Efficient Federated Learning over Wireless Networks

by

Zhixiong Chen

Doctor of Philosophy

School of Electronic Engineering and Computer Science
Queen Mary University of London
United Kingdom

March 2024

TO MY FAMILY

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my primary supervisor, Prof. Arumugam Nallanathan, for his careful guidance and continuous support of my PhD research. Prof. Nallanathan is kind, knowledgeable, and considerate. He encourages me to investigate promising and challenging problems. He also introduced me to numerous excellent peers and professors in my research area. His friendly personality, wide knowledge, and hard-working attitude have had a profound effect on my PhD study and future academic life.

Secondly, I would like to express my thanks to my second supervisor, Dr. Atm Alam and my independent accessor, Dr. Mona Jaber, for their great support of my PhD progression vivas. Meanwhile, I especially thank all my collaborators, Dr. Wenqiang Yi, Prof. Geoffrey Ye Li, Prof. Jonathon Chambers, Prof. Yuanwei Liu, and Prof. Hyundong Shin, for their insightful suggestions and comments on my research works. From these excellent collaborators, I have learned not only advanced techniques but also crucial characteristics of becoming a qualified researcher.

I would like to thank Bingjie Zhu, Chunfeng Xie, Haocheng Li, Kangda Zhi, Na Yan, Na Xue, Tuo Wu, Yuqin Liu, Zijian Zheng, and all my colleagues and friends in School of Electronic Engineering and Computer Science at Queen Mary University of London, for their constant support and encouragement. I have truly cherished the wonderful memories from my PhD journey and studies.

Last but not least, I sincerely thank my beloved parents and elder brother for their relentless support. My deepest gratitude goes to my wife, R. Liu, for her unconditional understanding and trust, which are my strongest motivation to complete this hard mission and pursue a better life.

Abstract

Federated learning (FL) is a promising distributed learning paradigm for protecting data privacy. In FL, edge devices collaboratively train machine learning (ML) models under the orchestration of a parameter server (PS), which only requires exchanging local learning models/gradients among devices and the PS instead of local private data. However, implementing FL in real-world wireless networks faces several challenges, e.g., data heterogeneity, device heterogeneity, limited wireless resources, and unreliable wireless channels. This thesis presents four original contributions to address these challenges by jointly designing the learning mechanism and wireless networks.

Firstly, a joint representativity and latency-aware device scheduling scheme is proposed to address the limited wireless resources for FL. Specifically, we theoretically revealed that the learning performance is degraded by the difference between the aggregated gradient of scheduled devices and the full participation gradient. Based on this, the proposed scheme aims to find a subset of representative devices and their corresponding pre-device stepsizes to approximate the full participation gradient while capturing the trade-off between learning performance and latency for FL. Compared to existing device scheduling algorithms, the proposed representativity-aware device scheduling algorithm improves 6.7% and 4.02% accuracies on two typical datasets under heterogeneous local data distributions, i.e., MNIST and CIFAR-10, respectively. In addition, the proposed latency- and representativity-aware scheduling algorithm saves over 16% and 12% training time for MNIST and CIFAR-10 datasets than the scheduling algorithms based on either latency or representativity individually.

Secondly, a novel knowledge-aided FL (KFL) framework is proposed to address the data heterogeneity and reduce the communication costs, which aggregates light high-level data features, namely knowledge, in the per-round learning process. This framework allows devices to design their machine-learning models independently and reduces the communication overhead in the training process. We theoretically revealed that allocating more resources in the early rounds achieves better learning performance when the total available resources are fixed during the entire learning course. Based on this, a joint device scheduling, bandwidth allocation, and power control approach is developed to optimize the learning performance of FL under limited energy budgets of devices. Experimental results on two typical datasets (i.e., MNIST and CIFAR-10) under highly heterogeneous local data distributions show that the proposed KFL is capable of reducing over 99% communication overhead while achieving better learning performance than the conventional model aggregation-based algorithms. In addition, the proposed device scheduling algorithm converges faster than the benchmark scheduling schemes.

Thirdly, a novel FL framework, namely FL with gradient recycling (FL-GR), is proposed to tackle the negative effects of unreliable wireless channels and constrained resources on FL. FL-GR recycles the historical gradients of unscheduled and transmission-failure devices to improve the learning performance of FL. We theoretically revealed that minimizing the average square of local gradients' staleness (AS-GS) helps improve learning performance. Based on this, a joint device scheduling, resource allocation and power control approach is proposed to minimize the AS-GS for global loss minimization. Compared to the FL algorithm without gradient recycling, FL-GR achieves over 4% accuracy improvement. In addition, the proposed device scheduling algorithm outperforms the benchmarks in convergence speed and test accuracy.

Finally, a novel adaptive model pruning-based FL (AMP-FL) framework is proposed to address the device heterogeneity, where the edge server dynamically generates submodels by pruning the global model for devices' local training to adapt their heterogeneous computation capabilities and time-varying channel conditions. We introduced an age of information (AoI) metric to characterize the staleness of local gradients and theoretically analyzed the convergence behaviour of AMP-FL. The convergence bound shows that scheduling devices with large AoI of gradients and pruning the model regions with small AoI for devices can improve learning performance. Inspired by this, a joint device scheduling, model pruning, and resource allocation scheme is developed to enhance the learning performance of FL. Experimental results show that the proposed AMP-FL is capable of achieving 1.9x and 1.6x speed up for FL on MNIST and CIFAR-10 datasets in comparison with the FL schemes with homogeneous model settings.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

6G	sixth-Generation
AMP-FL	Adaptive model pruning-based federated learning
AoI	Age of information
APFL	Adaptive personalized FL
AR	Augmented reality
AS-GS	Average square of local gradients' staleness
CNN	Convolutional neural network
CPU	Central processing unit
C-scheduling	Channel gain-aware device scheduling
DNN	Deep neural network
E-MBG	Estimating by mini-batch gradient
E-PG	Estimate by past gradient information
FC	Fully connected
FDMA	Frequency division multiple access
FedAvg	Federated averaging
FL	Federated learning
FL-GR	Federated learning with gradient recycling
FLOPs	Float-point operations
FP-SG	Full-participation stochastic gradient

GI-Scheduling	Gradient importance-aware scheduling
IoT	Internet-of-Things
KD	Knowledge distillation
KFL	Knowledge-aided federated learning
L&R-aware	latency- and representativity-aware device scheduling
Max-GNS	Maximum gradient norm scheduling
MC	Model compensation
ML	Machine learning
MLP	Multi-layer perceptron
NN	Neural network
NNs	Neural networks
NP-Hard	Non-deterministic polynomial-time hard
non-IID	Non-independent and identically distributed
OFDMA	Orthogonal frequency division multiple access
PC	Power-of-Choice scheduling
PR-scheduling	Pruning ratio minimization-aware device scheduling
RB	Resource block
RBs	Resource block
RHS	Right-hand-side

RS	Random scheduling
SGD	Stochastic gradient descent
SINR	Signal-to-interference-plus-noise ratio
STP-Scheduling	Successful transmission probability-aware scheduling
SVM	Support vector machines
VR	Virtual reality
W/GR	Without gradient recycling

List of Notation

$\mathcal{K}$	Set of devices
K	Size of $\mathcal{K}$
$\mathcal{D}_k$	Local dataset of device k;
D_k	Size of $\mathcal{D}_k$
$\mathcal{D}$	Overall dataset in the system
D	Size of $\mathcal{D}$
η;	Learning rate;
τ	Local iteration number
$\boldsymbol{S}_t$;	Scheduling policy in round t, i.e., the set of scheduled devices
$\alpha_{k,t}$	Scheduling indicator of device k in round t
$\widetilde{\boldsymbol{g}}_{k,t}$	Local gradient of device k in round t
$\widetilde{\boldsymbol{g}}_t$	Aggregated gradient of devices in $\boldsymbol{S}_t$
$\boldsymbol{g}_t$	Aggregated gradient of all devices in $\mathcal{K}$
L_b	Local batch size
f_k	CPU frequency of device k
p_k	Transmit power of device k
Q;	Number of elements of each local gradient;
q	Quantized bits of each gradient element
$\boldsymbol{\theta}_t$	The proportion of B allocated to devices in round t
$T_{k,t}^{\mathrm{L}}$	Computation time for device k in round t
$T_{k,t}^{\mathrm{C}}$	Communication time for device k in round t

$\mathcal{C}$	Set of data classes
C	Size of $\mathcal{C}$
$\mathcal{D}_{k,c}$	Local dataset of c class
$D_{k,c}$	Size of $\mathcal{D}_{k,c}$
$\boldsymbol{w}_k$	Local model
$\boldsymbol{u}_k$	Local feature extractor of device k
$\boldsymbol{v}_k$	Local predictor of device k
$\boldsymbol{W}$	All local models
$F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$	Local empirical loss function of device k
$F(\boldsymbol{W})$	Global empirical loss function
η_u	Learning rate for feature extractor
η_v	Learning rate for predictor
$L_k(\boldsymbol{u}_k)$	Local knowledge loss function
λ	Knowledge loss weight
$\mathbf{\Omega}_{k,c};$	Device k's knowledge about class c
$\mathbf{\Omega}_k$	Device k's knowledge for all classes
$\mathbf{\Omega}_c$	Global knowledge about class c
$\mathbf{\Omega}$	Global knowledge about all classes
τ	Local iteration number
C_k	Computation workload of one data sample at device k
$p_{k,t}$	Transmit power of device k in round t
$p_{k,\max}$	Maximum transmit power of device k
Q	Data size of local knowledge
E_k	Total energy budget of device k
$\mathcal{T}_{\max}$	Maximum completion time for each round

$\boldsymbol{w}_{k,t}^{(l)}$	Local model of device k in the l-th iteration of round t
$\boldsymbol{w}_t$	Global model in round t
$F_k(\boldsymbol{w})$	Local loss function of device k
$F(\boldsymbol{w})$	Global loss function
λ	Local iteration number
$\boldsymbol{G}_{k,t}$	Latest successfully transmitted gradient of device k in round t;
$\boldsymbol{p}_t$	Power control policy of devices in round t
C_k	Computation workload of one data sample at device k
Q	Data size of local gradient
$E_{k,\max}$	Energy constraint of device k
$\alpha_{k,t}$	Scheduling indicator of device k in round t

$\mathcal{R}$	Set of resource blocks
R	Size of $\mathcal{R}$
$\mathcal{I}$	Set of global model regions
I	Size of $\mathcal{I}$
$\boldsymbol{w}_t^{(i)}$	The i-th region of global model
$\boldsymbol{w}_{k,t,l}^{(i)}$	The i-th region of local model of device k in the l-th iteration of round t
λ	Local iteration number
$\boldsymbol{G}_{k,t}^{(i)}$	Latest received gradient at the edge server from device k's region i
$\tilde{\boldsymbol{g}}_{k,t}^{(i)}$	Stochastic gradient of i-th region of device k
$\mathcal{C}$	Computation load of one data sample
$\mathcal{Q}$	Number of global model parameters
$z_{k,t}^{(r)}$	Allocation indicator of RB r to device k
$\boldsymbol{Z}_t$	RB allocation policy for all devices
$m_{k,t}^{(i)}$	Pruning mask of i-th region of device k in round t
$\boldsymbol{m}_{k,t}$	$\boldsymbol{m}_{k,t} = \{m_{k,t}^{(i)} : \forall i \in \mathcal{I}\}$
$\beta_{k,t}$	Pruning ratio of device k in round t

Chapter 1

Introduction

1.1 Overview

In this section, the research motivation of federated learning (FL) is first introduced. Then, a typical FL framework is illustrated, which is the basis of our developed FL approaches. Finally, the research challenges of FL in practical wireless networks are presented to inspire the research works in the thesis.

1.1.1 Motivation of Federated Learning

The explosive growth of data generated at edge devices motivated deploying advanced machine learning (ML) techniques in sixth-generation (6G) wireless networks to exploit the data for serving diverse applications, e.g., autonomous driving, intelligent industry, augmented reality/virtual reality (AR/VR), and Internet-of-Things (IoT) applications [1]. The conventional centralized ML approach requires centralizing the raw user data on a single data center or cloud, which is inapplicable to support those emerging 6G applications due to high communication costs and privacy concerns [2–4]. In addition, in recent years, information privacy policies and laws such as GDPR [5] and CCPA [6] stipulate the sharing of data between companies to protect personal data from being abused.

To address the problems in centralized ML, wireless FL becomes a promising solution since it enables 6G devices to learn a global shared model while preserving data locally. In wireless FL, a parameter server orchestrates multiple devices via wireless channels to engage in the training process that repeatedly performs the alternative optimization process of device-local training and server-model aggregation. Instead of transmitting raw user data, wireless FL only shares the local model parameters of users. This unique property reduces the wireless communication load and simultaneously protects the users' privacy.

1.1.2 A Typical Federated Learning Framework

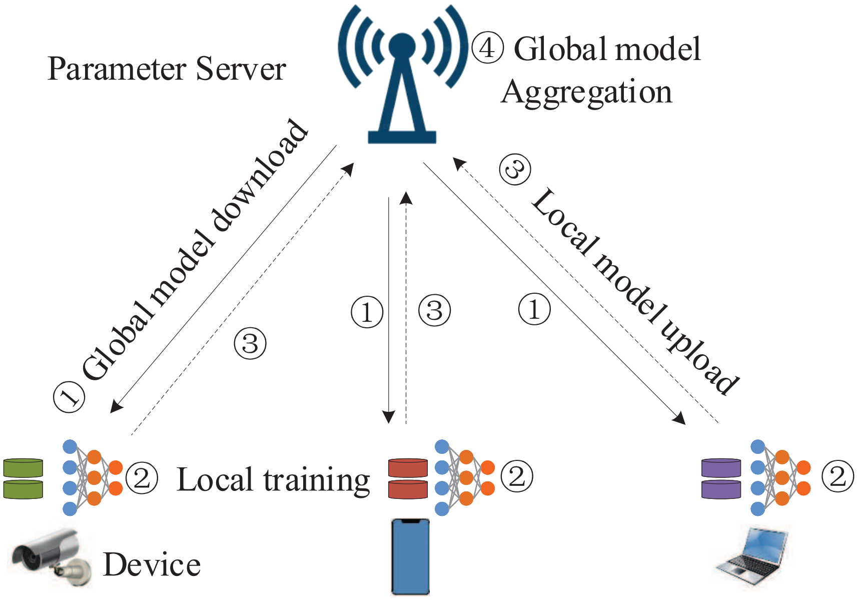


Figure 1.1: The framework of FedAvg.

