1-\rho_j^t)}{f_j} + \frac{fD_jE_j}{f_j} + \frac{q_j^t(1-\rho_j^t)W}{R_{nj}^t} \leq l_{th}. \tag{5.54}$$

Therefore, by rewriting the expression in Eqn. (5.54) and let $(e)^+ = \max(e, 0)$, the lower bound of pruning ratio ρ_j^t in the t^{th} global communication round is derived as

$$\rho_j^{t*} \geq \left(1 - \frac{R_{nj}^t(l_{th} - \frac{fD_jE_j}{f_j})}{R_{nj}^t \frac{\psi W}{f_j} + q_j^t W}\right)^+. \tag{5.55}$$

From Eqn. (5.55), it is witnessed that the pruning ratio is jointly decided by the computing capability of the local device (f_j) and transmission rate (R_{nj}^t). The latency threshold l_{th} also has an effect on the pruning ratio. When the latency threshold is higher, a lower value of pruning ratio can be adopted to allow more weight transmission and vice versa.

5.5.2 Optimization of bandwidth allocation

Based on the derived lower bound of the pruning ratio in Eqn. (5.55), the optimization problem in Eqn. (5.49) is rewritten as

$$\min_{a_{nj}} \sum_{j=1}^{|\mathcal{V}_t^n|} \left(1 - \frac{R_{nj}^t (l_{th} - \frac{fD_jE_j}{f_j})}{R_{nj}^t \frac{\psi W}{f_j} + q_j^t W} \right). \tag{5.56}$$

To prove that the optimization problem in Eqn. (5.56) is convex concerning the bandwidth fraction a_{nj}^t, let $f(a_{nj}^t) = \left(1 - \frac{a_{nj}^t J_1}{a_{nj}^t J_2 + J_3}\right)$, where $J_1 = \frac{R_{nj}^t}{a_{nj}^t}(l_{th} - \frac{fD_jE_j}{f_j}), J_2 = \frac{R_{nj}^t}{a_{nj}^t}\frac{\psi W}{f_j}, J_3 = q_j^t W, J_1, J_2, J_3 > 0$. We start with calculating the first-order derivative of $f(a_{nj}^t)$ as

$$f'(a_{nj}^t) = -\frac{J_1 J_3}{(a_{nj}^t J_2 + J_3)^2}, \tag{5.57}$$

Then the second-order derivative is computed as

$$f''(a_{nj}^t) = \frac{2 J_1 J_2 J_3}{(a_{nj}^t J_2 + J_3)^3}, \tag{5.58}$$

Since $a_{nj}^t > 0$, $f''(a_{nj}^t) > 0$. Therefore, the optimization problem in Eqn. (5.56) is convex with respect to bandwidth fraction a_{nj}^t. To this end, the optimal bandwidth allocation is derived via Lagrange multiplier, which is shown as follows. First, the Lagrange function is constructed as

$$\mathcal{L}(a_{nj}^t, \lambda) = \sum_{j=1}^{|\mathcal{V}_t^n|} \left(1 - \frac{a_{nj}^t B \log_2 \left(1 + \frac{p_{nj} \left|h'_{nj,t}\right|^2}{\sigma^2}\right) (l_{th} - \frac{fD_jE_j}{f_j})}{a_{nj}^t B \log_2 \left(1 + \frac{p_{nj} \left|h'_{nj,t}\right|^2}{\sigma^2}\right) \frac{\psi W}{f_j} + q_j^t W} \right) + \lambda \left(\sum_{j=1}^{|\mathcal{V}_t^n|} a_{nj}^t - 1\right),$$

where λ denotes the Lagrange multiplier. Subsequently, the *Karush-Kuhn-Tucker* (KKT) conditions are utilized to solve the problem as

$$\frac{\partial \mathcal{L}}{\partial a_{nj}^t} = \lambda - \frac{B\log_2\left(1+\frac{p_{nj}\left|h'_{nj,t}\right|^2}{\sigma^2}\right)(l_{th}-\frac{fD_jE_j}{f_j})q_j^tW}{\left[a_{nj}^t B\log_2\left(1+\frac{p_{nj}\left|h'_{nj,t}\right|^2}{\sigma^2}\right)\frac{\psi W}{f_j}+q_j^tW\right]^2} = 0, \tag{5.59}$$

$$\lambda(\sum_{j=1}^{|\mathcal{V}_t^n|} a_{nj}^t - 1) = 0, \lambda \geq 0. \tag{5.60}$$

Therefore, based on these KKT conditions, the optimal wireless bandwidth allocation is obtained as

$$a_{nj}^{t*} = \frac{\sqrt{\frac{B\log_2\left(1+\frac{p_{nj}\left|h'_{nj,t}\right|^2}{\sigma^2}\right)(l_{th}-\frac{fD_jE_j}{f_j})q_j^tW}{\lambda^*}} - q_j^tW}{B\log_2\left(1+\frac{p_{nj}\left|h'_{nj,t}\right|^2}{\sigma^2}\right)\frac{\psi W}{f_j}}, \tag{5.61}$$

where λ^* is the optimal Lagrange multiplier. Then, the lower bound of the optimal pruning ratio ρ_j^{t*} is calculated by plugging Eqn. (5.61) into Eqn. (5.55) as

$$\rho_j^{t*} = \left[1 - \frac{Y\left(l_{th}f_j - fD_jE_j\right)}{\psi W(Y+q_j^tW)}\right]^+, \tag{5.62}$$

where $Y = \sqrt{\frac{B\log_2\left(1+\frac{p_{nj}\left|h'_{nj,t}\right|^2}{\sigma^2}\right)(l_{th}-\frac{fD_jE_j}{f_j})q_j^tW}{\lambda^*}} - q_j^tW.$

Remark: From Eqn. (5.61), it can be inferred that, within the specified latency constraint, devices experiencing poorer channel conditions (h) should be assigned a higher portion of bandwidth resources. Similarly, devices with higher computing capability are allocated additional bandwidth to enhance the convergence rate. From Eqn. (5.62), it can be deduced that the pruning ratio decreases as bandwidth allocation increases. Practically, this implies that with more wireless resources allocated to the device, the value of ρ_j^t can be reduced to allow more weighs transmission, under the specified latency constraint. Furthermore, interesting trade-offs can also be observed; for instance, when using more quantization bits for transmitted weights, a higher pruning ratio should be employed to reduce overall communication overhead, ensuring compliance with the latency threshold constraint. Complete procedures of the proposed framework are shown in Algorithm 5.

Algorithm 5 Clustered fully distributed FL algorithm

Initialization:
  Initialize $R, N, \eta_t, \mathcal{V}_1^n, p_1^n, \forall n$
for $r = 1, \ldots, R$ **do**
  for $n = 1, \ldots, N$ **do**
    if $r \neq 1$ **then**
      $\mathbf{w}_t^n = \hat{\mathbf{w}}_t^{n,p}$
    end if
    for $t = \sum_{\hat{r}=1}^{r} S_{\hat{r}} - S_r, \ldots, \sum_{\hat{r}=1}^{r} S_{\hat{r}} - 1$ **do**
      Update local weights as:
        $\mathbf{w}_{t+1}^n = \mathbf{w}_t^n - \eta_t \nabla F_n(\mathbf{w}_t^n, \xi_t^n)$
    end for
    $\mathbf{w}_t^{n,p} = \textbf{pruning}(\mathbf{w}_t^n), \forall n$. ρ_n^{t*} is calculated according to Eqn. (5.62)
  end for
  if $r = 1$ **then**
    All devices initialize its local weights $\mathbf{w}_0^n, \forall n$
  else
    for $n = 1, \ldots, N$ **do**
      for $j = 1, \ldots, |\mathcal{V}_r^n|$ **do**
        send $\mathbf{w}_t^{j,p}$ to device n via quantization, modulation and fading channel,
        based on the bandwidth allocation a_{nj}^{t*} calculated in Eqn. (5.61).
      end for
      device n aggregates received weights and produces $\hat{\mathbf{w}}_t^{n,p}$.
    end for
  end if
  All devices select a random angle and move a random distance.
    $p_{r+1}^n \longleftarrow \textbf{update}(p_r^n), \forall n$
  Update device clusters.
    $\mathcal{V}_{r+1}^n \longleftarrow \textbf{kmeans}(p_{r+1}^n, \forall n)$.
end for