In 2016, the authors in [7] proposed the first FL algorithm, namely Federated Averaging (FedAvg), in which a server coordinates multiple devices to learn a global shared model. The learning process involves periodically exchanging model parameters between devices and the server without exposing the raw user data, as shown in Fig. 1.1. The learning process of FL is to repeat the following steps until the model is converged.

- Step 1: Device selection and global model broadcast: The server selects a set of devices meeting the eligibility requirements to participate the current-round training process. After device selection, the server broadcast the latest global model to the selected devices.

- Step 2: Local training: After receiving the global model, each selected device computes a local model update by executing the training program, which might for example run stochastic gradient descent (SGD) on its local dataset.

- Step 3: Local model update upload: After the local training process, each selected device uploads it local update to the server through wireless channels.

- Step 4: Global model aggregation: The server aggregates the local updates from all the selected devices by performing a weighted average of them. Then, the global model is updated based on the aggregated model update.

The above FL framework involves different components related to communication and computation, i.e., device selection, local computation, communication between devices and the server, and the global model update. In practical wireless networks, implementing FL faces several challenges, e.g., data heterogeneity, device heterogeneity, scarce wireless resources, and poor computation capabilities of devices. These challenges may restrict the performance of the above FL approach. Existing research works focused on jointly optimizing different components of FL to address these challenges.

1.1.3 Research Challenges

In practical wireless networks, implementing FL confronts several challenges. In the following, we discuss the effects of these challenges on the learning performance of FL.

1) Data Heterogeneity: In practical wireless networks, devices generate and collect data in a highly non-identically distributed manner [2, 8]. For example, autonomous vehicles run in rural and urban environments may collect different classes of road data information. Thus, different devices have distinct local datasets, the data label distributions across devices are heterogeneous, i.e., non-independent and identically (non-IID). Since the PS directly aggregates models learned from the different devices, the data heterogeneity presented on different devices may lead to weak generalization ability of the trained global model, even resulting in an unstable training process of FL [9, 10].

2) *Scarce Wireless Resources*: In practical wireless networks, the wireless bandwidth resources are usually limited, which only allow a small portion of devices to participate the per-round learning process [11, 12]. Since the local datasets among devices are typically non-IID, the limited participating devices may lead to biased model aggregations and greatly degrade the learning performance.

3) *High Communication Overhead:* In FL, the devices are required to transmit their local models to the server for aggregation. The uploading of model/gradient parameters is costly for devices since modern deep neural network (NN) architectures usually possess massive parameters [13, 14]. For instance, the widely used MobileNet [15], a convolutional NN (CNN) for on-device image processing, has 6.9 million parameters, corresponding to 27.6 MB. Training such a model requires devices to upload 27.6 MB of data per round. Considering hundreds of rounds and multiple devices, the communication overhead is heavy for wireless networks with limited spectrum and energy resources.

4) *Heterogeneous Local Models:* In practical wireless networks, devices are usually equipped with different neural networks (NNs) in terms of architectures and model sizes due to their heterogeneous computing capabilities and storage resources [16, 17]. In this case, the traditional model aggregation-based FL approaches fail to coordinate devices to perform the learning process.

5) *Heterogenous Devices*: In practice, edge devices are drastically different in com-

putation and communication capabilities. Most existing wireless FL studies focused on homogeneous model settings where all devices train identical models in each round. In this setting, the devices with poor capabilities delay global aggregation and slow down the learning convergence, as well as restrict the scale of the global model due to their resource bottlenecks [12, 18, 19]. It is worth mentioning that although the personalized FL approaches [12, 20] were developed to enable devices to train heterogeneous local models, they aim to train a customized local model for each device based on their individual local data distribution that may not generalize well on the classes out of their local data classes. When a device predicts classes are not in its local data, the personalized model shows lower performance than the generalized global model. Thus, it is essential to develop efficient methods for FL to train a generalized shared global model while mitigating the straggler effect in the homogeneous local model setting.

6) *Unreliable Communication*: In addition, the conventional FL algorithms assume an error-free wireless channel and ignore the unreliable nature of wireless communications [21]. Due to devices' constrained transmit power and bandwidth, it is hard to guarantee all the scheduled devices successfully transmit their parameters to the edge server [22]. This brings a new challenge for FL to enhance the robustness of the training process and mitigate the impact of erroneous transmission. An intuitive solution [23] is to discard the devices' parameter with errors, but it further reduces the number of participating devices and exacerbates the performance loss of FL. Thus, it is essential to develop innovative approaches for FL to address the unreliability of wireless transmissions.

1.2 Outline and Contributions

This thesis aims to develop several FL solutions to address the above challenges. Table 1-A summarizes the challenges addressed by each solution. The proposed solutions are all focused on jointly designing the learning mechanism and wireless networks to improve the learning performance of FL. In the following, the outline of this thesis is presented,

Table 1-A: Summary of challenges addressed by different solutions (✓: Addressed, ✗: Not Addressed)

Chapter	Data Heterogeneity	Scarce Wireless Resources	High Communication Overhead	Heterogeneous Local Models	Heterogenous Devices	Unreliable Communication
3	✓	✓	✗	✗	✓	✗
4	✓	✓	✓	✓	✓	✗
5	✓	✓	✗	✗	✓	✓
6	✓	✓	✓	✓	✓	✗

in which the key contributions of each chapter are briefly summarized.

Chapter 2 presents a comprehensive review of related literature, basic assumptions in the FL convergence analysis, and the main optimization techniques used in this thesis. Specifically, the review of literature on addressing the challenges in Section 1.2 is presented, including the resources optimization approaches for improving communication and computation efficiency, device scheduling approaches for coping with limited wireless resources, knowledge distillation schemes on tackling data and model heterogeneity, robust FL design under unreliable communication conditions, and model pruning approaches on tackling device heterogeneity. Then, the basic assumptions and corresponding illustrations for convergence analysis are provided. Finally, mathematical preliminaries of optimization techniques used in the proposed algorithms are presented.

Chapter 3 proposes a joint representativity and latency-aware device scheduling approach to improve the learning performance of FL in resource-limited wireless networks. Firstly, we theoretically characterize the convergence behaviour of the considered FL system, finding that the learning performance is degraded by the difference between the aggregated gradient of scheduled devices and the full participation gradient. Inspired by this, we propose to find a subset of representative devices and the corresponding predevice stepsizes to approximate the full participation aggregated gradient. Considering the limited wireless bandwidth, we formulate a problem to capture the trade-off between representativity and latency by optimizing device scheduling and bandwidth allocation policies. Then, we derive the optimal bandwidth allocation policies for devices using the convex optimization technique. By proving the non-monotone submodularity of the problem, we develop a double greedy algorithm to solve the device scheduling policy. In addition, to avoid the local training of unscheduled devices, we utilize the historical gradient information of devices to estimate the current gradient for device scheduling design. The experimental results verified the effectiveness of the proposed approach on accelerating convergence speed and improving learning accuracy for FL.

Chapter 4 proposes a novel knowledge-aided FL (KFL) framework and an online device scheduling approach to address the data and model heterogeneity issues and reduce communication costs. In KFL, the server aggregates light high-level data features of devices, namely knowledge, in the per-round learning process. This framework allows devices to design their machine-learning models independently and reduces the communication overhead in the training process. We then theoretically analyze the convergence bound of the KFL, revealing that scheduling more data volume in each round helps to improve the learning performance. In addition, large data volume should be scheduled in early rounds if the total scheduled data volume during the entire learning course is fixed. Inspired by this, we define a new objective function, i.e., the weighted scheduled data sample volume, to transform the inexplicit global loss minimization problem into a tractable one for device scheduling, bandwidth allocation, and power control. To deal with unknown time-varying wireless channels, we transform the considered problem into a deterministic problem for each round with the assistance of the Lyapunov optimization framework. Then, we derive the optimal bandwidth allocation and power control solution by convex optimization techniques. We also develop an efficient online device scheduling algorithm to achieve an energy-learning trade-off in the learning process. Experimental results on two typical datasets (i.e., MNIST and CIFAR-10) under highly heterogeneous local data distributions show that the proposed KFL is capable of reducing over 99% communication overhead while achieving better learning performance than the conventional model aggregation-based algorithms.

Chapter 5 proposes a novel propose a novel FL framework, namely FL with gradient recycling (FL-GR), to tackle the unreliable communications and limited wireless resources. In FL-GR, the server recycles the historical gradients of unscheduled and transmission-failure devices to improve the learning performance of FL. In addition, to reduce the hardware requirements for implementing FL-GR in the practical network, we develop a memory-friendly FL-GR that is equivalent to FL-GR but requires low memory of the edge server. We then theoretically analyze how the wireless network parameters affect the convergence bound of FL-GR, revealing that minimizing the average square of local gradients' staleness (AS-GS) helps improve the learning performance. Based on this, we formulate a joint device scheduling, resource allocation and power control optimization problem to minimize the AS-GS for global loss minimization. To solve the problem, we first derive the optimal power control policy for devices and transform the AS-GS minimization problem into a bipartite graph matching problem. Through detailed analysis, we further transform the bipartite matching problem into an equivalent linear program which is convenient to solve. Extensive simulation results verified the efficacy of the proposed methods.

Chapter 6 develops an adaptive model pruning-based FL (AMP-FL) framework to tackle the device heterogeneity. In AMP-FL, the edge server dynamically generates sub-models by pruning the global model for devices' local training to adapt their heterogeneous computation capabilities and time-varying channel conditions. Since the involvement of diverse structures of devices' sub-models in the global model updating may negatively affect the training convergence, we propose compensating for the gradients of pruned model regions by devices' historical gradients. We then introduce an age of information (AoI) metric to characterize the staleness of local gradients and theoretically analyze the convergence behaviour of AMP-FL. The convergence bound suggests scheduling devices with large AoI of gradients and pruning the model regions with small AoI for devices to improve the learning performance. Inspired by this, we develop a joint device scheduling, model pruning, and resource block allocation approach to enhance the learning performance of FL in wireless networks. Experimental results demonstrate the effectiveness and superiority of the proposed approaches.

Chapter 7 concludes this thesis and provide some thoughts for future work.

1.3 Author's Publications

- **Journal Papers:**

  1. **Z. Chen**, W. Yi, H. Shin, and A. Nallanathan, "Adaptive Semi-Asynchronous Federated Learning in Wireless Edge Networks," *IEEE Transactions on Communications*, 2024. (Early Access, `DOI:10.1109/TCOMM.2024.3425635`)

  2. **Z. Chen**, W. Yi, A. Nallanathan, and G. Y. Li, "Efficient Wireless Federated Learning with Partial Model Aggregation," *IEEE Transactions on Communications*, 2024. (Early Access, `DOI:10.1109/TCOMM.2024.3396748`)

  3. **Z. Chen**, W. Yi, Y. Liu, and A. Nallanathan, "Robust Federated Learning for Unreliable and Resource-limited Wireless Networks," *IEEE Transactions on Wireless Communications*, 2024. (Early Access, `DOI:10.1109/TWC.2024.3366393`)

  4. **Z. Chen**, W. Yi, and A. Nallanathan, "Exploring Representativity in Device Scheduling for Wireless Federated Learning," *IEEE Transactions on Wireless Communications*, vol. 23, no. 1, pp. 720-735, 2024.

  5. **Z. Chen**, W. Yi, H. Shin, and A. Nallanathan, "Adaptive Model Pruning for Communication and Computation Efficient Wireless Federated Learning," *IEEE Transactions on Wireless Communications*, 2023. (Early Access, `DOI:10.1109/TWC.2023.3342626`)

  6. **Z. Chen**, W. Yi, Y. Liu, and A. Nallanathan, "Knowledge-aided Federated

Learning for Energy-limited Wireless Networks," *IEEE Transactions on Communications*, vol. 71, no. 6, pp. 3368-3386, 2023.

- **Conference Papers:**

  1. **Z. Chen**, W. Yi, H. Shin, and A. Nallanathan, "Fast Wireless Federated Learning with Adaptive Synchronous Degree Control", in *Proc. IEEE Vehicular Technology Conference (VTC-Spring)*, June, 2024. Singapore.

  2. **Z. Chen**, W. Yi, S. Lambotharan and A. Nallanathan, "Efficient Wireless Federated Learning with Adaptive Model Pruning", in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, December, 2023. Kuala Lumpur, Malaysia.

  3. **Z. Chen**, W. Yi, Y. Liu and A. Nallanathan, Convergence Analysis for Wireless Federated Learning with Gradient Recycling, in *Proc. International Wireless Communications and Mobile Computing (IWCMC)*, June, 2023. Marrakesh, Morocco.

  4. **Z. Chen**, W. Yi, Y. Liu, and A. Nallanathan, "Communication-efficient Federated Learning with Heterogeneous Devices" in *Proc. IEEE International Conference on Communications (ICC)*, May, 2023. Rome, Italy.

  5. **Z. Chen**, W. Yi, A. Nallanathan, and G. Y. Li, "Is Partial Model Aggregation Energy-efficient for Federated Learning Enabled Wireless Networks?" in *Proc. IEEE International Conference on Communications (ICC)*, May, 2023. Rome, Italy.

  6. **Z. Chen**, W. Yi, Y. Deng, and A. Nallanathan, "Device scheduling for wireless federated learning with latency and representativity", in *Proc. International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, November, 2022. Maldives, Maldives.

Chapter 2

Literature Review and Mathematical Preliminaries

2.1 Review of Related Works

In this section, the existing works in different aspects of FL that are related to this thesis are presented, including device scheduling approaches, resource optimization methods, aggregation mechanisms, transmission faults addressing schemes, and model pruning techniques.

2.1.1 Device Scheduling and Resource Allocaton

In wireless FL, the main challenge is that the limited communication resource and stringent training latency only allow a small proportion of devices to upload their local models in each round for aggregation. Due to the few participant devices, the learning performance of wireless FL may be drastically degraded [12]. Thus, it is crucial to judiciously design the device scheduling and wireless resource management policies that maximize the learning performance of FL.

Existing device scheduling works in FL mainly focused on channel-condition-aware scheduling [24, 25], parameter-importance-aware scheduling [26–30], as well as their joint scheduling [31, 32]. Specifically, the joint device scheduling and bandwidth allocation scheme in [24] that maximizes the scheduled data samples is efficient in improving learning performance. A probabilistic device scheduling policy has been developed in [25] to minimize communication time. Although channel-condition-aware scheduling algorithms increase the number of participating devices in the learning process, they may degrade the learning performance due to the significant variance introduced in the device selection procedure. In the parameter-importance-aware scheduling schemes, the selected probability of each device is determined proportionally to its importance measured by the norm of gradients [26, 27], loss function values [28], test accuracy [29], or data diversity [30]. By measuring the significance of devices with their gradient norm, a parameter importance-aware user selection scheme has been developed in [27] to minimize the convergence time of FL. Allocating larger scheduling probabilities to devices with higher gradient norms has been proposed in [26], which is capable of accelerating the convergence for FL. It has been revealed in [28] that scheduling the devices with higher local loss achieves faster convergence. Maximizing the scheduling probability of clients with higher test accuracy in [29] has proven effective in stateful FL. However, it is ineffective in stateless FL and requires pre-known testing accuracies of devices. Although the above parameter-importance-aware scheduling schemes improve the learning performance, they need all devices to perform local training in each round. In [30], prioritizing devices with rich and diverse datasets in the device scheduling policy has achieved higher accuracy and lower learning costs than random device scheduling. To accelerate the learning convergence of FL in practical wireless networks, several device scheduling works considered both channel conditions and parameter importance. The scheduling policy based on both devices' channel conditions and gradient norms in [31] provides a better learning performance than scheduling policies based on a single metric, in which the norm of model update measures the model significance. By measuring the clients' potential contributions with the information entropy of their gradients, the joint channel and contribution-aware scheduling algorithm in [32] significantly improve the model accuracy and convergence speed of FL.

In addition to the above device scheduling policies, the joint optimization of device scheduling and wireless resource allocation [27, 33–38] helps further improve the learning performance of FL. Specifically, the joint time allocation, power control, and computation frequency scaling approach in [33] can substantially reduce the energy consumption of FL while satisfying latency requirements. In [34], a multi-dimensional control policy, including bandwidth allocation and workload partitioning, has been studied to improve the energy efficiency of FL. The joint device scheduling and resource management approaches in [33] and [27] effectively reduced the energy consumption and convergence time of wireless FL, respectively. In [35], the co-design of device selection and wireless networks significantly improved the learning accuracy of wireless FL. A joint communication and computation resource allocation scheme has been proposed in [36] to capture the trade-offs among convergence, wallclock time, and energy consumption. In [37], a time-sharing communication scheme was proposed to maximize the scheduled device number for improving the learning convergence of FL.

2.1.2 FL Framework to Address Model Heterogeneity

To allow devices equipped with heterogeneous models in FL, knowledge distillation (KD)-based FL approaches were developed and attracted much attention. In practical wireless networks, devices usually possess different computation capabilities and communication resources. Thus, requiring all the local models to be of the same architecture in many application scenarios may be ineffective. KD is a teacher-student paradigm which transfers the knowledge distilled from the teacher model to the student model [39, 40]. Integrating KD into FL allows devices to independently design their models according to channel conditions and computation capabilities [41–43]. Specifically, the federated KD approach in [44] effectively enabled federated training between heterogeneous models by aggregating local models' logits on a public dataset. In [45], an auxiliary distillation dataset generated by mixing local training data was adopted to empower the FL process, effectively reducing convergence time. In [46], a lightweight generator was deployed at the server to ensemble user information and broadcast to devices to regulate their local training process. By deploying an unlabelled dataset on both the server and devices, a global model was trained using the averaged outputs of local models on this dataset as the supervision label [47, 48]. The adaptive mutual KD and dynamic gradient compression approach in [49] significantly reduced communication costs and achieved competitive results with centralized model learning. The federated distillation method [50] regularized local models to mitigate overfitting during training by treating the global model as the teacher and the local models as the students. Besides enabling devices to design their machine learning models independently, the KD-based FL substantially reduces the transmitted data volume in the wireless channels because output logits are required to upload in the learning process instead of heavy model/gradient parameters. However, these KD-based FL approaches require an extra public dataset to align the student and teacher models' outputs, increasing the computation costs. Moreover, their performance may significantly degrade with the increase in the distribution divergence between the public and on-device datasets that are usually non-IID.

2.1.3 FL Algorithms for Tackling Unreliable Communication

To cope with unreliable communications between devices and the edge server, existing works focused on improving the successful transmission probabilities of devices by resource management [35, 51–53] and retransmission mechanism design [54–56], as well as compensate for the unsuccessful received devices' models by the past models [57, 58]. The device scheduling and resource allocation algorithm in [35] can maximize the expected number of devices with successful transmissions. A joint wireless resources and quantization bits allocation scheme has been developed in [51] to alleviate the effects of quantization errors and transmission outages on FL convergence performance. The joint device selection and resource allocation approach in [52] can effectively increase the successful information exchange probabilities over wireless networks and thus improve the learning performance of FL. The power allocation and gradient quantization scheme in [53] can improve the convergence speed of FL over a noisy wireless network. However, it only schedules a single device per iteration based on the channel condition. The retransmission protocol in [54] significantly increases the success probability of devices' model uploading, in which devices transmit their local model parameters multiple times, and the edge server uses the received signal with the highest signal-to-interference-plus-noise ratio (SINR) to recover the local models. Different from [54], the retransmission mechanism in [55] utilized the arithmetic mean of the received multiple-times signals from devices to update the global model, effectively reducing global model aggregation errors induced by channel fading in over-the-air FL. While demonstrably effective, the above approaches that maximize devices' successful transmission probabilities only aggregate the successfully uploaded devices' models and thus reduce the number of participants. In addition, the retransmission approaches may cause additional latency and energy consumption for FL. In the presence of decentralized FL systems, by reusing past local models, the robust decentralized SGD approach proposed in [57] under transmission error situations can achieve the same asymptotic convergence rate as the vanilla decentralized SGD with perfect communications. It has been proved in [58] that the FedAvg algorithm replacing error models with past local models in case of devices' model uploading error converges to the same global model parameters as the perfect FedAvg (without communication errors). However, the approaches in [57, 58] assumed that all devices participate in the per-round learning process and did not consider the design of wireless networks.

2.1.4 Model Pruning-Based FL Algorithms

To learn a generalized shared global model while allowing devices to train heterogeneous local models that adapted their communication and computation capabilities, model pruning-based FL approaches were developed to reduce the resource demands for devices and achieve an approximate performance of the original models. Existing model pruning works can be categorised into unstructured weight pruning [59–62] and structured model pruning [63–68]. Specifically, the weight pruning approach prunes the weight parameters in the fully connected (FC) layer of the deep neural network (NN) to achieve both parameters and computation load reduction. The weight importance-aware pruning method in [59] removed the unimportant weights in deep NN, which effectively reduced the model size incurring only a small performance loss. The random pruning mechanism in [60] significantly reduced device communication and computation overhead and avoided model overfitting. In [61], the pruning ratio and bandwidth allocation scheme improved the convergence speed of FL. However, these unstructured weight pruning approaches may be ineffective in reducing the computation load of the convolution NN since the pruned weight connections are from the FC layers. In contrast, the computation overhead is mainly concentrated in convolution layers. For instance, in VGG-16, the FC layers account for 90% of the total parameters but only occupy less than 1% of the overall floating point operations [69]. Moreover, the unstructured pruning approach usually results in irregular weight matrixes in the pruned models that are difficult to compress. Thus, unstructured pruning-based FL requires specialized hardware and software libraries to accelerate the training speed, which may slow its implementation in practical scenarios [70].

To effectively decrease computation and communication overhead, the structured model pruning approach [63–65, 71, 72] was developed to prune both filters in convolution layers and neurons in FC layers to generate sub-models for devices to train. Note that in centralized learning, pruning filters in convolution layers have been demonstrated can effectively accelerate the learning speed without sacrificing too much accuracy [70, 73]. The random sub-model generation scheme in [63] effectively decreased the server-to-client communication and device-side computation costs. The static model pruning approach in [64, 71] or local model composition approach in [74] distributed heterogeneous sub-models to devices for training and then aggregated them into a global inference model, which effectively reduced resource consumption for FL. The model shrinking and gradient compression approach in [72] enabled the local model training with elastic computation and communication overheads. The model pruning method in [65] that dynamically adjusted the model size for resource-limited devices significantly improved the cost-efficiency of FL and obtained an approximate accuracy as the original model. Although these structured model pruning approaches effectively reduced the communication and computation overhead for wireless FL, the different parts in the global model may not be trained evenly across devices. This may induce the different parts in the global model to drift toward different devices and degrade the learning performance of FL.

2.2 Mathematical Preliminaries

For the sake of facilitating convergence analysis throughout this thesis, we introduce two widely used assumptions, i.e., smooth and convexity, in the following. We also introduce some commonly used equivalent conditions of these two assumptions and provide their proof. In addition, some widely used inequalities in FL convergence analysis are presented.

2.2.1 Smooth Functions

Definition 1. *(L-smooth [75]): A function $F(\cdot)$ is L-smooth function if it is continuously differentiable and its gradient $\nabla F(\cdot)$ is Lipschitz continuous with Lipschitz constant L, i.e.,*

$$\|\nabla f(\boldsymbol{w}) - \nabla f(\boldsymbol{v})\| \leq L \|\boldsymbol{w} - \boldsymbol{v}\|, \tag{2.1}$$

holds for any $\boldsymbol{w}, \boldsymbol{v} \in \mathbb{R}^n$.

For the L-smooth functions, we identify five conditions that can be implied by the L-smooth in the following lemma. These conditions are usually used in the convergence analysis of FL.

Lemma 1. *(Implication conditions of smooth): For a L-smooth function $F(\cdot) : \mathbb{R}^n \to \mathbb{R}$, the following conditions are all implied by the smooth with parameter L:*

***Condition** (1): $H(\boldsymbol{w}) = \frac{L}{2}\|\boldsymbol{w}\|^2 - F(\boldsymbol{w})$ is convex, $\forall \boldsymbol{w} \in \mathbb{R}^n$.*

***Condition** (2): $F(\boldsymbol{w}) \leq F(\boldsymbol{v}) + \langle \nabla F(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle + \frac{L}{2}||\boldsymbol{w} - \boldsymbol{v}||^2$, $\forall \boldsymbol{w}, \boldsymbol{v} \in \mathbb{R}^n$*

***Condition** (3): $F(\boldsymbol{u}) \geq F(\boldsymbol{w}) + \langle \nabla F(\boldsymbol{w}), \boldsymbol{u} - \boldsymbol{w} \rangle + \frac{1}{2L}\|\nabla F(\boldsymbol{u}) - \nabla F(\boldsymbol{w})\|^2$.*

***Condition** (4): For all F that are L-smooth with domain $\mathbb{R}^n$, if $\exists \inf_{\boldsymbol{w} \in \mathbb{R}^n} F(\boldsymbol{w}) := F(\boldsymbol{w}^*)$, we have $\frac{1}{2L}\|\nabla F(\boldsymbol{w})\| \leq F(\boldsymbol{w}) - F(\boldsymbol{w}^*)$.*

***Condition** (5): $\langle \nabla F(\boldsymbol{w}) - \nabla F(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle \leq L||\boldsymbol{w} - \boldsymbol{v}||^2$, $\forall \boldsymbol{w}, \boldsymbol{v} \in \mathbb{R}^n$.*

Proof. In the following, we prove the five conditions one by one from the definition of L-smooth. Firstly, for **Condition** (1), the second-order partial derivative of $H(\boldsymbol{w}) = \frac{L}{2}\|\boldsymbol{w}\|^2 - F(\boldsymbol{w})$ is

$$\frac{\partial^2 H(\boldsymbol{w})}{\partial \boldsymbol{w}^2} = L - \nabla^2 F(\boldsymbol{w}). \tag{2.2}$$

Based on the L-smooth of $F(\boldsymbol{w})$, $\frac{\partial^2 H(\boldsymbol{w})}{\partial \boldsymbol{w}^2} \geq 0$. Thus, $H(\boldsymbol{w}) = \frac{L}{2}\|\boldsymbol{w}\|^2 - F(\boldsymbol{w})$ is a convex function with respect to $\boldsymbol{w}$. Thus, **Condition** (1) is proved.

Secondly, we prove **Condition** (2) as follows: From the first-order condition of convex function $H(\boldsymbol{w}) = \frac{L}{2}\|\boldsymbol{w}\|^2 - F(\boldsymbol{w})$, we have

$$H(\boldsymbol{w}) \geq H(\boldsymbol{v}) + \langle \nabla H(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle. \tag{2.3}$$

Substituting $H(\boldsymbol{w}) = \frac{L}{2}\|\boldsymbol{w}\|^2 - F(\boldsymbol{w})$ into the (2.3), we obtain the following inequality:

$$F(\boldsymbol{w}) \leq F(\boldsymbol{v}) + \langle \nabla F(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle + \frac{L}{2}||\boldsymbol{w} - \boldsymbol{v}||^2. \tag{2.4}$$

Thus, **Condition** (2) is proved.