5.6 Simulation Results

5.6.1 Simulation setup

5.6.1.1 Communication, clustering and model Pruning

The bandwidth is set to be 20 MHz. Number of devices N is set to be 10, number of global rounds $R = 100$, latency threshold $l_{th} = 30$ ms. Investigation in prior research [134, 136, 149] has shown that, selecting a larger number of clusters (K) increases communication and computation overhead at local clients, while smaller values of K may not achieve satisfactory learning performance in the framework of federated learning with clustered clients. In our simulation, we chose $K = 3$, consistent with the setup in [136]. This choice has been demonstrated to be both effective and efficient for the number of clients considered. Gaussian noise is assumed to have zero mean and unit variance at local device side. The model weights are quantized using a 16-bit uniform quantization

scheme, followed by Quadrature Phase Shift Keying (QPSK) for modulation. For i.i.d. Rayleigh fading model, channel state information is assumed known at devices and we set $A_0 = 1, d_0 = 50$ m and $\gamma = 3$. The computing capability of the device is 20 GFLOPs. For model pruning, unless otherwise noted, pruning ratio and number of pruning iterations are set to be 50% and 6, respectively. The devices are positioned on a rectangular planar surface of size 100×100 m^2. Following each global round, each device will independently determine a random angle and then proceed to move a distance uniformly selected from the range of $[0, 10$ m$]$. If the device reaches the boundary of the considered plane, it will reflect back.

In the analysis, we assume that the devices are spread on a circle area, while in the simulation, we consider a square plane. We would like to note that both the circular and square areas are bounded finite regions where devices are spread uniformly. This means that while the exact geometry differs, the overall properties of random, bounded movement and uniform spatial distribution remain consistent. Further, considering the symmetric property, the behavior of distances in a rectangular area can also be approximated using Gamma distribution as long as the aspect ratio of the rectangle is not too extreme. For a square area (aspect ratio is 1), we consider in the simulation, the difference between circular and square distance distributions is minimal.

We also want to note that, the mobility model used in the current is a variant of the Random Waypoint Model (RWM). In our mobility model, instead of moving to a randomly selected destination (which is conducted in RWM), devices select a random angle and move a random distance within a bounded range. Further, The random distance selection in our mobility model serves a similar purpose to the random destination in the RWP model, ensuring devices are spread out and move independently across the area.

5.6.1.2 Training setup

Each device has 50 training samples locally and has a testing set of 200 samples. In the i.i.d. case, overall training and testing sets are randomly shuffled and evenly split up among all devices. Conversely, in the non-i.i.d. scenario, two labels are randomly selected to serve as dominant labels for each local device. We introduce the concept of the "non-i.i.d. factor", which represents the proportion of the training set occupied by samples characterized by dominant labels. It is noteworthy that the testing images within the testing sets maintain a uniform distribution, implying that they are i.i.d. testing data samples. Learning rate and batch size are set to 0.01 and 10, respectively. Local CNN model consists of three convolutional layers with ReLU activation, a fully connected layer and a final classification layer with softmax function. The numbers of the filters in these three convolutional layers are 8, 16 and 32, respectively, and each of them is followed by a 2×2 max-pooling layer.

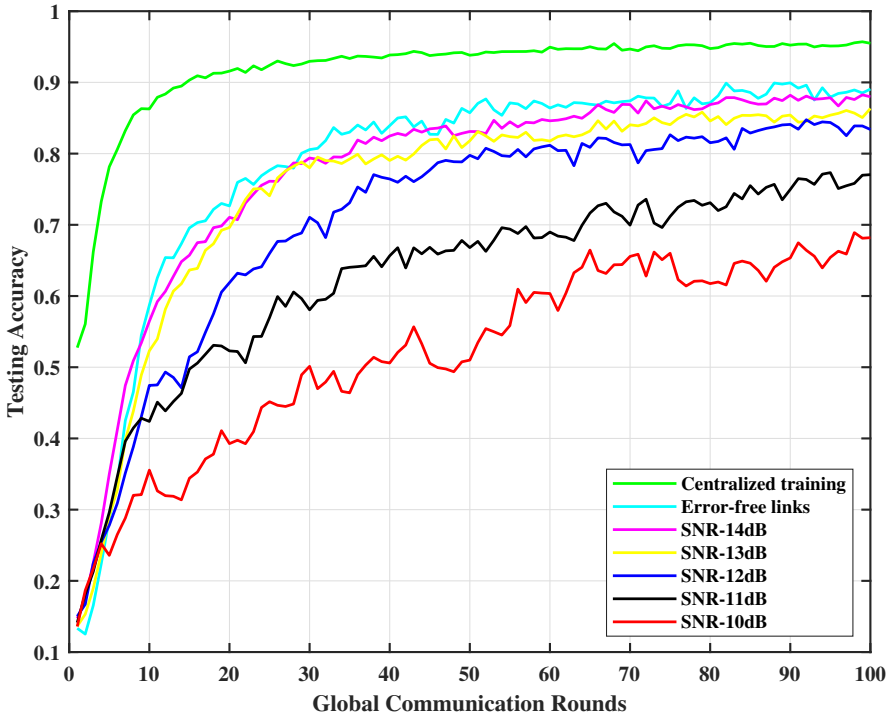


FIGURE 5.2: Comparison of the system performance on MNIST dataset in terms of testing accuracy for different values of SNR.