Thirdly, we prove **Condition** (3) as follows: For the sake of proof, we define an auxiliary function $L_{\boldsymbol{w}}(\boldsymbol{v}) = F(\boldsymbol{v}) - \langle \nabla F(\boldsymbol{w}), \boldsymbol{v} \rangle$, which has a minimizer $\boldsymbol{v}^* = \boldsymbol{w}$. In addition, let $\Xi(\boldsymbol{v}) = \frac{L}{2}\|\boldsymbol{v}\|^2 - L_{\boldsymbol{w}}(\boldsymbol{v}) = \frac{L}{2}\|\boldsymbol{v}\|^2 - F(\boldsymbol{v}) + \langle \nabla F(\boldsymbol{w}), \boldsymbol{v} \rangle$. According to

Condition (1), $\frac{L}{2}\left\|\boldsymbol{v}\right\|^2 - F(\boldsymbol{v})$ is a convex function with respect to $\boldsymbol{v}$. Thus, $\Xi(\boldsymbol{v})$ is a convex function with respect to $\boldsymbol{v}$ due to the convexity of $\langle\nabla F(\boldsymbol{w}), \boldsymbol{v}\rangle$. Utilizing the first-order condition of convex function, i.e., $\Xi(\boldsymbol{v}) \geq \Xi(\boldsymbol{u}) + \langle\nabla\Xi(\boldsymbol{u}), \boldsymbol{v}-\boldsymbol{u}\rangle$, we have

$$L_{\boldsymbol{w}}(\boldsymbol{v}) \leq L_{\boldsymbol{w}}(\boldsymbol{u}) + \langle\nabla L_{\boldsymbol{w}}(\boldsymbol{u}), \boldsymbol{v}-\boldsymbol{u}\rangle + \frac{L}{2}\|\boldsymbol{v}-\boldsymbol{u}\|^2. \tag{2.5}$$

Taking minimization with $\boldsymbol{v}$ on the both sides of the above inequation,

$$\begin{aligned}\inf_{\boldsymbol{v}} L_{\boldsymbol{w}}(\boldsymbol{v}) = L_{\boldsymbol{w}}(\boldsymbol{w}) &\leq \inf_{\boldsymbol{v}}\left\{L_{\boldsymbol{w}}(\boldsymbol{u}) + \langle\nabla L_{\boldsymbol{w}}(\boldsymbol{u}), \boldsymbol{v}-\boldsymbol{u}\rangle + \frac{L}{2}\|\boldsymbol{v}-\boldsymbol{u}\|^2\right\}\\ &= \inf_{\|\boldsymbol{x}\|=1}\inf_{\boldsymbol{y}}\left\{L_{\boldsymbol{w}}(\boldsymbol{u}) + \boldsymbol{y}\left\langle\nabla L_{\boldsymbol{w}}(\boldsymbol{u}), \boldsymbol{x}\right\rangle + \frac{L}{2}\boldsymbol{y}^2\right\}\\ &= \inf_{\|\boldsymbol{x}\|=1}\left\{L_{\boldsymbol{w}}(\boldsymbol{u}) - \frac{\|\langle\nabla L_{\boldsymbol{w}}(\boldsymbol{u}), \boldsymbol{x}\rangle\|^2}{2L}\right\}\\ &= L_{\boldsymbol{w}}(\boldsymbol{u}) - \frac{\|\nabla L_{\boldsymbol{w}}(\boldsymbol{u})\|^2}{2L}. \end{aligned} \tag{2.6}$$

Substituting $L_{\boldsymbol{w}}(\boldsymbol{w})$ and $L_{\boldsymbol{w}}(\boldsymbol{u})$ into (2.6), we have

$$F(\boldsymbol{u}) \geq F(\boldsymbol{w}) + \langle\nabla F(\boldsymbol{w}), \boldsymbol{u}-\boldsymbol{w}\rangle + \frac{1}{2L}\|\nabla F(\boldsymbol{u}) - \nabla F(\boldsymbol{w})\|^2. \tag{2.7}$$

Thus, **Condition** (3) is proved.

Fourthly, we prove **Condition** (4) as follows: Let $F(\boldsymbol{w}^*)$ denote the optimal value, i.e., $F(\boldsymbol{w}^*) \leq F(\boldsymbol{w}), \forall\boldsymbol{w} \in \mathbb{R}^n$. Based on the L-smooth of $F(\boldsymbol{w})$, we have

$$\begin{aligned}F(\boldsymbol{w}^*) &\leq F\left(\boldsymbol{w} - \frac{1}{L}\nabla F(\boldsymbol{w})\right)\\ &\leq F(\boldsymbol{w}) - \left\langle\nabla F(\boldsymbol{w}), \frac{1}{L}\nabla F(\boldsymbol{w})\right\rangle + \frac{1}{2L}\|\nabla F(\boldsymbol{w})\|^2\\ &= F(\boldsymbol{w}) - \frac{1}{2L}\|\nabla F(\boldsymbol{w})\|^2. \end{aligned} \tag{2.8}$$

By rearranging the above inequality, $F(\boldsymbol{w})$ with L-smooth satisfies

$$\|\nabla F(\boldsymbol{w})\|^2 \leq 2L\left(F(\boldsymbol{w}) - F(\boldsymbol{w}^*)\right). \tag{2.9}$$

Thus, **Condition** (4) is proved.

Lastly, we prove **Condition** (5) as follows: According to the monotone gradient condition for convexity of $H(\boldsymbol{w}) = \frac{L}{2}\|\boldsymbol{w}\|^2 - F(\boldsymbol{w})$, i.e., $\langle\nabla H(\boldsymbol{w}) - \nabla H(\boldsymbol{v}), \boldsymbol{w}-\boldsymbol{v}\rangle \geq 0$, we have

$$(L\boldsymbol{w} - \nabla F(\boldsymbol{w}) - L\boldsymbol{v} + \nabla F(\boldsymbol{v}))^T(\boldsymbol{w} - \boldsymbol{v}) \geq 0. \tag{2.10}$$

By rearranging the above inequality, we have

$$L\|\boldsymbol{w} - \boldsymbol{v}\|^2 \geq \langle \nabla F(\boldsymbol{w}) - \nabla F(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle . \tag{2.11}$$

Thus, **Condition** (5) is proved. □

2.2.2 Strongly Convex Functions

Definition 2. *(μ-strongly convex [76]): A continuously differentiable function $F : \mathbb{R}^n \to \mathbb{R}$ is called strongly convex if there is $\mu > 0$ such that*

$$F(\boldsymbol{w}) \geq F(\boldsymbol{v}) + \langle \nabla F(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle + \frac{1}{2}\mu \|\boldsymbol{w} - \boldsymbol{v}\|_2^2 \tag{2.12}$$

holds for any $\boldsymbol{w}, \boldsymbol{v} \in \mathbb{R}^n$.

In the following lemma, we identify four conditions that can be implied by the μ-strongly convex condition. These conditions are commonly used in the convergence analysis of FL.

Lemma 2. *(Implication conditions of strongly convexity): For a strongly convex function $F : \mathbb{R}^m \to \mathbb{R}$, the following conditions are all implied by strongly convexity with parameter μ:*

***Condition** (1): $H(\boldsymbol{w}) = F(\boldsymbol{w}) - \frac{\mu}{2}\|\boldsymbol{w}\|^2$ is convex, $\forall \boldsymbol{w} \in \mathbb{R}^n$.*

***Condition** (2): $F(\alpha\boldsymbol{w} + (1-\alpha)\boldsymbol{v}) \leq \alpha F(\boldsymbol{w}) + (1-\alpha)F(\boldsymbol{v}) - \frac{\alpha(1-\alpha)\mu}{2}\|\boldsymbol{w} - \boldsymbol{v}\|^2$.*

***Condition** (3): $\langle \nabla F(\boldsymbol{w}) - \nabla F(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle \geq \mu \|\boldsymbol{w} - \boldsymbol{v}\|_2^2$.*

***Condition** (4): $\|\nabla F(\boldsymbol{w})\|^2 \geq 2\mu[F(\boldsymbol{w}) - F^*]$.*

Proof. In the following, we prove the implication conditions of strongly convexity one by one. Firstly, to prove **Condition** (1), let $\boldsymbol{w}, \boldsymbol{v} \in \mathbb{R}^n$, we have

$$\begin{aligned}
H(\boldsymbol{w}) - H(\boldsymbol{v}) &= F(\boldsymbol{w}) - F(\boldsymbol{v}) - \frac{\mu}{2}(\|\boldsymbol{w}\|^2 - \|\boldsymbol{v}\|^2) \\
&\overset{(a)}{\geq} \langle \nabla F(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle + \frac{\mu}{2}\|\boldsymbol{w} - \boldsymbol{v}\|^2 - \frac{\mu}{2}(\|\boldsymbol{w}\|^2 - \|\boldsymbol{v}\|^2) \\
&= \langle \nabla F(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle - \mu \langle \boldsymbol{w}, \boldsymbol{v} \rangle + \mu\|\boldsymbol{v}\|^2 \\
&= \langle \nabla F(\boldsymbol{v}) - \mu\boldsymbol{v}, \boldsymbol{w} - \boldsymbol{v} \rangle \\
&= \langle \nabla H(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle, \qquad (2.13)
\end{aligned}$$

where (a) is due to the μ-strongly convexity of $F(\cdot)$. Based on the first-order condition of convex function, $H(\boldsymbol{w}) = F(\boldsymbol{w}) - \frac{\mu}{2}\|\boldsymbol{w}\|^2$ is convex. Thus, **Condition** (1) is proved.

Secondly, we prove **Condition** (2) as follows: According to the convexity of $H(\boldsymbol{w}) = F(\boldsymbol{w}) - \frac{\mu}{2}\|\boldsymbol{w}\|^2$, we have $H(\alpha\boldsymbol{w} + (1-\alpha)\boldsymbol{v}) \leq \alpha H(\boldsymbol{w}) + (1-\alpha)H(\boldsymbol{v})$. Thus

$$\begin{aligned}
H(\alpha\boldsymbol{w} + (1-\alpha)\boldsymbol{v}) &= F(\alpha\boldsymbol{w} + (1-\alpha)\boldsymbol{v}) - \frac{\mu}{2}\left\|\alpha\boldsymbol{w} + (1-\alpha)\boldsymbol{v}\right\|^2 \\
&\leq \alpha H(\boldsymbol{w}) + (1-\alpha)H(\boldsymbol{v}) \\
&= \alpha F(\boldsymbol{w}) - \alpha\frac{\mu}{2}\|\boldsymbol{w}\|^2 + (1-\alpha)F(\boldsymbol{v}) - (1-\alpha)\frac{\mu}{2}\|\boldsymbol{v}\|^2. \qquad (2.14)
\end{aligned}$$

By rearranging the above inequality, we have

$$\begin{aligned}
&F(\alpha\boldsymbol{w} + (1-\alpha)\boldsymbol{v}) \\
&\leq \alpha F(\boldsymbol{w}) + (1-\alpha)F(\boldsymbol{v}) - \alpha\frac{\mu}{2}\|\boldsymbol{w}\|^2 - (1-\alpha)\frac{\mu}{2}\|\boldsymbol{v}\|^2 + \frac{\mu}{2}\left\|\alpha\boldsymbol{w} + (1-\alpha)\boldsymbol{v}\right\|^2 \\
&= \alpha F(\boldsymbol{w}) + (1-\alpha)F(\boldsymbol{v}) - \frac{\mu}{2}\alpha(1-\alpha)\|\boldsymbol{w} - \boldsymbol{v}\|^2. \qquad (2.15)
\end{aligned}$$

Thus, **Condition** (2) is proved.

Thirdly, we prove **Condition** (3) as follows: By using the monotone gradient condition for convexity of $H(\boldsymbol{w}) = F(\boldsymbol{w}) - \frac{\mu}{2}\|\boldsymbol{w}\|^2$, we have $\langle \nabla H(\boldsymbol{w}) - \nabla H(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle \geq 0$. Thus,

$$\begin{aligned}
&\langle \nabla H(\boldsymbol{w}) - \nabla H(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle \\
&= \langle \nabla F(\boldsymbol{w}) - \mu\boldsymbol{w} - \nabla F(\boldsymbol{v}) + \mu\boldsymbol{v}, \boldsymbol{w} - \boldsymbol{v} \rangle \\
&= \langle \nabla F(\boldsymbol{w}) - \nabla F(\boldsymbol{v}) + \mu(\boldsymbol{v} - \boldsymbol{w}), \boldsymbol{w} - \boldsymbol{v} \rangle
\end{aligned}$$

$$
\begin{aligned}
&= \langle \nabla F(\boldsymbol{w}) - \nabla F(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle - \mu \|\boldsymbol{w} - \boldsymbol{v}\|^2 \\
&\geq 0. \qquad (2.16)
\end{aligned}
$$

By rearranging the above inequality, we have

$$\langle \nabla F(\boldsymbol{w}) - \nabla F(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle \geq \mu \|\boldsymbol{w} - \boldsymbol{v}\|^2. \qquad (2.17)$$

Thus, **Condition** (3) is proved.

Finally, we prove **Condition** (4) as follows: Taking minimization respect to $\boldsymbol{w}$ on both sides of (2.12), **Condition** (4) can be proved. Specifically, for the left-hand-side of (2.12), we have

$$\min_{w} F(\boldsymbol{w}) = F(\boldsymbol{v}^*). \qquad (2.18)$$

For the right-hand-side of (2.12), we have

$$\frac{\partial^2 \left(F(\boldsymbol{v}) + \langle \nabla F(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle + \frac{1}{2}\mu \left\| \boldsymbol{w} - \boldsymbol{v} \right\|_2^2 \right)}{\partial \boldsymbol{w}^2} = \nabla F(\boldsymbol{v}) + \mu(\boldsymbol{w} - \boldsymbol{v}) = 0. \qquad (2.19)$$

Thus,

$$\boldsymbol{w} = \boldsymbol{v} - \frac{1}{\mu} \nabla F(\boldsymbol{v}). \qquad (2.20)$$

Substituting (2.18) and (2.20) into (2.12), the proof of **Condition** (4) is completed. □

2.2.3 Important Inequalities

In this subsection, some widely used inequalities in the FL convergence analysis are presented.

Inequality 1. *(Arithmetic mean-Geometric mean inequality: AM-GM inequality): For positive real numbers $x_1, x_2, \cdots, x_n$, the following inequality holds:*

$$\frac{x_1 + x_2 + \cdots + x_n}{n} \geq \sqrt[n]{x_1 x_2 \cdots x_n}, \qquad (2.21)$$

with equality if and only if $x_1 = x_2 = \cdots = x_n$.

There are many existing methods that can be used to prove the AM-GM inequality, e.g., forwardbackward induction method [77] and Lagrangian multipliers method [78].

Inequality 2. *(Jensen's inequality): If f is a convex function in $[a,b]$, the following inequality is true for all $x_i \in [a,b]$ $(i = 1, 2, \cdots, n)$:*

$$f\left(\frac{x_1 + x_2 + \cdots + x_n}{n}\right) \leq \frac{f(x_1) + f(x_2) + \cdots + f(x_n)}{n}. \tag{2.22}$$

Similarly, if f is concave in $[a,b]$, the sign of the above inequality turns over. The proof of Jensen's inequality can be found in [79].

Note that, one special case of Jensen's inequality that is widely used in the FL convergence analysis is $||\frac{1}{n}\sum_{i=1}^{n} x_i||^2 \leq \frac{1}{n}\sum_{i=1}^{n}||x_i||^2$.

Inequality 3. *(Cauchy-Schwarz inequality): For any real numbers x_i, y_i $(i = 1, 2, \cdots, n)$, the following inequality holds:*

$$\sum_{i=1}^{n} x_i^2 \cdot \sum_{i=1}^{n} y_i^2 \geq (\sum_{i=1}^{n} x_i y_i)^2, \tag{2.23}$$

with equality if the sequences are proportional, i.e., $\frac{x_1}{y_1} = \frac{x_2}{y_2} = \cdots = \frac{x_n}{y_n}$. For the proof of Cauchy-Schwarz inequality, please refer to [80].

Inequality 4. *(Young's inequality for products): For any $x \geq 0$ and $y \geq 0$ are nonnegative real numbers, if $p > 1$ and $q > 1$ are real numbers such that $\frac{1}{p} + \frac{1}{q} = 1$, the following inequality holds:*

$$xy \leq \frac{x^p}{p} + \frac{y^q}{q}, \tag{2.24}$$

with equality if and only if $x^p = y^q$. It can be found the proof of Young's inequality for products from [81].

Inequality 5. *(Triangle inequality): Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ are two real vectors, then the triangle inequality states that:*

$$\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|. \tag{2.25}$$

The triangle inequality gets its name from a more geometric interpretation. Two additional widely used formats of triangle inequalities are given as follows:

- $||\boldsymbol{y} - \boldsymbol{x}||^2 \geq \frac{1}{2}||\boldsymbol{y} - \boldsymbol{z}||^2 - ||\boldsymbol{x} - \boldsymbol{z}||^2$, *where $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^n$ are n-dimensional vectors.*

- $||\boldsymbol{x} + \boldsymbol{y}||^2 \leq (1 + \frac{1}{v})||\boldsymbol{x}||^2 + (1 + v)||\boldsymbol{y}||^2$, *where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ are n-dimensional vectors* *and $v > 0$ is a real number.*

Inequality 6. *For any real number $x > -1$ and $x \neq 0$, the following inequality holds:*

$$\ln(1+x) > \frac{x}{1+x}. \tag{2.26}$$

Proof. For the sake of proof, let $f(x) = \frac{x}{1+x} - \ln(1+x)$ as an auxiliary function. The first-order derivative of $f(x)$ is given by

$$\frac{df(x)}{dx} = -\frac{x}{(1+x)^2}. \tag{2.27}$$

Thus, when $x > 0$, we have $\frac{df(x)}{dx} < 0$. That is, $f(x)$ is a decreasing function for all $x > 0$. Hence, $f(x) < f(0) = 0, \forall x > 0$. In addition, when $-1 < x < 0$, we have $\frac{df(x)}{dx} > 0$, i.e., $f(x)$ is a increasing function for all $-1 < x < 0$. Hence, $f(x) < f(0) = 0, \forall - 1 < x < 0$. Based on the above analysis, we have:

$$f(x) = \frac{x}{1+x} - \ln(1+x) < f(0) = 0, \forall x > 0. \tag{2.28}$$

By rearranging the above inequality, we prove that inequality 6 holds when $x > 0$. □

Inequality 7. *For any real number $x > 0$, the following inequality holds:*

$$(1+x)^{\frac{1}{x}} \leq e, \tag{2.29}$$

where e is Euler's number.

Proof. For the sake of proof, we introduce $f(x) = (1+x)^{\frac{1}{x}}$ and $g(x) = \ln f(x) = \frac{1}{x}\ln(1+x)$ as the auxiliary functions. We have $f(x) = e^{g(x)}$. The first-order derivative of $g(x)$ is given by

$$\frac{dg(x)}{dx} = -\frac{1}{x^2}\ln(1+x) + \frac{1}{x(1+x)} = \frac{1}{x}\left(-\frac{1}{x}\ln(1+x) + \frac{1}{1+x}\right). \tag{2.30}$$

According to inequality 6, we have $\frac{dg(x)}{dx} < 0$. Thus,

$$\frac{df(x)}{dx} = e^{g(x)}\frac{dg(x)}{dx} < 0. \tag{2.31}$$

That is, $f(x)$ is a decreasing function with respect to x when $x > 0$. Hence,

$$(1+x)^{\frac{1}{x}} < \lim_{x\to 0}(1+x)^{\frac{1}{x}} = e. \tag{2.32}$$

□

Chapter 3

Exploring Representativity in Device Scheduling

3.1 Introduction

In wireless FL, the main challenge is that the limited communication resource and stringent training latency only allow a small proportion of devices to upload their local models in each round for aggregation. Thus, it is essential to carefully design the device scheduling policy for improving the learning performance of FL. However, traditional device scheduling methods do not utilize computing resources efficiently. Although the approaches in [27, 30, 31, 36] consider both device importance and communication conditions for device scheduling policy design, they require all devices to perform local training in each round and upload corresponding indicators, e.g., the gradient norm. This may produce extra training costs. In addition, the existing scheduling policies that measure device importance based on gradient norm [31, 36], inner product [27], or the diversity of local dataset [30] trend to schedule devices with similar gradient information in each round. This may exacerbate the global model bias toward the scheduled devices and further degrade the learning performance in heterogeneous data distribution scenarios.

To tackle the above issues, this chapter theoretically analyzes the convergence bound of FL with partial device participation, revealing that device scheduling policy affects the convergence through the difference between the aggregated gradient of scheduled devices and the full participation gradient. Based on this, we attempts to select a subset of devices to approximate the full devices' aggregated gradient for accelerating the FL. Considering the limited bandwidth resources in wireless networks, we propose a novel latency- and representativity-aware device scheduling algorithm to accelerate the learning process of FL, in which the heterogeneous communication, computation, and representativity among devices are all taken into account. The main contributions of this chapter are summarized as follows:

- To enable effective FL in bandwidth-limited wireless networks, this chapter theoretically characterize the convergence bound of the considered FL system under the general non-convex loss function setting, finding a new metric, i.e., the difference between the aggregated gradient of scheduled devices and the full participation gradient, which negatively affect the convergence. By minimizing this metric, the convergence speed of FL can be improved.

- To minimize the difference between the scheduled devices' gradient and the full participation gradient, this chapter aim to find a subset of devices and the corresponding pre-device step sizes to approximate the full participation aggregated gradient. To this end, this chapter characterize the representativity of a device set as the approximation error of its aggregated gradient for the full participation gradient. The small approximation error contributes to strong representative. In addition, this chapter utilize the past gradient information of devices to determine the scheduling policy in each round, avoiding the unscheduled devices to perform local training.

- To balance the representativity and latency for the device scheduling policy, this chapter formulate a problem to minimize the weighted sum of gradient approximation error and latency through jointly optimizing the device scheduling and bandwidth allocation policy, which is intractable to solve. The analysis reveals that the optimal bandwidth allocation policy is optimal when all scheduled devices have the same latency. Furthermore, by proving the submodularity of the problem, a double-greedy algorithm is developed to obtain a sub-optimal device scheduling policy.

- Experiments show that the proposed scheduling algorithm achieves faster convergence speed and higher model accuracy than the benchmarks. Specifically, compared to other benchmark algorithms, the proposed device representativity-aware schedule algorithm is able to boost 6.7% and 4.02% accuracies on MNIST and CIFAR-10 datasets, respectively. The proposed latency- and representativity-aware scheduling algorithm saves over 16% and 12% training time for MNIST and CIFAR-10 datasets than the scheduling algorithms based on either latency and representativity individually.

The remainder of this chapter is organized as follows: Section 3.2 introduces the FL system and the training latency model. The convergence analysis of the considered FL algorithm and the problem formulation are illustrated In Section 3.2. Section 3.4 develops three device scheduling algorithms for FL. Section 3.5 verifies the effectiveness of the proposed device scheduling algorithms by simulation. The summary is drawn in Section 3.6.

3.2 System Model

We consider a typical wireless FL system, in which one edge server undertakes the role of the parameter server to coordinate K devices for training a machine learning model. The server and all devices communicate via bandwidth-limited wireless channels, as shown in Fig. 3.1. In each global round, the selected devices first perform local training and then upload their training model parameters to the edge server through the allocated wireless channels for global model updating. The devices are indexed by $\mathcal{K} = \{1, 2, \cdots, K\}$. Each

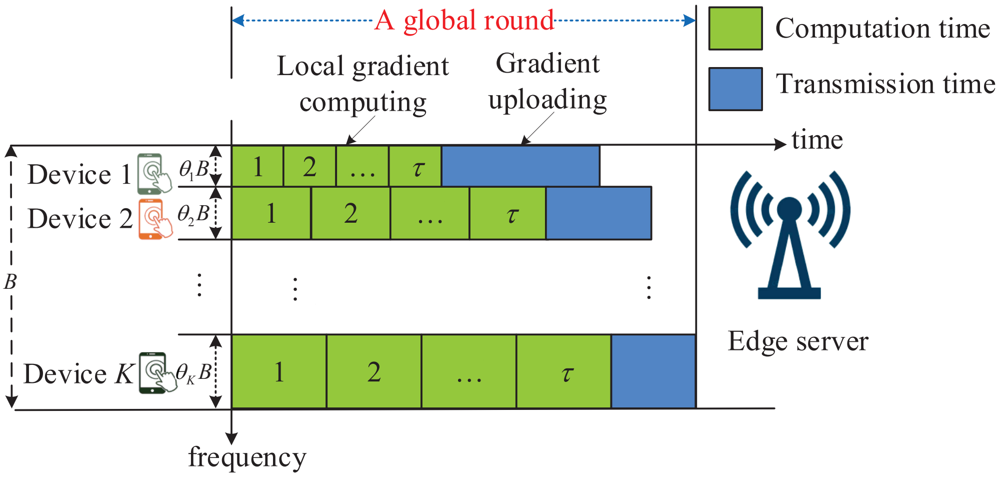


Figure 3.1: An implementation of FL via FDMA, where the scheduled devices perform τ local iterations and upload gradients to the edge server.

device k ($k \in \mathcal{K}$) has a local dataset $\mathcal{D}_k$ with $D_k = |\mathcal{D}_k|$ data samples. Without loss of generality, it is assumed that there is no overlapping for datasets from different devices, i.e., $\mathcal{D}_k \cap \mathcal{D}_h = \emptyset, (\forall k, h \in \mathcal{K})$. Thus, the entire dataset is denoted by $\mathcal{D} = \cup \{\mathcal{D}_k\}_{k=1}^K$ with the total number of samples $D = \sum_{k=1}^K D_k$.

Let $\zeta = (\boldsymbol{x}, y)$ denote a data sample in $\mathcal{D}$, where $\boldsymbol{x} \in \mathbb{R}^d$ is the d-dimensional input feature vector of the sample, and $y \in \mathbb{R}$ is the corresponding ground-truth label. For a machine learning model $\boldsymbol{w}$, let $f(\boldsymbol{w}; \zeta)$ denote its sample-wise loss function on the data sample ζ, which quantifies the error between the ground-truth label y and its predicted output of $\boldsymbol{x}$. Thus, the local loss function of device k that measures the model error on its local dataset $\mathcal{D}_k$ can be defined as

$$F_k(\boldsymbol{w}) \triangleq \frac{1}{D_k} \sum_{\zeta \in \mathcal{D}_k} f(\boldsymbol{w}; \zeta). \tag{3.1}$$

Accordingly, the global loss function associated with all distributed local datasets is given by

$$F(\boldsymbol{w}) \triangleq \sum_{k=1}^{K} p_k F_k(\boldsymbol{w}), \tag{3.2}$$

where p_k is the weight of device k such that $\sum_{k=1}^K p_k = 1$. Similar to many existing works, e.g., [25, 34, 82], this chapter considers a balance device datasets scenario and sets $p_k = \frac{1}{K}$.

3.2.1 Federated Learning Algorithm

The goal of FL is to train a model $\boldsymbol{w}$ by leveraging the devices' local datasets. To preserve the data privacy of devices, the devices collaboratively learn $\boldsymbol{w}$ by only uploading local gradients to the edge server for periodical aggregation, instead of transmitting the raw training data. The edge server orchestrates the training process, by repeating the following steps until the model converge:

1) The edge server selects a subset of devices from $\mathcal{K}$ to participate the training in the current communication round, denoted by $\boldsymbol{S}_t$. Let $\alpha_{k,t} \in \{0, 1\}$ denote the schedule indicator of device k in round t, where $\alpha_{k,t} = 1$ represents device k is scheduled, and $\alpha_{k,t} = 0$ otherwise. Thus, $\boldsymbol{S}_t = \{k : \alpha_{k,t} = 1, \forall k \in \mathcal{K}\}$.