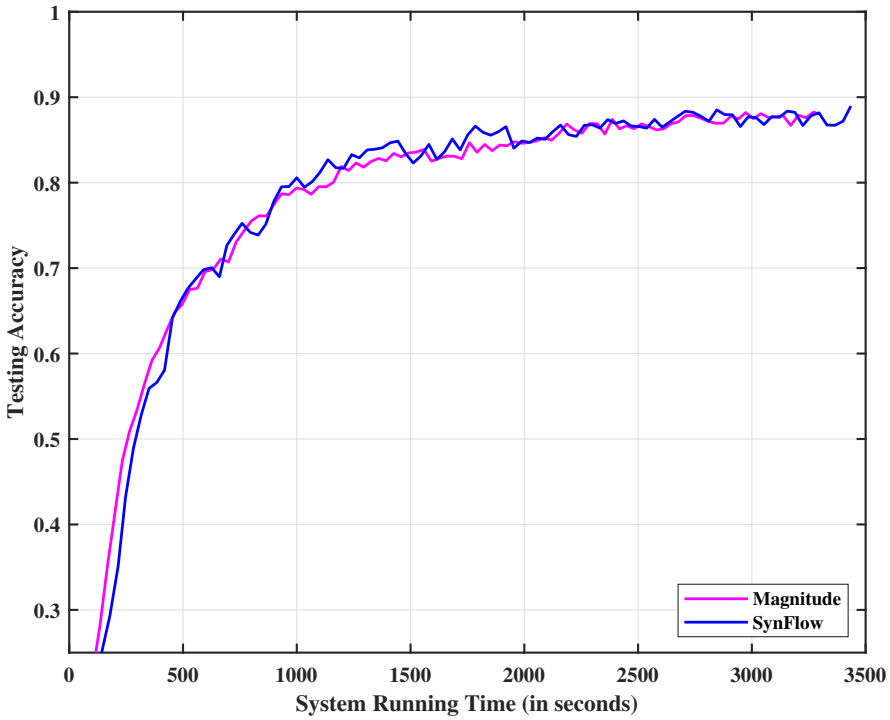
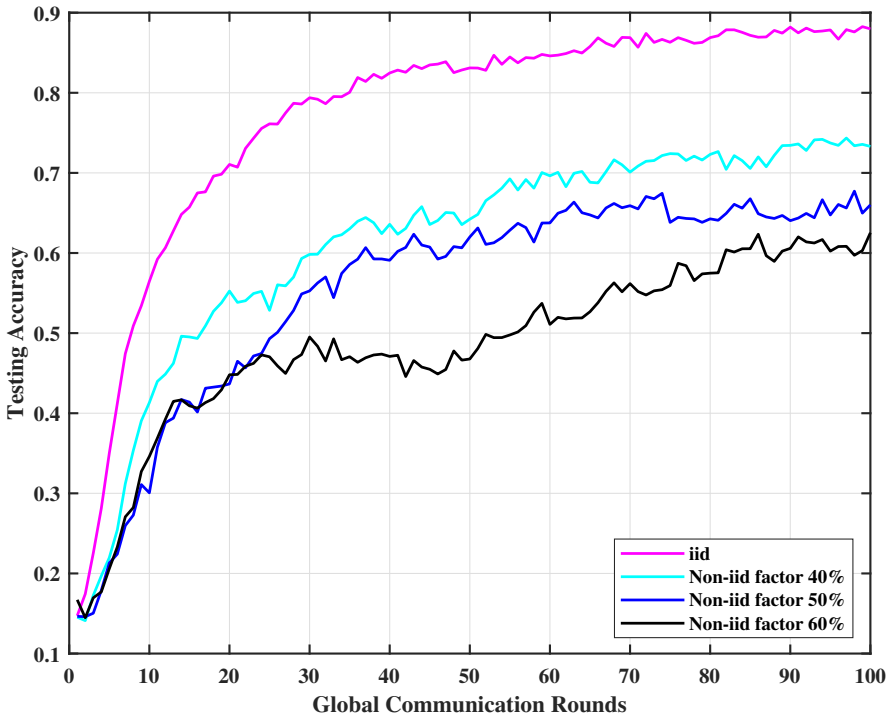
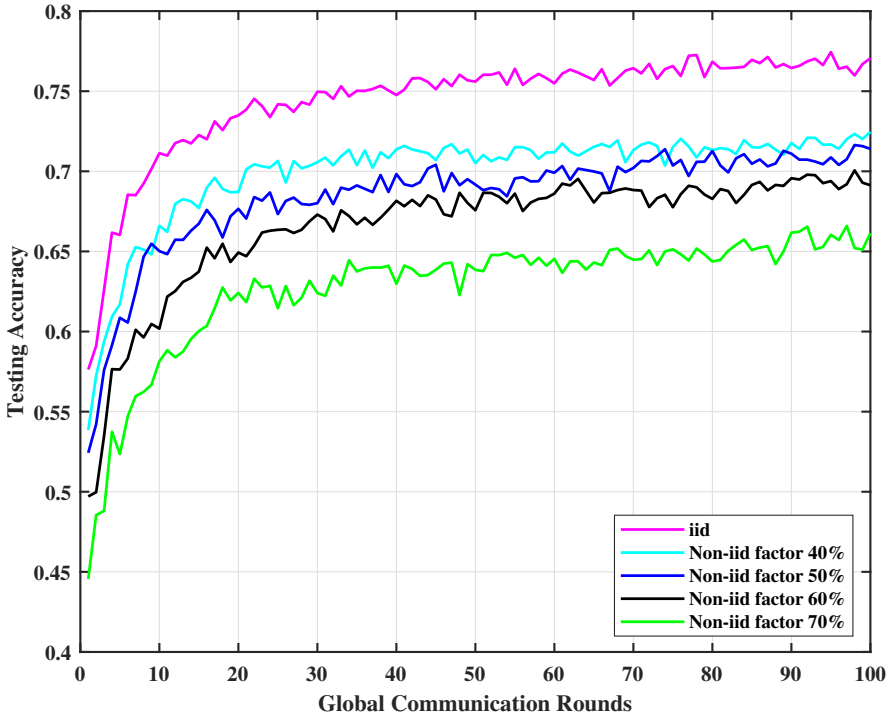


FIGURE 5.3: Testing accuracy on MNIST dataset with different pruning methods. SNR is 14 dB.



(a)



(b)

FIGURE 5.4: Testing accuracy on MNIST (a) and FMNIST (b) datasets with different non-i.i.d. factors. Pruning ratio is 50%. Value of SNR is 14 dB.

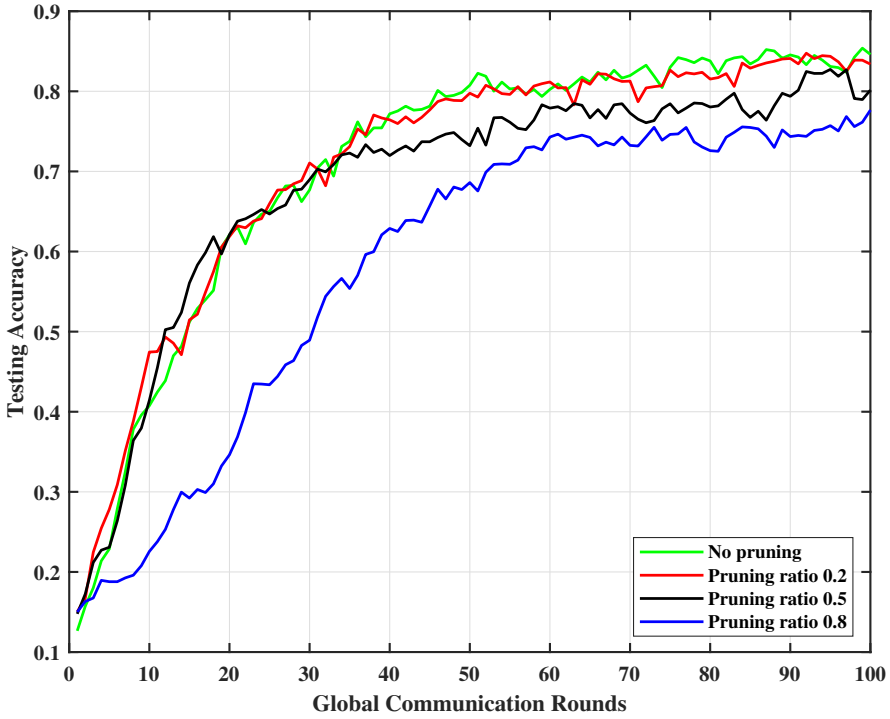


FIGURE 5.5: Testing accuracy on MNIST datasets with different values of pruning ratios. Value of SNR is 12 dB.

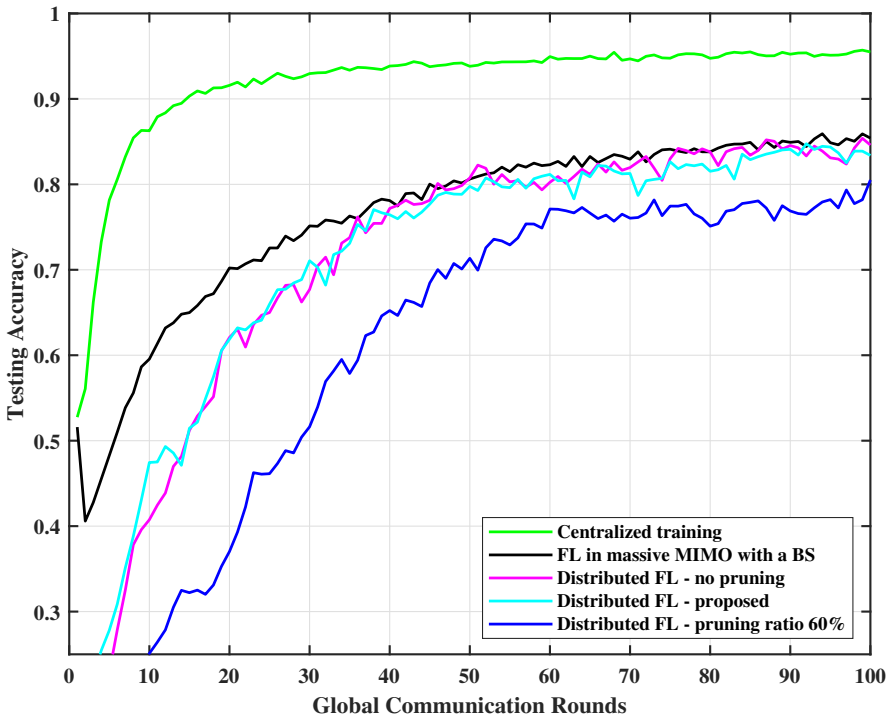


FIGURE 5.6: Comparison of testing accuracy between the proposed FL framework with other baselines.

5.6.2 For different values of SNRs

In Fig. 5.2, testing accuracy with different values of SNR is plotted. Simulations are averaged over 10 instances. It can be seen that with more errors in the D2D links, the

TABLE 5.1: System running time of each FL scheme in seconds.