2) The edge server broadcasts the latest global model to the scheduled devices for local training. It is worth mentioning that only the scheduled devices perform local training and upload their local gradients to the edge server for the global model update. Thus, in the FL process, the edge server only broadcasts the latest global model to scheduled devices instead of all devices. After the FL process, the edge server broadcasts the trained global model to all devices for serving them.

3) After receiving the global model, each selected device computes the local gradient $\boldsymbol{g}_{k,t}$ by running τ steps SGD on its local dataset, according to

$$\widetilde{\boldsymbol{g}}_{k,t} = \sum_{l=0}^{\tau-1} \nabla F_k(\boldsymbol{w}_{k,t,l}; \mathcal{B}_{k,t,l}), \tag{3.3}$$

where

$$\nabla F_k(\boldsymbol{w}_{k,t,l}; \mathcal{B}_{k,t,l}) = \frac{1}{L_b} \sum_{\zeta \in \mathcal{B}_{k,t,l}} \nabla f(\boldsymbol{w}_{k,t,l}; \zeta) \tag{3.4}$$

is the gradient in iteration l $(0 \leq l \leq \tau - 1)$, $\mathcal{B}_{k,t,l}$ is a local mini-batch data uniformly sampled from $\mathcal{D}_k$ with $L_b = |\mathcal{B}_{k,t,l}|$ data samples.

4) After all selected devices accomplish local gradients computing, they upload their

gradients to the edge server for aggregation as follows:

$$\widetilde{\boldsymbol{g}}_t = \frac{1}{|\boldsymbol{S}_t|} \sum_{k \in \boldsymbol{S}_t} \widetilde{\boldsymbol{g}}_{k,t}. \tag{3.5}$$

Then, the edge server updates the global model as $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \widetilde{\boldsymbol{g}}_t$, where η denotes the learning rate. For ease of comparison in the following discussion, let $\boldsymbol{g}_t = \frac{1}{K} \sum_{k \in \mathcal{K}} \widetilde{\boldsymbol{g}}_{k,t}$ denote the aggregated gradient for all devices, namely the full-participation stochastic gradient (FP-SG).

3.2.2 Latency Model

In the following, the one-round latency for the FL process is analyzed.

1) Computation latency: Denote C_k as the number of float-point operations (FLOPs) required to process one data sample at device k. Let f_k denote the central processing unit (CPU) frequency of device k. Thus, the local gradient calculation latency of device k can be expressed as

$$T_{k,t}^{\mathrm{L}} = \frac{\tau L_b C_k}{f_k}, \forall t. \tag{3.6}$$

2) Communication latency: Let Q denote the number of elements in each local gradient. Each element is quantized by q bits. In this work, the frequency division multiple access (FDMA) technique is deployed in the system with total B Hz wireless bandwidth for devices to upload their local gradients. Note that, in practical systems using orthogonal frequency-division multiple access (OFDMA), the number of sub-carriers is typically very large (e.g., thousands in 5G systems), and the bandwidth splitting in an OFDMA-based system can be considered continuous [83]. Thus, the bandwidth allocation solution in this chapter based on FDMA can be directly generalised to OFDMA-based systems, where the derived bandwidth allocation ratios by the proposed scheme can be regarded as the ratios of sub-carrier numbers allocated to devices. Let p_k denote the transmit power of device k. We assume that the channel gain, including both small-scale fading and path loss, between device k and the edge server, i.e., $h_{k,t}$, remains unchanged within one round

but varies independently and identically over rounds. Let $\theta_{k,t} \in [0,1]$ denote the fraction of the overall bandwidth allocated to device k in round t, and $\boldsymbol{\theta}_t = (\theta_{1,t}, \theta_{2,t}, \cdots, \theta_{K,t})$. The uplink rate of device k can be characterized by $r_{k,t} = \theta_{k,t} B \log(1 + \frac{p_k h_{k,t}}{\sigma^2})$, where σ^2 is the variance of Gaussian additive noise. Thus, the local gradient uploading latency of device k is

$$T_{k,t}^{\mathrm{C}} = \frac{Qq}{r_{k,t}} = \frac{Qq}{\theta_{k,t} B \log\left(1 + \frac{p_k h_{k,t}}{\sigma^2}\right)}. \tag{3.7}$$

According to the above models, the completion time of each participating device $k (k \in \mathcal{K})$ includes the local computation time $T_{k,t}^{\mathrm{L}}$ and communication time $T_{k,t}^{\mathrm{C}}$, as shown in Fig. 1. The one round latency determined by the slowest device is given by

$$\mathcal{T}_t(\boldsymbol{S}_t) = \max_{k \in \boldsymbol{S}_t} \left\{ T_{k,t}^{\mathrm{L}} + T_{k,t}^{\mathrm{C}} \right\}. \tag{3.8}$$

The above discussion ignored the global model broadcasting and updating latency, because the broadcasting process occupies the entire bandwidth and the edge server has large transmit power, the broadcasting latency is negligible. Moreover, the edge server is usually computational powerful, and the global model update latency can be ignored compared to the communication and computation latencies.

In addition, it is worth mentioning that the above-discussed latency is the training latency for ML models instead of inference latency. In practical applications, the inference (i.e., data processing) latency requirements are usually stringent. For example, to make the virtual reality (VR) world realistic, VR systems require a latency of less than 20 milliseconds, with an ideal target of under 7 milliseconds. Similarly, in the world of autonomous driving, even a 100-millisecond delay can be critical, potentially being the difference between life and death for a pedestrian or car passenger. The traditional centralized learning settings make it difficult to satisfy the low latency requirements as the data processing of one device can affect the whole system in server-dependent systems. To tackle this issue, FL decouples the latency-sensitive applications from server to local devices and is able to provide ultra-low latency for data processing. For instance, a self-driving car (a local device) needs to act as soon as it detects a possible collision. As FL suggests, with a local model on a learner, the learner is decoupled from the server. Consequently, the learner does not require the server's decision, does not need to communicate, and does not wait for a response from the server [84]. Compared to inference latency, training latency is less critical in practical applications. However, it remains important because lower training latency enables more frequent machine learning updates, which improves adaptation to new environments [85].

3.3 Convergence Analysis and Problem Formulation

This section starts with the convergence analysis of the considered FL system under the general non-convex loss function setting, finding a metric, i.e., device representativity, to guide the device scheduling policy design. Then, we formulate an optimization problem for device scheduling which balance the latency and representative ability in each round.

3.3.1 Convergence Analysis

To develop a concrete metric to evaluate the representativity of each local gradient, we first analyze the convergence behavior of the FL system. To this end, we make the following assumptions to the local loss function $F_k(\cdot)$:

Assumption 1. *(Lipschitz gradient continuity): Each local loss functions $F_k(\cdot)(k \in \mathcal{K})$ is continously differentiable, and its gradient $\nabla F_k(\boldsymbol{w})$ is L-Lipschitz continuous, that is*

$$\left\|\nabla F_k(\boldsymbol{w}) - \nabla F_k(\boldsymbol{v})\right\| \leq L \left\|\boldsymbol{w} - \boldsymbol{v}\right\|. \tag{3.9}$$

Assumption 2. *(Unbiased stochastic gradient): For the mini-batch data samples $\mathcal{B}_{k,t}$ that uniformly sampled from $\mathcal{D}_k$ on device k ($k \in \mathcal{K}$), the resulting stochastic gradient is unbiased and variance bounded, that is*

$$\mathbb{E}\left[\nabla F_k(\boldsymbol{w}_{k,t}; \mathcal{B}_{k,t})\right] = \nabla F_k(\boldsymbol{w}_{k,t}), \tag{3.10}$$

and

$$\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{k,t};\mathcal{B}_{k,t})-\nabla F_k(\boldsymbol{w}_{k,t})\right\|^2 \leq G^2. \tag{3.11}$$

Assumption 3. *(Bounded stochastic gradient): The expected squared norm of stochastic gradients is uniformly bounded, i.e.,* $\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{k,t};\mathcal{B}_{k,t})\right\|^2 \leq \chi^2$.

Assumption 1, 2, and 3 are widely used in the convergence analysis of FL systems and satisfied by loss functions for widely used learning models, e.g., support vector machines (SVM), Logistic regression, and most neural networks [86]. In particular, according to [87], a deep neural network defined by a composition of functions is a Lipschitz neural network if the functions in all layers are Lipschitz. It has been proved in [88] and [89] that the convolution layer, linear layer, some nonlinear activation functions (e.g., Sigmoid, tanh, Leaky ReLU, and SoftPlus), and the widely used cross-entropy function have Lipschitz smooth gradients. That is, the loss functions of most neural networks that are consisted of Lipschitz layers and loss functions are Lipschitz continuous. Based on this, we provide the one round convergence bound of the considered FL system in Theorem 1, proved in Appendix A.1.

Theorem 1. *Let Assumption 1, 2, and 3 hold, and the learning rate satisfy* $\eta \leq \frac{1}{L}$, *we have*

$$\begin{aligned}\mathbb{E}\left[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)\right] \leq & -\frac{L}{2}\eta^2\mathbb{E}\|\nabla F(\boldsymbol{w}_t)\|^2+L\eta^2(\tau-1)^2\chi^2 \\ & +L\eta^2(2\tau^2-2\tau+1)G^2+L\eta^2\|-\boldsymbol{g}_t+\widetilde{\boldsymbol{g}}_t\|^2. \end{aligned} \tag{3.12}$$

According to Theorem 1, the expected gap of the loss function values between two global round is bounded by four terms: 1) the squared norm of the ground-truth global gradient $\|\nabla F(\boldsymbol{w}_t)\|^2$; 2) the expected squared norm of stochastic gradients χ^2, 3) the variance of stochastic gradient G^2, 4) the difference between the aggregated gradient of the scheduled devices and the FP-SG that aggregates all devices' stochastic gradients, i.e., $\|-\boldsymbol{g}_t+\widetilde{\boldsymbol{g}}_t\|^2$. The first three terms are independent with the device scheduling decision. The last term is an explicit form related to the device scheduling policy. Thus, the learning performance can be improved by minimizing the $\|-\boldsymbol{g}_t+\widetilde{\boldsymbol{g}}_t\|^2$. Based on

Theorem 1, we further characterize the convergence bound of the considered FL system after T rounds training in Corollary 1, proved in Appendix A.1.1.

Corollary 1. *Let Assumption 1, 2, and 3 hold, and the learning rate satisfy $\eta \leq \frac{1}{L}$, the T-round convergence is upper-bounded by*

$$\begin{aligned}\mathbb{E}[F(\boldsymbol{w}_T) - F(\boldsymbol{w}^*)] \leq & (1 - L^2\eta^2)^{T-1}\mathbb{E}[F(\boldsymbol{w}_0) - F(\boldsymbol{w}^*)] \\ & + \frac{1 - L^2\eta^2 - (1 - L^2\eta^2)^T}{L}\left((2\tau^2 - 2\tau + 1)G^2 + (\tau - 1)^2\chi^2\right) \\ & + \sum_{t=1}^{T-1}(1 - L^2\eta^2)^t L\eta^2 \|{-\boldsymbol{g}_t} + \widetilde{\boldsymbol{g}}_t\|^2. \end{aligned} \tag{3.13}$$

Corollary 1 presents the expected gap between the global loss after T rounds and the optimal loss training, which is bounded by the expected gap between the initial global loss and the optimal one, the variance of SGD, the bounded norm of stochastic gradient, and the cumulative difference of gradient between full participation and partial participation. By minimizing the difference in gradient between full and partial participation in each round, the learning performance can be improved.

3.3.2 Device Representativity Measurement

The previous works, e.g., [31, 90], prone to select the devices with maximum gradient norm to minimize $\|{-\boldsymbol{g}_t} + \widetilde{\boldsymbol{g}}_t\|^2$. To further minimize the $\|{-\boldsymbol{g}_t} + \widetilde{\boldsymbol{g}}_t\|^2$ and accelerate the learning convergence, we aim to find a subset of devices (i.e., $\boldsymbol{S}_t \subseteq \mathcal{K}$) and the corresponding pre-device stepsizes γ_k ($\forall k \in \boldsymbol{S}_t$) in each global round t, such that the aggregated gradient approximate the FP-SG (i.e., $\boldsymbol{g}_t$) that aggregated by all the K devices. Toward this end, we define a mapping function $\varphi : \mathcal{K} \rightarrow \boldsymbol{S}_t$, which maps each device $k \in \mathcal{K}$ to a scheduled device $\varphi(k) \in \boldsymbol{S}_t$ such that the gradient $\nabla F_k(\boldsymbol{w})$ from device k is approximated by the gradient from $\varphi(k)$. For each device $h \in \boldsymbol{S}_t$, let $\mathcal{C}_h = \{k : k \in \mathcal{K}, \varphi(k) = h\}$ denote the set of devices approximated by device h, and $\gamma_h = |\mathcal{C}_h|$. Thus, we have

$$\sum_{k=1}^{K} \widetilde{\boldsymbol{g}}_{k,t} = \sum_{k=1}^{K} \left(\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{\varphi(k),t} + \widetilde{\boldsymbol{g}}_{\varphi(k),t}\right)$$

$$= \sum_{k=1}^{K} \left(\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{\varphi(k),t}\right) + \sum_{k\in\boldsymbol{S}_t} \gamma_k \widetilde{\boldsymbol{g}}_{k,t}. \tag{3.14}$$