Schemes	System running time
Centralized training	2988.57
Star-topology FL in mMIMO	5024.52
Proposed FL	3724.28
Distributed FL no pruning	4663.45
Distributed FL equal pruning	4281.77

accuracy is lower and takes longer to converge. With 10 dB SNR, the achieved accuracy is approximately 76% of the accuracy for the error-free links. Green curve stands for the baseline of centralized training, where a single device contains all the training data.

5.6.3 For different pruning scores

In Fig. 5.3, two different pruning metrics namely, magnitude score and synaptic flow conservation (SynFlow) score are considered for the proposed system. It is witnessed that both pruning methods render similar system performance in terms of testing accuracy. However, the system running time incurred by the SynFlow scheme is a little longer. Therefore, for the rest of the simulations, we adopt the magnitude score as the pruning metric.

5.6.4 For non-i.i.d. training datasets

In Fig. 5.4, we vary the non-i.i.d. factors and plot the testing accuracy on MNIST (a) and FMNIST (b) datasets. In both cases, the performance degradation is witnessed as the value of non-i.i.d. factor is increased, that is, higher degree of non-i.i.d.-ness in the training samples.

5.6.5 For different values of pruning ratios

In Fig. 5.5, we illustrate the effect of varying pruning ratios on testing accuracy. As the pruning ratio increases, we observe a decrease in convergence rate and testing accuracy. This trend is attributed to the reduction in model weights associated with higher pruning ratios, resulting in diminished learning capacity of the model.

5.6.6 Pruning ratio and bandwidth optimization

In Fig. 5.6, we compare the proposed FL framework with the following baselines.

- **Centralized training:** One single device contains all training samples.
- **Star-topology FL in mMIMO [24]:** A wireless single-cell multi-device massive multi-input-multi-output (mMIMO) network is considered. The base station (BS) utilized the zero-forcing (ZF) technique to null the intra-cell interference. For a fair comparison, we consider the same local model structure and number of devices as mentioned above.
- **Distributed FL - no pruning:** A fully distributed FL framework is considered and no model pruning is employed.
- **Distributed FL - equal pruning:** Model pruning is utilized in the distributed FL, and each device adopts the same pruning ratio.

From Fig. 5.6, it is observed that the centralized training (green curve) shows the best performance since no communication errors are introduced in this scheme and no model pruning is employed to eliminate redundant model weights. The black curve depicts the mMIMO-integrated FL framework which considers a base station, that is; a star-topology FL. The proposed FL scheme with pruning ratio and bandwidth optimization (cyan curve) achieves comparable testing accuracy to the magenta curve, where no pruning is considered. However, due to the joint optimization of pruning ratio and bandwidth, the training latency required by the proposed framework is shorter than that of the scheme without pruning, as will be shown later in Table 5.1. The blue curve represents the scenario where all devices adopt the same pruning ratio. In this scenario, the dynamic environment of wireless channels is ignored, resulting in the worst performance, with an achieved accuracy of approximately 96% of that of the proposed scheme.

In Table 5.1, we present a summary of the system running times required to complete 100 global communication rounds for the various FL schemes mentioned above. The centralized training scheme exhibits the shortest system running time, as it does not involve wireless communication procedures. Additionally, the system running time required by our proposed FL framework is lower than that of the star-topology FL [24], distributed FL scheme without model pruning, and distributed FL with equal model pruning. Specifically, the training time needed by our system is approximately 88% of that required by the system with equal pruning and 80% of that required by the system without pruning. This is because in our proposed FL scheme, the optimal pruning ratio is calculated for each device based on its wireless channel conditions. It is worth noting that, in the mMIMO-integrated FL scheme, precoding and combining operations occur at BS, thus contributing to a longer system running time.

5.7 Conclusion

In this chapter, we have investigated a FL framework with model pruning in D2D communication networks. Considering the physical location of local devices, dynamic clustering has been employed to guarantee efficient model aggregation. To decrease the training latency, we adopt the model pruning technique. Further, based on the convergence analysis, in order to accelerate the convergence and guarantee learning accuracy, we have employed the KKT conditions to jointly optimize the bandwidth allocation and pruning ratio. Simulation results show that, without sacrificing testing accuracy, the proposed FL algorithm necessitates only 88% of the training time required by the system employing equal pruning and 80% of the training time needed by the system without pruning.

Chapter 6

Conclusion and Future Work

6.1 Summary of Contributions

The evolution of future wireless communication systems calls for the penetration of AI into every aspect of heterogeneous and large-scale networks. Federated Learning, with its distributed training and privacy-preserving nature, is considered as a promising paradigm in future networks to decouple the process of data acquisition and computation and fully exploit the potentials offered by next-generation communication systems. In this thesis, we have comprehensively investigated the feasibility of FL algorithms as well as their variants in wireless networks from the perspectives of system design, mathematical analysis, and practical implementation and simulation, etc. We have highlighted the advantages and potentials provided by this emerging framework. Our contributions can be summarized as follows.

- *mMIMO-integrated FL framework:* In Chapter 3, we started with combining the "AI" and "wireless communication" ingredients by integrating FL algorithm into a single-cell multi-user mMIMO communication network. During each global communication round, model weights were updated at both BS and local user sides. Using ZF and MMSE precoding/combining techniques, we nulled the intra-cell interference to ensure accurate model weight transmission. We proved that our algorithm converged in time via mathematical induction and highlighted how the noise variance caused by imperfect communication links would affect the convergence rate. Notably, based on our analysis of the considered system, we found that the reliability of DL links was more important than that of UL links in terms of guaranteeing successful training and we verified this through simulations on practical datasets and applications. Additionally, we investigated the impact of non-i.i.d. local training datasets, discussing and comparing how our proposed FL variants performed under this data distribution.

- *Communication-efficient FL design:* Motivated by the convergence bound that had been derived in Chapter 3, we found that some communication factors could be varied through the training process to save communication resources. Specifically, we decreased the transmit power at earlier rounds of training to tolerate more "reckless" trials since the weights were not converged. At later rounds, when the weights were close to convergence, we allocated more power to ensure accurate communication and weight updates. We also carefully selected the modulation and quantization schemes to meet the requirement of overall incurred communication overhead. Additionally, we employed different user-selecting policies, such as random, CSI-based, and l_2-norm-based scheduling to further improve the communication efficiency. Motivated by the similar design idea from power variation schem, we proposed a training step variation scheme that targeted at accelerating the training. We demonstrated both the effectiveness and efficiency of these designs by comparing communication cost and system running time in the simulations.

- *mMIMO-integrated FD framework:* Extending our research in Chapter 3, in Chapter 4, we focused on a variant of FL, namely, FD algorithm. We detailed how this variant evolved from FL by exploiting the model heterogeneity offered by the distributed learning paradigm. Specifically, this was achieved by adopting model output logits exclusively as the information exchanged via communication links. We considered a single-cell multi-user mMIMO network, highlighted the novelty of our dual-distillation approach compared with existing literature, and underscored the significance of some system parameter choices in our system, such as degrees of non-i.i.d.-ness, numbers of users, training iterations, BS and user antennas, etc. Utilizing the statistic and stochastic properties of loss functions and SGD, we considered both the local training and distillation processes jointly to derive the convergence bound and justified how our mathematical analysis was different from the approaches in the existing literature. Additionally, we compared the proposed framework with its FL counterpart (Chapter 3) to demonstrate the tremendous communication payload reduction (FD needed only 1% of the FL's communication overhead).

- *Communication channel-aware FD algorithm:* In Chapter 4, we also proposed a communication channel-aware mechanism, which helped the server-side model to intelligently learn the effects imposed by different channel quality on received logits, i.e., be aware of the channel impact. We justified how this design benefited the training process, especially in a rich scattering environment where the CSI among users was usually nonidentical. This design, again, demonstrated the advantages of our dual-distillation FD paradigm by incorporating more flexibility into the server-side manipulation. In the simulation, we showed that this proposed scheme outperformed other methods.

- *Fully distributed FL in D2D networks:* The works in Chapter 3 and 4 considered a star-topology network architecture, i.e., a central server or BS is included in the system during the training process. In order to extend the FL paradigm to a more general communication scenario, in Chapter 5, we investigated the FL algorithm in a serverless D2D wireless network where users could only share information with their near neighbors, namely, a fully distributed FL algorithm. In each global round, to enable efficient communication, we employed the K-means algorithm to perform dynamic clustering of users based on their physical location.