By rearranging (3.14) and then taking the norm of both sides, the approximation error of the gradient aggregated from $\boldsymbol{S}_t$ (i.e., $\sum_{k\in\boldsymbol{S}_t} \gamma_k \widetilde{\boldsymbol{g}}_{k,t}$) on the FP-SG satisfies

$$\begin{aligned}\left\|\sum_{k=1}^{K} \widetilde{\boldsymbol{g}}_{k,t} - \sum_{k\in\boldsymbol{S}_t} \gamma_k \widetilde{\boldsymbol{g}}_{k,t}\right\| &= \left\|\sum_{k=1}^{K} \left(\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{\varphi(k),t}\right)\right\| \\ &\leq \sum_{k=1}^{K} \left\|\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{\varphi(k),t}\right\|, \end{aligned} \tag{3.15}$$

where the inequality follows from the triangle inequality. The upper-bound in (3.15) is minimized when φ maps each $k \in \mathcal{K}$ to an device in $\boldsymbol{S}_t$ with minimum Euclidean distance between their gradient. That is, $\varphi(k) = \arg\min_{h\in\boldsymbol{S}_t} \|\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{h,t}\|$. Hence, the approximation error in (3.15) satisfies

$$\left\|\sum_{k=1}^{K} \widetilde{\boldsymbol{g}}_{k,t} - \sum_{k\in\boldsymbol{S}_t} \gamma_k \widetilde{\boldsymbol{g}}_{k,t}\right\| \leq \sum_{k=1}^{K} \min_{h\in\boldsymbol{S}_t} \|\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{h,t}\|. \tag{3.16}$$

Thus, the approximation error can be minimized by minimizing the right-hand side of (3.16). Substituting (3.16) into (3.12), the one-round convergence bound can be expressed as:

$$\begin{aligned}\mathbb{E}\left[F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}_t)\right] \leq& -\frac{L}{2}\eta^2 \mathbb{E}\|\nabla F(\boldsymbol{w}_t)\|^2 + L\eta^2(\tau-1)^2\chi^2 \\ &+ L\eta^2(2\tau^2 - 2\tau + 1)G^2 + L\eta^2 \left(\sum_{k=1}^{K} \min_{h\in\boldsymbol{S}_t} \|\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{h,t}\|\right)^2. \end{aligned} \tag{3.17}$$

The convergence bound in (3.17) shows how the device scheduling policy affects the convergence bound. According to (3.17), the learning performance can be improved by minimizing the upper bound of the approximation error of the gradient aggregated from $\boldsymbol{S}_t$ on the FP-SG, i.e., $\sum_{k=1}^{K} \min_{h\in\boldsymbol{S}_t} \|\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{h,t}\|$. We define $\mathcal{H}(\boldsymbol{S}_t) = \sum_{k=1}^{K} \min_{h\in\boldsymbol{S}_t} \|\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{h,t}\|$ to quantify the approximate error of a device scheduling decision $\boldsymbol{S}_t \subseteq \mathcal{K}$.

3.3.3 Problem Formulation

To accelerate the learning convergence, one should schedule the devices with the lowest latency (implemented by good channel conditions and powerful computing capability) as well as the smallest FP-SG approximation error. However, it rarely happens that a device always has the lowest latency and smallest FP-SG approximation error simultaneously in a practical system. Similar to many existing works, e.g., [36, 91], we aim to capture the trade-offs between device representativity and latency for improving the learning performance of FL. Towards this end, we define two weight factors $\rho_1 \geq 0$ and $\rho_2 \geq 0$ to capture the Pareto-optimal trade-offs among the device representativity and latency, the values of which depend on specific scenarios. A large ρ_1 and small ρ_2 emphasis more on device representativity, while a small ρ_1 and large ρ_2 pay more attention to devices' latency. In addition, similar to many existing works, e.g., [27, 30, 31, 36], we optimize the FL performance in each round since the available bandwidth and devices are independent among rounds, instead of optimizing the FL performance over all rounds under long-term resource constraints as the existing paper [23, 24, 92]. Thus, we formulate the problem as follows:

$$\min_{\boldsymbol{S}_t, \boldsymbol{\theta}_t} \quad \rho_1 \mathcal{H}(\boldsymbol{S}_t) + \rho_2 \mathcal{T}(\boldsymbol{S}_t) \tag{3.18}$$

$$\text{s. t.} \quad \alpha_{k,t} \in \{0,1\}, \tag{3.18a}$$

$$\sum_{k \in \boldsymbol{S}_t} \theta_{k,t} \leq 1, \tag{3.18b}$$

$$0 \leq \theta_{k,t} \leq 1. \tag{3.18c}$$

In problem (3.18), (3.18a) indicates which devices are scheduled in each round. (3.18b) assures that the wireless bandwidth resource allocated to all devices would not exceed the total available bandwidth resource. (3.18c) imposes restrictions on the wireless bandwidth resource allocated to each device. Notably, similar to [36], we can adapt to the problem with hard constraints on latency via setting "virtual devices". According to Lemma 3 in Section 3.4, the optimal bandwidth allocation policy is achieved when all scheduled devices have the same latency. Thus, by setting a virtual device whose latency is the delay constraint into the scheduled device set, the latency of devices can satisfy the delay constraint by adjusting the bandwidth allocation policy. There are two major challenges in solving problem (3.18):

1) **Unknown gradient information of devices**: Problem (3.18) requires devices' gradient information that can only be acquired after local gradient computing and uploading. However, the device scheduling decision should be made before gradient computation.

2) **Non-deterministic polynomial-time hard (NP-Hard)**: Problem (3.18) involves a combinatorial optimization over the multi-dimensional discrete and continuous space, which is challenging to solve. In the following analysis, we show that two special cases of problem (3.18), i.e., latency-aware device scheduling problem and representativity-aware device scheduling problem are both submodular maximization problem, which has been proven to be NP-Hard. Thus, Problem (3.18) is NP-Hard in fact.

3.4 Device Scheduling Policies for Federate Learning

In this section, we develop an efficient algorithm to solve the problem (3.18) within polynomial time complexity. To facilitate the algorithm design, we first focus on analyzing two special cases of problem (3.18): 1) $\rho_1 = 0$ and $\rho_2 = 1$ for the latency aware device scheduling problem, 2) $\rho_1 = 1$ and $\rho_2 = 0$ for the device representativity aware scheduling problem. Then, based on the obtained properties of these two special-case problems, we prove that problem (3.18) is a non-monotone submodular minimization problem. Finally, we develop an efficient double greedy algorithm to solve problem (3.18) and obtain the joint latency and device representativity aware device scheduling policy.

3.4.1 Optimal Wireless Bandwidth Allocation

In this subsection, we solve the optimal bandwidth allocation policy for any given device scheduling policy $\boldsymbol{S}_t$. Given the scheduled device set $\boldsymbol{S}_t$, the optimal bandwidth allocation problem can be decomposed from (3.18) as follows:

$$\begin{aligned} &\min_{\boldsymbol{\theta}_t} \quad \max_{k \in \boldsymbol{S}_t} \left\{ T_{k,t}^{\mathrm{L}} + T_{k,t}^{\mathrm{C}} \right\} \\ &\text{s. t.} \quad (3.18\text{b}), (3.18\text{c}). \end{aligned} \tag{3.19}$$

Problem (3.19) is a typical convex optimization problem [93], we obtain its optimal solution by using Lemma 3, proved in Appendix A.1.2.

Lemma 3. *The optimal wireless bandwidth allocation solution for problem* (3.19) *satisfies the following condition:*

$$\theta_{k,t} = \frac{Qq}{\left(\mathcal{T}_t^*(\boldsymbol{S}_t) - \frac{\tau L_b C_k}{f_k}\right) B \log\left(1 + \frac{p_k h_{k,t}}{\sigma^2}\right)}, \forall k \in \boldsymbol{S}_t, \tag{3.20}$$

where $\mathcal{T}_t^*(\boldsymbol{S}_t)$ is the optimal latency for device scheduling decision $\boldsymbol{S}_t$ in round t, its value is determined by the equation $\sum_{k \in \boldsymbol{S}_t} \theta_{k,t} = 1$.

In Lemma 3, there is still an unknown variable $\mathcal{T}_t^*(\boldsymbol{S}_t)$ in the optimal expression of bandwidth allocation policy. Since $\theta_{k,t}(\mathcal{T}_t(\boldsymbol{S}_t))$ is a monotonically decreasing function with respect to $\mathcal{T}_t(\boldsymbol{S}_t)$, the bisection method can be deployed to obtain the optimal bandwidth allocation policy. To this end, we derive the lower bound and upper bound of $\mathcal{T}(\boldsymbol{S}_t)$ in the following. To derive the lower bound of $\mathcal{T}_t(\boldsymbol{S}_t)$, we have the minimal fraction of bandwidth allocated to devices in $\boldsymbol{S}_t$ should less than $\frac{1}{|\boldsymbol{S}_t|}$, i.e., $\min_{k \in \boldsymbol{S}_t} \theta_{k,t}(\mathcal{T}_t(\boldsymbol{S}_t)) \leq \frac{1}{|\boldsymbol{S}_t|}$. Hence,

$$\frac{\min\limits_{k \in \boldsymbol{S}_t} \frac{Qq}{B \log\left(1 + \frac{p_k h_{k,t}}{\sigma^2}\right)}}{\max\left(\mathcal{T}_t(\boldsymbol{S}_t) - \frac{\tau L_b C_k}{f_k}\right)} \leq \frac{1}{|S_t|}. \tag{3.21}$$

Thus, the lower bound of $\mathcal{T}_t(\boldsymbol{S}_t)$ is

$$\mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t) = \min_{k \in \boldsymbol{S}_t} \frac{|\boldsymbol{S}_t|\, Qq}{B \log\left(1 + \frac{p_k h_{k,t}}{\sigma^2}\right)} + \min_{k \in \boldsymbol{S}_t} \frac{\tau L_b C_k}{f_k}. \tag{3.22}$$

Then, to derive the upper bound of $\mathcal{T}_t(\boldsymbol{S}_t)$, we use $\max_{k \in \boldsymbol{S}_t} \theta_{k,t}(\mathcal{T}_t(\boldsymbol{S}_t)) \geq \frac{1}{|\boldsymbol{S}_t|}$. The

Algorithm 1 Optimal Wireless Bandwidth Allocation

```
Inputs: The scheduled device set $\boldsymbol{S}_t$, the CPU frequency, transmit power, and channel gain of devices in $\boldsymbol{S}_t$.
Initialize the precision requirement $\varepsilon > 0$, compute the lower bound ($\mathcal{T}_{t,\text{lb}}(\boldsymbol{S}_t)$) and upper bound ($\mathcal{T}_{t,\text{ub}}(\boldsymbol{S}_t)$) of the latency based on (3.22) and (3.23), respectively.
repeat
    Set $\mathcal{T} = (\mathcal{T}_{t,\text{lb}}(\boldsymbol{S}_t) + \mathcal{T}_{t,\text{ub}}(\boldsymbol{S}_t))/2$.
    For each device $k \in \boldsymbol{S}_t$, compute the required bandwidth allocation ratio $\theta_{k,t}(\mathcal{T})$ based on (3.20).
    Compute the summation of required bandwidth allocation ratio $\sum_{k \in \boldsymbol{S}_t} \theta_{k,t}(\mathcal{T})$.
    if $\sum_{k \in \boldsymbol{S}_t} \theta_{k,t}(\mathcal{T}) > 1$ then
        Halve the searching region by setting $\mathcal{T}_{t,\text{lb}}(\boldsymbol{S}_t) = \mathcal{T}$ and $\mathcal{T}_{t,\text{ub}}(\boldsymbol{S}_t) = \mathcal{T}_{t,\text{ub}}(\boldsymbol{S}_t)$.
    else if $0 < \sum_{k \in \boldsymbol{S}_t} \theta_{k,t}(\mathcal{T}) < 1 - \varepsilon$ then
        Halve the searching region by setting $\mathcal{T}_{t,\text{lb}}(\boldsymbol{S}_t) = \mathcal{T}_{t,\text{lb}}(\boldsymbol{S}_t)$ and $\mathcal{T}_{t,\text{ub}}(\boldsymbol{S}_t) = \mathcal{T}$.
    else
        Break the circulation.
    end if
until $|\mathcal{T}_{t,\text{ub}}(\boldsymbol{S}_t) - \mathcal{T}_{t,\text{lb}}(\boldsymbol{S}_t)| < \varepsilon$
return The optimal latency $\mathcal{T}_t^*(\boldsymbol{S}_t) = \mathcal{T}$ and the optimal bandwith allocation policy $\boldsymbol{\theta}_t$
```

derivation of the upper bound is similar to that of lower bound, and thus omitted for brevity. The upper bound of $\mathcal{T}_t(\boldsymbol{S}_t)$ is

$$\mathcal{T}_{t,\text{ub}}(\boldsymbol{S}_t) = \max_{k \in \boldsymbol{S}_t} \frac{|\boldsymbol{S}_t|\, Qq}{B \log\left(1 + \frac{p_k h_{k,t}}{\sigma^2}\right)} + \max_{k \in \boldsymbol{S}_t} \frac{\tau L_b C_k}{f_k}. \tag{3.23}$$

According to the lower and upper bounds above, the bisection method is deployed to solve the optimal $\mathcal{T}_t^*(\boldsymbol{S}_t)$. For clarity, we summarize the detailed steps for solving the optimal bandwidth allocation policy in Algorithm 1. The bisection process will halve the searching region in every iteration and terminate when the given precision requirement (i.e., ε) is satisfied. Thus, the time complexity of this bisection method is $\mathcal{O}\left(\log_2 \frac{\mathcal{T}_{t,\text{ub}}(\boldsymbol{S}_t) - \mathcal{T}_{t,\text{lb}}(\boldsymbol{S}_t)}{\varepsilon}\right)$. Based on above analysis, we have the following remark.