- *Adaptive model pruning-enable FL algorithm:* For the fully distributed FL system model in Chapter 5, we further enhanced the communication efficiency of the framework by incorporating the model pruning technique, resulting in fewer numbers of model weights being transmitted in each global round. Accounting for the error introduced by D2D communication and model pruning, we conducted a mathematical analysis to prove the guaranteed convergence of the proposed system. Further, we modeled the computation and communication latency of the algorithm. To maximize the convergence rate and ensure learning accuracy, we utilized KKT conditions to jointly optimize the pruning ratio and wireless bandwidth allocation under given constraints. The derived optimal solutions provided some interesting insights about the trade-offs between channel condition, pruning ratio, number of quantization bits, bandwidth allocation, and computing capability of local devices, which could utilized as guidelines for the system settings and modifications in possible future work. We also demonstrated the reduced training latency of the proposed adaptive model pruning-enabled distributed FL algorithm via simulations.

6.2 Future Work

6.2.1 FL in mMIMO networks

- *Individual updating policy for enhanced communication efficiency:* In Chapter 3, we propose a communication-efficient FL variant by adjusting the number of local training steps and the value of transmit power globally ($(S_1, S_2, \ldots, S_R)$ and $(P_{UL,1}, P_{UL,2}, \ldots, P_{UL,R})$), that is, all the local users modify the training settings in the same way. We qualitatively verify the feasibility and effectiveness of such modification. However, in practical wireless networks, due to the different distances between users and BS, it is more reasonable to design the individual updating policy for each n^{th} local user ($(S_{n1}, S_{n2}, \ldots, S_{nR})$ and $(P_{UL,n1}, P_{UL,n2}, \ldots, P_{UL,nR})$), based on their specific situation. For example, for a typical global communication round and a given training latency threshold, users that are closer to the BS are able to perform more local training steps, compared with the ones far from the BS.

Similarly, under the given power consumption constraint, users closer to the BS can reduce the transmit power since their transmitted model weights experience less path loss attenuation, compared with their distant counterparts. In this way, the design algorithm is able to provide enhanced communication efficiency.

- *Homomorphic encryption for FL:* In FL, although the local private training dataset $\mathcal{D}_n$ of the n^{th} user is kept locally, the trained model weights $\mathbf{w}_n$ need to be transmitted and shared. In this way, recent studies have revealed that private information can still be divulged to some extent by analyzing the differences of the transmitted model weights [150, 151]. In order to solve this information leakage issue, most of the existing works consider the differential privacy method [152–154], adding artificial noise to the transmitted weights. However, such methods naturally capture a trade-off between privacy and convergence performance in the training process and are not robust to model-inversion attack [155], that is, local users' private datasets will still be leaked to the intruders.

  Recently, the homomorphic encryption (HE) technique has emerged as a promising solution to the privacy leakage problem. HE is a special category of encryption scheme that allows some mathematical operations to be performed directly on the encrypted data (ciphertexts) [156]. Specifically, the encryption function $F_{en}(\cdot)$ and its corresponding decryption function $F_{de}(\cdot)$ are homomorphic with respect to a class of mathematical operations $\mathcal{F}$ (such as addition, multiplication, etc). For any function $f \in \mathcal{F}$ and a input x, we can construct a function g such that

$$f(x) = F_{de}(g(F_{en}(x))). \tag{6.1}$$

  Taking the addition operation as an example, if we have two inputs x_1 and x_2, then, we have the following result

$$x_1 + x_2 = F_{de}(F_{en}(x_1) + F_{en}(x_2)). \tag{6.2}$$

  This property offered by HE inspires us with a new direction for mitigating the privacy leakage in our proposed FL framework. Our main design idea is to include the encryption process before the UL transmission and the decryption process after the DL process, i.e., the encrypted model weights are exchanged via communication links. In this way, the overall composite process in Eqn. (3.9a) is updated as

$$\phi_n\left(\mathbf{v}_t^n, \forall n\right) = F_{de}(\phi_{n,DL} \circ \phi_{av} \circ \phi_{UL}\left(F_{en}(\mathbf{v}_t^n), \forall n\right)), \tag{6.3}$$

  where $F_{en}(\cdot)$ and $F_{de}(\cdot)$ denote the encryption and decryption functions, respectively. In this way, since only the local user has access to the private key required for performing decryption, even though the intruders steal the transmitted information, they will not be able to retrieve the local user's data, thus providing enhanced privacy protection. Furthermore, there is no need for the server to do

the decryption since the aggregation can be directly performed on the received encrypted weights, adding an extra layer of protection. We can also update the expression of overall effective noise $\bar{\mathbf{z}}_t^n$ in Lemma 3.1, derive the corresponding convergence bound, and analyze the impact caused by the integration of HE schemes on the system convergence.

6.2.2 FD in mMIMO networks

- *Strengthened distillation loss function:* For the model distillation process, in Eqn. (4.3), we consider evaluating the loss function between the model output logits and received logits on the public dataset. To some extent, we are treating the received logits as the "pseudo-ground-truth". If the ground-truth labels of the public dataset are available, we can further enhance the distillation loss function. To do that, we need to first exploit the concept of "temperature" of the softmax function and rewrite the Eqn. (2.29) as follows

$$\hat{y}_{ij} = \text{softmax}(\mathbf{z}_i)_{j,T} = \frac{e^{\frac{z_{ij}}{T}}}{\sum_{m=1}^{C} e^{\frac{z_{im}}{T}}}, \tag{6.4}$$

where T denotes the "temperature" of distillation, higher values of T can generate a softer distribution of output class probability [87]. In Eqn. (2.29), T is set to 1. When the ground-truth labels of the public dataset are available, we can strengthen the distillation loss function in Eqn. (4.3) as

$$\begin{aligned} &Q_n\left(\mathbf{w};\mathcal{Z}\right) \\ &= \frac{1}{|\mathcal{Z}|} \sum_{(\mathbf{x}_p,\mathbf{z}_p,\mathbf{y}_p)\in\mathcal{X}\times\mathcal{Z}\times\mathcal{Y}} \beta \underbrace{\psi(T) l_{KL}\left(\text{LSM}\left(F_n(\mathbf{x}_p;\mathbf{w}),T\right),\text{SM}\left(\mathbf{z}_p,T\right)\right)}_{\text{between model output logits and received logits}} \\ &+ (1-\beta) \underbrace{l_{CCE}\left(\text{SM}\left(F_n(\mathbf{x}_p;\mathbf{w}),1\right),\mathbf{y}_p\right)}_{\text{between model output logits and ground-truth}}, \end{aligned} \tag{6.5}$$

where $\text{LSM}\left(\mathbf{z},T\right)$ and $\text{SM}\left(\mathbf{z},T\right)$ denote the log-softmax and softmax functions with temperature T, respectively. $\mathbf{y}_p \in \mathcal{Y}$ denotes the set of ground-truth labels of the public dataset. β is the weighting factor and l_{CCE} is the categorical cross-entropy loss function. The inclusion of the temperature parameter T causes the gradient of the loss function to scale by a specific factor [87]. This scaling behavior, characterized by the function $\psi(T)$, requires further derivation. Note that in Eqn. (6.5), for the second term (cross-entropy loss), the temperature is still set to 1.

Based on this strengthened distillation loss function, we can dynamically adjust the value of the weighting factor β. For instance, when the channel quality is poor, we can decrease the value of β and trust less on the first term in Eqn. (6.5) (KL divergence loss, since it is evaluated on the received logits), but trust more on

the second term (CE loss, which is calculated with the ground-truth). Similarly, we can also design a varying scheme for the temperature parameter T. In this way, models at both the server and local user sides can digest the knowledge more efficiently, potentially enhancing testing performance. Consequently, the convergence analysis must be rederived, and the convexity of the updated $Q_n(\cdot)$ needs to be verified.

- *Multi-task FD:* As explained in Chapter 4, a significant advantage of FD over FL is that FD leverages model heterogeneity by allowing local users to train models with different architectures. In our simulations, we assume users employ various models such as ResNet, ShuffleNet, and MobileNet. Although these models differ in architecture, they all belong to the CNN family and are aimed at image classification tasks.

  A promising future research direction would be to further exploit the model heterogeneity offered by the FD paradigm to support different training tasks within the same system, effectively creating a multi-task federated distillation framework. For instance, we could incorporate a mixture of CNNs and LSTMs, where users with CNNs focus on image classification while those with LSTMs tackle NLP-related tasks. In such a scenario, it would be crucial for the BS to distinguish between different groups of users. Additionally, the corresponding convergence analysis would become more complex due to the inclusion of different loss functions. Addressing these challenges would provide deeper insights into the capabilities of FD and extend its applicability to a wider range of tasks, enhancing the overall efficiency and scalability of distributed learning systems.

- *Enhanced channel-aware mechanism:* In Chapter 4, we propose a group-based FD algorithm that enables the server-side model to learn the impacts of inconsistent channel quality on the received logits. Currently, we consider only two user groups with "good" and "bad" channel quality. In future work, we plan to enhance this channel-aware mechanism by accounting for a greater degree of CSI inconsistency, i.e., more user groups. This modification will allow us to evaluate our proposed algorithm in a more practical wireless network setting.

- *Investigation on distillation steps:* In Theorem 4.5, it can be concluded that the derived convergence bound also depends on the number of distillation steps (S), This insight suggests that dynamically adjusting the loss functions for local training and distillation during the training process can accelerate model convergence. In our future work, we will further investigate this aspect and modify our FD framework accordingly.

6.2.3 Distributed FL with adaptive model pruning

- *Model pruning in the training phase:* In Chapter 5, we consider utilizing the post-training model training technique to eliminate redundant model weights of the trained model, i.e., the pruning takes place after the training process. In reality, the pruning can also be implemented during the training phase and this will grant our framework more flexibility and potentials in terms of exploiting device heterogeneity. Correspondingly, the expression for training latency in Eqn. (5.6) is updated as

$$\hat{l}_{n,t}^{train} = \frac{f(1-\rho_n^t)D_nE_n}{f_n}, \tag{6.6}$$

and the lower bound for the optimal pruning ratio in Eqn. (5.55) can be updated as

$$\rho_j^{t*} \geq \left(1 - \frac{R_{nj}^t l_{th}}{R_{nj}^t \frac{\psi W + f D_n E_n}{f_j} + q_j^t W}\right)^+. \tag{6.7}$$

We could also reformulate the corresponding optimization problem, verify its convexity, and solve it with KKT conditions.

- *Adaptive model pruning with FedDrop scheme:* In [157–159], federated dropout (FedDrop) are studied to improve the communication and computation efficiency of FL. FedDrop is proposed based on the concepts of the dropout scheme in neural networks. Specifically, in each iteration of the FL algorithm, multiple subnets were independently generated from the global model at the server, utilizing heterogeneous dropout rates. Each subnet's dropout rate was adapted to the condition of its assigned channel. FedDrop is a simple method to prevent the model from overfitting and it is used during the training phase. However, during the testing and information exchange phases, the whole model is still transmitted. In our future work, we can combine the FedDrop and post-training model pruning schemes together to enhance the efficiency of the FL algorithm, at both the training and testing phases.

6.2.4 Some other general extensions

- *More comprehensive training tasks:* In the research works presented in this thesis, to verify the feasibility of the proposed algorithms, in the simulations, we mainly consider image classification as the local training task. Actually, in the field of machine/deep learning, there are a lot more interesting and challenging NLP and CV tasks, such as speech recognition and enhancement, object detection, image segmentation, etc. Exploring these tasks could provide a more comprehensive analysis and evaluation of our proposed frameworks. Moreover, just like we combine the "wireless communication" and "AI" aspects in our proposed algorithms, we

can utilize these algorithm to solve some practical wireless communication problems, such as wireless edge caching, interference alignment, channel estimation, etc. This interdisciplinary approach could lead to innovative solutions and further advancements in both fields.

- *More flexible aggregation scheme:* In Chapter 3, in order to highlight the importance of the quantization scheme, we reproduce the results in [40], where the authors design a quantization scheme adaptive to the channel status, model weight importance, etc. This inspires us in the direction to propose a more flexible and intelligent aggregation scheme on the server side, such as adjusting the weighting factor α_n among users and excluding certain users whose transmitted weights might lead to slow convergence. Such modifications can work jointly with the scheduling policies proposed in Chapter 3 to further improve the communication efficiency and accelerate the convergence.

- *Hierarchical FL in multi-cell networks:* In Chapter 3 and 4, our focus was on maintaining consistency with the star-topology FL architecture, thus considering a single-cell scenario involving one BS and several users. However, modern wireless communication systems typically operate in multi-cell environments. Therefore, it might be an interesting research direction to extend and modify the current works to a multi-cell scenario. In this case, the convergence analysis will be more challenging due to the necessity of accounting for both intra-cell and inter-cell interference. Such an extension could lead to the development of a three-tier hierarchical FL architecture, where coordination among different BSs becomes a critical design aspect. Exploring these dynamics in multi-cell networks will provide deeper insights and potentially enhance the robustness and scalability of FL in practical applications.

Bibliography

[1] Ericsson, "Ericsson mobility report," 2023. [Accessed: May 18, 2024]. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report.

[2] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6g: Ai empowered wireless networks," *IEEE communications magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[3] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6g wireless communication systems: Applications, requirements, technologies, challenges, and research directions," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 957–975, 2020.

[4] A. Y. Chua, A. Pal, and S. Banerjee, "Ai-enabled investment advice: Will users buy it?," *Computers in Human Behavior*, vol. 138, p. 107481, 2023.

[5] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*. prentice hall professional technical reference, 2002.

[6] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[7] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[8] Y. E. Wang, G.-Y. Wei, and D. Brooks, "Benchmarking tpu, gpu, and cpu platforms for deep learning," *arXiv preprint arXiv:1907.10701*, 2019.

[9] S. Sagiroglu and D. Sinanc, "Big data: A review," in *Proc. 2013 international conference on collaboration technologies and systems (CTS)*, pp. 42–47, IEEE, 2013.

[10] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving,

M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[11] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[12] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[13] L. Zheng and D. N. C. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels," *IEEE Transactions on information theory*, vol. 49, no. 5, pp. 1073–1096, 2003.

[14] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor, *MIMO wireless communications*. Cambridge university press, 2007.

[15] B. Sklar, *Digital communications: fundamentals and applications*. Pearson, 2021.

[16] L. Bai, J. Choi, and Q. Yu, *Low complexity MIMO receivers*. Springer, 2014.

[17] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell labs technical journal*, vol. 1, no. 2, pp. 41–59, 1996.

[18] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive mimo: Benefits and challenges," *IEEE journal of selected topics in signal processing*, vol. 8, no. 5, pp. 742–758, 2014.

[19] T. L. Marzetta, "Massive mimo: an introduction," *Bell Labs Technical Journal*, vol. 20, pp. 11–22, 2015.

[20] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," *Advances in neural information processing systems*, vol. 30, 2017.

[21] K. O'shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[23] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[24] Y. Mu, N. Garg, and T. Ratnarajah, "Federated learning in massive mimo 6g networks: Convergence analysis and communication-efficient design," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 6, pp. 4220–4234, 2022.

[25] Y. Mu, N. Garg, and T. Ratnarajah, "Communication-efficient federated learning for massive mimo systems," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 578–583, IEEE, 2022.

[26] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[27] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3347–3366, 2021.

[28] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, "Hybridalpha: An efficient approach for privacy-preserving federated learning," in *Proc. 12th ACM workshop on artificial intelligence and security*, pp. 13–23, 2019.

[29] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," in *Proc. 12th ACM workshop on artificial intelligence and security*, pp. 1–11, 2019.

[30] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang, "Towards efficient and privacy-preserving federated deep learning," in *Proc. IEEE international conference on communications (ICC)*, pp. 1–6, IEEE, 2019.

[31] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[32] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[33] Y. Liu, J. Peng, J. Kang, A. M. Iliyasu, D. Niyato, and A. A. Abd El-Latif, "A secure federated learning framework for 5g networks," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 24–31, 2020.

[34] M. Isaksson and K. Norrman, "Secure federated learning in 5g mobile networks," in *Proc. IEEE Global Communications Conference*, pp. 1–6, IEEE, 2020.

[35] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6g communications: Challenges, methods, and future directions," *China Communications*, vol. 17, no. 9, pp. 105–118, 2020.

[36] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6g: Applications, challenges, and opportunities," *Engineering*, vol. 8, pp. 33–41, 2022.

[37] H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial-intelligence-enabled intelligent 6g networks," *IEEE network*, vol. 34, no. 6, pp. 272–280, 2020.

[38] Y. Xiao, G. Shi, and M. Krunz, "Towards ubiquitous ai in 6g with federated learning," *arXiv preprint arXiv:2004.13563*, 2020.

[39] J. Konecnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, vol. 8, 2016.

[40] S. Zheng, C. Shen, and X. Chen, "Design and analysis of uplink and downlink communications for federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2150–2167, 2020.

[41] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "Federated learning with quantization constraints," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8851–8855, IEEE, 2020.

[42] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," *arXiv preprint arXiv:1704.05021*, 2017.

[43] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," *Advances in neural information processing systems*, vol. 30, 2017.

[44] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3400–3413, 2019.

[45] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2021.

[46] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE conference on computer communications*, pp. 1387–1395, IEEE, 2019.

[47] C. T. Dinh, N. H. Tran, M. N. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 398–409, 2020.

[48] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, 2020.

[49] X. Mo and J. Xu, "Energy-efficient federated edge learning with joint communication and computation design," *Journal of Communications and Information Networks*, vol. 6, no. 2, pp. 110–124, 2021.

[50] Y. Liu, Y. Zhu, and J. James, "Resource-constrained federated edge learning with heterogeneous data: Formulation and analysis," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3166–3178, 2021.

[51] H. H. Yang, A. Arafa, T. Q. Quek, and H. V. Poor, "Age-based scheduling policy for federated learning in mobile edge networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8743–8747, IEEE, 2020.

[52] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE transactions on communications*, vol. 68, no. 1, pp. 317–333, 2019.

[53] W. Shi, S. Zhou, and Z. Niu, "Device scheduling with fast convergence for wireless federated learning," in *Proc. IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2020.

[54] H. Yang, J. Zhao, Z. Xiong, K.-Y. Lam, S. Sun, and L. Xiao, "Privacy-preserving federated learning for uav-enabled networks: Learning-based joint scheduling and resource management," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 3144–3159, 2021.

[55] W. Y. B. Lim, S. Garg, Z. Xiong, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Uav-assisted communication efficient federated learning in the era of the artificial intelligence of things," *IEEE network*, vol. 35, no. 5, pp. 188–195, 2021.

[56] X. Wei and C. Shen, "Federated learning over noisy channels: Convergence analysis and design examples," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 1253–1268, 2022.

[57] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Transactions on Wireless Communications*, vol. 19, pp. 3546–3557, Feb. 2020.

[58] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 491–506, 2019.

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

[60] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE transactions on wireless communications*, vol. 19, no. 3, pp. 2022–2035, 2020.

[61] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, 2019.

[62] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *Proc. International conference on artificial intelligence and statistics*, pp. 2021–2031, PMLR, 2020.

[63] Y.-S. Jeon, M. M. Amiri, J. Li, and H. V. Poor, "A compressive sensing approach for federated learning over massive mimo communication systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1990–2004, 2020.

[64] M. Duan, D. Liu, X. Chen, R. Liu, Y. Tan, and L. Liang, "Self-balancing federated learning with global imbalanced data in mobile systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 59–71, 2020.

[65] A. K. Papazafeiropoulos, H. Q. Ngo, and T. Ratnarajah, "Performance of massive mimo uplink with zero-forcing receivers under delayed channels," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3158–3169, 2016.

[66] A. K. Papazafeiropoulos and T. Ratnarajah, "Deterministic equivalent performance analysis of time-varying massive mimo systems," *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5795–5809, 2015.

[67] R. M. Neal and R. M. Neal, "Priors for infinite networks," *Bayesian learning for neural networks*, pp. 29–53, 1996.

[68] I. Goodfellow, "Deep learning," 2016.

[69] H. Yu, S. Yang, and S. Zhu, "Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proc. AAAI conference on artificial intelligence*, pp. 5693–5700, 2019.

[70] J. Sun, A. Li, B. Wang, H. Yang, H. Li, and Y. Chen, "Soteria: Provable defense against privacy leakage in federated learning from representation perspective," in *Proc. IEEE/CVF conference on computer vision and pattern recognition*, pp. 9311–9319, 2021.

[71] Y. Zhang, M. J. Wainwright, and J. C. Duchi, "Communication-efficient algorithms for statistical optimization," *Advances in neural information processing systems*, vol. 25, 2012.

[72] X. Liu, T. Ratnarajah, M. Sellathurai, and Y. C. Eldar, "Adaptive model pruning and personalization for federated learning over wireless networks," *arXiv preprint arXiv:2305.09042*, 2024.

[73] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Transactions on Communications*, vol. 68, pp. 317–333, Jan. 2020.

[74] A. Li, J. Sun, P. Li, Y. Pu, H. Li, and Y. Chen, "Hermes: An efficient federated learning framework for heterogeneous mobile clients," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pp. 420–437, Oct. 2021.

[75] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[76] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *Proc. international joint conference on neural networks (IJCNN)*, pp. 2921–2926, IEEE, 2017.

[77] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[78] Z. Qin, G. Y. Li, and H. Ye, "Federated learning and wireless communications," *IEEE Wireless Communications*, vol. 28, no. 5, pp. 134–140, 2021.

[79] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 375–388, 2020.

[80] M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive mimo detection techniques: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3109–3132, 2019.

[81] R. Chataut and R. Akl, "Massive mimo systems for 5g and beyond networks—overview, recent trends, challenges, and future research direction," *Sensors*, vol. 20, no. 10, p. 2753, 2020.

[82] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive mimo for next generation wireless systems," *IEEE communications magazine*, vol. 52, no. 2, pp. 186–195, 2014.

[83] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[84] A. Agrawal, A. Ali, S. Boyd, *et al.*, "Minimum-distortion embedding," *Foundations and Trends® in Machine Learning*, vol. 14, no. 3, pp. 211–378, 2021.

[85] Y. Mu, N. Garg, and T. Ratnarajah, "Federated distillation in massive mimo networks: Dynamic training, convergence analysis and communication channel-aware learning," *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2024.

[86] Y. Mu, N. Garg, and T. Ratnarajah, "Communication channel-aware federated distillation for massive mimo systems," in *Proc. IEEE 24th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 436–440, IEEE, 2023.

[87] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[88] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: federated distillation and augmentation under non-iid private data," *arXiv preprint arXiv:1811.11479*, 2018.

[89] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Transactions on wireless communications*, vol. 9, no. 11, pp. 3590–3600, 2010.

[90] E. Björnson, J. Hoydis, M. Kountouris, and M. Debbah, "Massive mimo systems with non-ideal hardware: Energy efficiency, estimation, and capacity limits," *IEEE Transactions on information theory*, vol. 60, no. 11, pp. 7112–7139, 2014.

[91] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser mimo systems," *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1436–1449, 2013.

[92] F. Sattler, T. Korjakow, R. Rischke, and W. Samek, "Fedaux: Leveraging unlabeled auxiliary data in federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[93] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2351–2363, 2020.

[94] Y. Mi, Y. Mu, S. Zhou, and J. Guan, "Fedmdr: Federated model distillation with robust aggregation," in *Proc. Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pp. 18–32, Springer, 2021.

[95] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[96] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. International Conference on Machine Learning*, pp. 12878–12889, PMLR, 2021.

[97] X. Li, B. Chen, and W. Lu, "Feddkd: Federated learning with decentralized knowledge distillation," *Applied Intelligence*, pp. 1–17, 2023.

[98] C. Wang, G. Yang, G. Papanastasiou, H. Zhang, J. J. Rodrigues, and V. H. C. de Albuquerque, "Industrial cyber-physical systems-based cloud iot edge for federated heterogeneous distillation," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5511–5521, 2020.

[99] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature Communications*, vol. 13, no. 1, pp. 1–8, 2022.

[100] S. Oh, J. Park, E. Jeong, H. Kim, M. Bennis, and S.-L. Kim, "Mix2fld: Downlink federated learning after uplink federated distillation with two-way mixup," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2211–2215, 2020.

[101] J.-H. Ahn, O. Simeone, and J. Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," in *Proc. IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6, IEEE, 2019.

[102] J.-H. Ahn, O. Simeone, and J. Kang, "Cooperative learning via federated distillation over fading channels," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8856–8860, IEEE, 2020.

[103] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Communication-efficient federated distillation with active data sampling," in *Proc. IEEE International Conference on Communications*, pp. 201–206, IEEE, 2022.

[104] S. Itahara, T. Nishio, Y. Koda, M. Morikura, and K. Yamamoto, "Distillation-based semi-supervised federated learning for communication-efficient collaborative

training with non-iid private data," *IEEE Transactions on Mobile Computing*, vol. 22, no. 1, pp. 191–205, 2021.

[105] F. Sattler, A. Marban, R. Rischke, and W. Samek, "Cfd: Communication-efficient federated distillation via soft-label quantization and delta coding," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2025–2038, 2021.

[106] X. Qu, J. Wang, and J. Xiao, "Quantization and knowledge distillation for efficient federated learning on edge devices," in *Proc. IEEE 22nd International Conference on High Performance Computing and Communications*, pp. 967–972, IEEE, 2020.

[107] J. Park, S. Samarakoon, A. Elgabli, J. Kim, M. Bennis, S.-L. Kim, and M. Debbah, "Communication-efficient and distributed learning over wireless networks: Principles and applications," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 796–819, 2021.

[108] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3579–3605, 2021.

[109] S. Ioffe, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[110] J. Pennington, S. Schoenholz, and S. Ganguli, "The emergence of spectral universality in deep networks," in *International Conference on Artificial Intelligence and Statistics*, pp. 1924–1932, PMLR, 2018.

[111] J. Howard, "Imagenette," 2022. [Accessed: December 15, 2023]. [Online]. Available: https://github.com/fastai/imagenette.

[112] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," *Tech Report*, 2009.

[113] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th international conference on artificial intelligence and statistics*, pp. 215–223, JMLR Workshop and Conference Proceedings, 2011.

[114] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE conference on computer vision and pattern recognition*, pp. 248–255, IEEE, 2009.

[115] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[116] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[117] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[118] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[119] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018.

[120] Y. Mu, X. Liu, and T. Ratnarajah, "Optimizing efficiency through adaptive model pruning in fully distributed federated learning," in *Proc. IEEE 25th international workshop on signal processing advances in wireless communications (SPAWC)*, IEEE, 2024. accepted.

[121] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[122] D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, and W.-J. Hwang, "Federated learning for smart healthcare: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–37, 2022.

[123] T. Yu, T. Li, Y. Sun, S. Nanda, V. Smith, V. Sekar, and S. Seshan, "Learning context-aware policies from multiple smart homes via federated multi-task learning," in *Proc. IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 104–115, IEEE, 2020.

[124] F. Liu, X. Wu, S. Ge, W. Fan, and Y. Zou, "Federated learning for vision-and-language grounding problems," in *Proc. AAAI Conference on Artificial Intelligence*, pp. 11572–11579, 2020.

[125] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 86–92, 2014.

[126] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[127] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," *IEEE Transactions on Wireless Communications*, vol. 20, pp. 3643–3658, Jun. 2021.

[128] E. Georgatos, C. Mavrokefalidis, and K. Berberidis, "Fully distributed federated learning with efficient local cooperations," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.

[129] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4289–4301, 2022.

[130] K. Cheng, F. Guo, and M. Peng, "An efficient distributed machine learning framework in wireless d2d networks: Convergence analysis and system implementation," *IEEE Transactions on Vehicular Technology*, 2023.

[131] H. Xing, O. Simeone, and S. Bi, "Decentralized federated learning via sgd over wireless d2d networks," in *Proc. IEEE 21st international workshop on signal processing advances in wireless communications (SPAWC)*, pp. 1–5, IEEE, 2020.

[132] H. Xing, O. Simeone, and S. Bi, "Federated learning over wireless device-to-device networks: Algorithms and convergence analysis," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3723–3741, 2021.

[133] Z. Jiang, G. Yu, Y. Cai, and Y. Jiang, "Decentralized edge learning via unreliable device-to-device communications," *IEEE Transactions on Wireless Communications*, vol. 21, no. 11, pp. 9041–9055, 2022.

[134] C. Li, G. Li, and P. K. Varshney, "Federated learning with soft clustering," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7773–7782, 2021.

[135] F. Sattler, K.-R. Müller, 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, 2020.

[136] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19586–19597, 2020.

[137] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *Proc. International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, IEEE, 2020.

[138] S. Vahidian, M. Morafah, and B. Lin, "Personalized federated learning by structured and unstructured pruning under data heterogeneity," in *Proc. IEEE 41st international conference on distributed computing systems workshops (ICDCSW)*, pp. 27–34, IEEE, 2021.

[139] P. Prakash, J. Ding, R. Chen, X. Qin, M. Shu, Q. Cui, Y. Guo, and M. Pan, "Iot device friendly and communication-efficient federated learning via joint model pruning and quantization," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13638–13650, 2022.

[140] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[141] H. Xu, M. Chen, Z. Meng, Y. Xu, L. Wang, and C. Qiao, "Decentralized machine learning through experience-driven method in edge networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 2, pp. 515–531, 2022.

[142] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[143] D. Wen, M. Bennis, and K. Huang, "Joint parameter-and-bandwidth allocation for improving the efficiency of partitioned edge learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 8272–8286, 2020.

[144] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *arXiv preprint arXiv:1611.06440*, 2016.

[145] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in neural information processing systems*, vol. 28, 2015.

[146] H. Tanaka, D. Kunin, D. L. Yamins, and S. Ganguli, "Pruning neural networks without any data by iteratively conserving synaptic flow," *Advances in neural information processing systems*, vol. 33, pp. 6377–6389, 2020.

[147] S. Liu, G. Yu, R. Yin, J. Yuan, L. Shen, and C. Liu, "Joint model pruning and device selection for communication-efficient federated edge learning," *IEEE Transactions on Communications*, vol. 70, no. 1, pp. 231–244, 2021.

[148] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified sgd with memory," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[149] G. Long, M. Xie, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning: Clients clustering for better personalization," *World Wide Web*, vol. 26, pp. 481–500, Jun. 2022.

[150] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. IEEE conference on computer communications*, pp. 2512–2520, IEEE, 2019.

[151] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE transactions on information forensics and security*, vol. 15, pp. 3454–3469, 2020.

[152] S. Wang, L. Huang, Y. Nie, X. Zhang, P. Wang, H. Xu, and W. Yang, "Local differential private data aggregation for discrete distribution estimation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 9, pp. 2046–2059, 2019.

[153] N. Wu, F. Farokhi, D. Smith, and M. A. Kaafar, "The value of collaboration in convex machine learning with differential privacy," in *Proc. IEEE Symposium on Security and Privacy (SP)*, pp. 304–317, IEEE, 2020.

[154] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. C. Hui, H. Shin, J. Shin, and G. Yu, "Collecting and analyzing multidimensional data with local differential privacy," in *Proc. IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 638–649, IEEE, 2019.

[155] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC conference on computer and communications security*, pp. 1322–1333, 2015.

[156] F. Boemer, Y. Lao, R. Cammarota, and C. Wierzynski, "ngraph-he: a graph compiler for deep learning on homomorphically encrypted data," in *Proc. 16th ACM international conference on computing frontiers*, pp. 3–13, 2019.

[157] D. Wen, K.-J. Jeon, and K. Huang, "Federated dropout—a simple approach for enabling federated learning on resource constrained devices," *IEEE wireless communications letters*, vol. 11, no. 5, pp. 923–927, 2022.

[158] G. Cheng, Z. Charles, Z. Garrett, and K. Rush, "Does federated dropout actually work?," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3387–3395, 2022.

[159] B. Nader, H. Jiahui, Z. Hui, and L. Xin, "Adaptive federated dropout: Improving communication efficiency and generalization for federated learning," *arXiv preprint arXiv:2011.04050*, 2020.