Remark 1. *From* (3.20)*, the proportion of the wireless bandwidth allocated to device k ($k \in \mathcal{K}$), i.e., $\theta_{k,t}$, is monotonically decreasing with its CPU frequency f_k and its channel gain $h_{k,t}$. That is, more bandwidth should be allocated to the devices with low computation capability and weak channel conditions.*

3.4.2 Latency-aware Device Scheduling Policy

In this subsection, we investigate a special case of problem (3.18), i.e., the latency-aware device scheduling problem. By setting $\rho_1 = 0$ and $\rho_2 = 1$ in problem (3.18), we formulate the latency-aware device scheduling problem as follows:

$$\min_{\boldsymbol{S}_t, \boldsymbol{\theta}_t} \quad \mathcal{T}(\boldsymbol{S}_t) \tag{3.24}$$

$$\text{s. t.} \quad |\boldsymbol{S}_t| = N, \tag{3.24a}$$

$$(3.18\text{a}), (3.18\text{b}), (3.18\text{c}).$$

Note that, we add a constraint (3.24a) into problem (3.24) since the objective function is monotone with respect to device set size (as shown in the following Lemma 4). Without constraint (3.24a), the solution of problem (3.24) is trivial simply taking the empty device scheduling set (i.e., $\boldsymbol{S}_t = \emptyset$) as the solution. However, by adding constraint (3.24a), the device scheduling problem (3.24) is non-trivial.

Problem (3.24) involving wireless bandwidth allocation and device scheduling is a typical mixed-integer non-linear programming that is generally difficult to solve in polynomial time. Based on the above analysis, the optimal bandwidth allocation policy for any device scheduling set $\boldsymbol{S}_t$ can be obtained by using Algorithm 1, the corresponding optimal latency is denoted as $\mathcal{T}_t^*(\boldsymbol{S}_t)$. Substituting $\mathcal{T}_t^*(\boldsymbol{S}_t)$ into problem (3.24), we transform problem (3.24) into the following equivalent problem:

$$\min_{\boldsymbol{S}_t} \quad \mathcal{T}_t^*(\boldsymbol{S}_t) \tag{3.25}$$

$$\text{s. t.} \quad (3.18\text{c}), (3.24\text{a}).$$

For problem (3.25), an intuitive method to obtain the optimal device scheduling policy is to compute the optimal latency for all the possible device scheduling policies and then select the one with minimal latency. However, there are total C_K^N possible device scheduling policies. In the practical systems, the overall number of devices (i.e., K) is large while the participating device number (i.e., N) in each round is small, inducing a large number of possible scheduling device set. Thus, computing the latency for all possible device scheduling policies is impractical due to the high time complexity. In the following, we prove that problem (3.25) is a submodular set cover problem. Based on this, we find a near-optimal solution for problem (3.25) by using greedy algorithm with polynomial time complexity. To this end, we first introduce the definition of submodular function as follows:

Definition 3. *(Submodular function)[94]: A function $\phi : 2^{\mathcal{K}} \to \mathbb{R}$ is submodular if for every $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2 \subseteq \mathcal{K}$ and $h \in \mathcal{K}\backslash\boldsymbol{S}_2$, it holds $\Delta(h\,|\boldsymbol{S}_1\,) \geq \Delta(h\,|\boldsymbol{S}_2\,)$, where $\Delta(h\,|\boldsymbol{S}_1\,) = \phi(\boldsymbol{S}_1 \cup \{h\}) - \phi(\boldsymbol{S}_1)$ is the discrete derivative of ϕ at $\boldsymbol{S}_1$ with respect to h, also named as marginal gain.*

According to Definition 3, we have the following lemma for the optimal latency function $\mathcal{T}_t^*(\boldsymbol{S}_t)$, proved in Appendix A.1.3.

Lemma 4. *The optimal latency function $\mathcal{T}_t^*(\boldsymbol{S}_t)$ is monotonically increasing with respect to the device set $\boldsymbol{S}_t$, i.e., for device set $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2$, we have $\mathcal{T}_t^*(\boldsymbol{S}_1) < \mathcal{T}_t^*(\boldsymbol{S}_2)$. Moreover, the negative of $\mathcal{T}_t^*(\boldsymbol{S}_t)$, i.e., $-\mathcal{T}_t^*(\boldsymbol{S}_t)$, is a monotonically decreasing submodular function with respect to the device set $\boldsymbol{S}_t$. That is, for device set $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2 \subseteq \mathcal{K}$ and $h \in \mathcal{K}\backslash\boldsymbol{S}_2$, we have*

$$\mathcal{T}_t^*(\{h\} \cup \boldsymbol{S}_1) - \mathcal{T}_t^*(\boldsymbol{S}_1) \leq \mathcal{T}_t^*(\{h\} \cup \boldsymbol{S}_2) - \mathcal{T}_t^*(\boldsymbol{S}_2). \tag{3.26}$$

According to Lemma 4, problem (3.25) is a cardinality constraint submodular maximization problem, which is general NP-Hard. Below we find a near-optimal solution of problem (3.25) by using greedy algorithm [95], which starts from $\boldsymbol{S}_t = \emptyset$, and adds one client $k \in \mathcal{K}\backslash\boldsymbol{S}_t$ with the greatest marginal gain to $\boldsymbol{S}_t$ in every step, i.e. $k = \arg\min_{k\in\mathcal{K}\backslash\boldsymbol{S}_t}(\mathcal{T}_t^*(\boldsymbol{S}_t \cup \{k\}) - \mathcal{T}_t^*(\boldsymbol{S}_t))$. For clarity, we summarize the detail steps for latency-aware device scheduling algorithm in Algorithm 2. Note that Algorithm 2 performs optimal bandwidth allocation (i.e., Algorithm 1) at most KN times for select N devices. Thus, the time complexity of Algorithm 2 is $\mathcal{O}(KN\log_2\frac{\mathcal{T}_{t,\mathrm{ub}}(\boldsymbol{S}_t)-\mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t)}{\varepsilon})$. Based on the performance analysis in [95], the greedy device scheduling algorithm is able to achieve a worst-case approximation factor of $1-\frac{1}{e}$ for the optimal solution, where e is the Euler's number.

Algorithm 2 Greedy Algorithm for Latency-aware Device Scheduling

```
Initialize $\boldsymbol{S}_t \leftarrow \emptyset$ and $\mathcal{T}_t^*(\boldsymbol{S}_t) = 0$, the number of selected devices $N$.
while $|\boldsymbol{S}_t| < N$ do
    for $k \in \mathcal{K} \backslash \boldsymbol{S}_t$ do
        Compute the optimal latency for device set $\boldsymbol{S}_t \cup \{k\}$ as $\mathcal{T}_t^*(\boldsymbol{S}_t \cup \{k\})$ by using Algorithm 1.
    end for
    $k^* = \arg\min_{k \in \mathcal{K} \backslash \boldsymbol{S}_t} (\mathcal{T}_t^*(\boldsymbol{S}_t \cup \{k\}) - \mathcal{T}_t^*(\boldsymbol{S}_t))$.
    $\boldsymbol{S}_t \leftarrow \boldsymbol{S}_t \cup \{k^*\}$.
end while
return The device scheduling set $\boldsymbol{S}_t$.
```

3.4.3 Device Representativity-aware Scheduling Policy

In this subsection, we investigate another special case of problem (3.18), i.e., the device representativity-aware scheduling problem which aims to find a subset of devices and the corresponding pre-device stepsizes to approximate the FP-SG. By setting $\rho_1 = 1$ and $\rho_2 = 0$ in problem (3.18), the device representativity aware scheduling problem can be formulated as follows:

$$\min_{\boldsymbol{S}_t} \quad \mathcal{H}(\boldsymbol{S}_t) \tag{3.27}$$

$$\text{s. t.} \quad |\boldsymbol{S}_t| = N. \tag{3.27a}$$

Similar to the formulation of problem (3.24), we also add a scheduled device number constraint in problem (3.27) since its objective function is monotone with respect to device set size. Without constraint (3.27a), problem (3.27) is trivial simply taking all devices (i.e., $\boldsymbol{S}_t = \mathcal{K}$) to the solution. However, problem (3.27) is still difficult to solve since the edge server requires the gradient information of all devices. The gradient information can only be obtained after local gradient computing and uploading by devices. If the edge server collected all the gradient information for devices, it can directly aggregate all local gradients to minimize the convergence bound in (3.12), and the device scheduling is meaningless. To tackle this challenge, there are two heuristics in the following to estimate the gradient information at the start of each global round.

1) *Estimating by mini-batch gradient (E-MBG)*: Compute the gradient of devices with

a smaller mini-batch data (the batch size is less than L_b), and upload all local gradients to the edge server. This method can only reduce part of the computation cost compared to the method of uploading complete gradient information computed by L_b data samples at each device.

2) *Estimate by past gradient information (E-PG)*: The edge server straightforwardly uses the most recently received gradients from devices to approximate their current gradients for solving the problem (3.27) for device scheduling.

In addition to the above two heuristics, there are some neural-network-based methods, e.g., [27], to predict devices' local gradients, which require collecting devices' gradient information to train extra machine learning models. This may produce extra training time and energy consumption for the FL system. However, the two heuristics are convenient to implement. In particular, the E-PG method simply uses the past gradients of devices to approximate the current one and does not require extra computation and communication costs compared to the E-MBG and neural network-based methods. In addition, the experimental results in Section 3.5 verify that the use of past gradients can effectively approximate devices' current gradients.

To evaluate the effectiveness of these two methods, we show in Fig. 3.2 in Section 3.5 the difference between the recently received gradients at the edge server and the current one of an arbitrary device, under the considered datasets. It is observed that E-MBG ($L_b \in \{4, 8, 16\}$) performs not well due to the high variance of the stochastic gradients, while E-PG has a more accurate estimation of the current one. Note that, similar to many existing works in [27, 30, 31, 36], the E-MBG method requires all devices to compute their gradient with mini-bath data samples and upload their gradient to the edge server. This produces extra computing and transmission costs since the estimated gradients of devices are not used for the global model aggregation. In contrast, the E-PG method only requires the edge server to save the past gradients information for devices and does not require extra computation and transmission. Thus, E-PG is computation and transmission-free compared to E-MBG.

In fact, the past gradient information has been successfully used in FL to estimate the current gradient of devices. For example, replacing the lost gradient (induced by transmission error) in decentralized SGD with the past gradients is able to achieve the same asymptotic convergence rate as the decentralized SGD with no transmission error [57]. Using the most recent ℓ_2-norm of the local gradient to estimate the current one at each device to decide the transmit power has proved to be effective in the over-the-air FL system [92]. Motivated by this, we apply the most recent gradient information of devices uploaded to the edge server to compute the device scheduling policy in the problem (3.27).

For problem (3.27), we have the following lemma, proved in appendix A.1.4.

Lemma 5. *The objective function of problem* (3.27)*, i.e.,* $\mathcal{H}(\boldsymbol{S}_t)$ *is monotonically decreasing with respect to the device set* $\boldsymbol{S}_t$*, i.e., for device set* $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2$*, we have* $\mathcal{H}(\boldsymbol{S}_1) \geq \mathcal{H}(\boldsymbol{S}_2)$*. The negative of* $\mathcal{H}(\boldsymbol{S}_t)$*, i.e.,* $-\mathcal{H}(\boldsymbol{S}_t)$*, is a monotonically increasing submodular function with respect to the device set* $\boldsymbol{S}_t$*, i.e., for device set* $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2 \subseteq \mathcal{K}$*, and* $h \in \mathcal{K} \backslash \boldsymbol{S}_2$*, we have*

$$\mathcal{H}(\boldsymbol{S}_1 \cup \{h\}) - \mathcal{H}(\boldsymbol{S}_1) \leq \mathcal{H}(\boldsymbol{S}_2 \cup \{h\}) - \mathcal{H}(\boldsymbol{S}_2). \tag{3.28}$$

According to Lemma 5, problem (3.27) is also a cardinality constraint submodular maximization problem. Thus, the greedy algorithm [95] is deployed to obtain a suboptimal solution in polynomial time complexity. Similarly, the greedy algorithm starts from $\boldsymbol{S}_t \leftarrow \emptyset$, and adds one device k with the maximum marginal gain, i.e., $k = \arg\min_{h \in \mathcal{K} \backslash \boldsymbol{S}_t} (\mathcal{H}(\boldsymbol{S}_t \cup \{h\}) - \mathcal{H}(\boldsymbol{S}_t))$ in every iteration, until $|\boldsymbol{S}_t| = N$. The detailed steps for finding the representativity-aware device scheduling policy are similar to Algorithm 2, and thus omitted for brevity.

3.4.4 Latency and Representativity-aware Scheduling Policy

In the above subsections, we develop the latency-aware and representativity-aware device scheduling policies. However, devices usually have different computing capabilities and channel conditions in the practical system, as well as different representativity in the different global rounds. Thus, the device scheduling policy should simultaneously consider the devices' latency and gradient representativity for accelerating the learning convergence. In this subsection, by utilizing the properties of latency and device representativity obtained in the above discussions, we develop an efficient algorithm to solve problem (3.18), which balances devices' latency and gradient representativity.

According to Lemma 3, the optimal latency for any device scheduling set $\boldsymbol{S}_t \subseteq \mathcal{K}$ can be obtained by using Algorithm 1, denoted as $\mathcal{T}_t^*(\boldsymbol{S}_t)$. Substituting $\mathcal{T}_t^*(\boldsymbol{S}_t)$ into problem (3.18), we transform (3.18) into the following equivalent problem:

$$\begin{aligned} \min_{\boldsymbol{S}_t} \quad & \mathcal{R}(\boldsymbol{S}_t) = \rho_1 \mathcal{H}(\boldsymbol{S}_t) + \rho_2 \mathcal{T}_t^*(\boldsymbol{S}_t) \\ \text{s. t.} \quad & (3.18\text{a}). \end{aligned} \tag{3.29}$$

For problem (3.29), we have the following remark:

Remark 2. *According to Lemma 4 and Lemma 5, $-\mathcal{T}_t^*(\boldsymbol{S}_t)$ is a monotonically decreasing submodular function with respect to the device set $\boldsymbol{S}_t$, while $-\mathcal{H}(\boldsymbol{S}_t)$ is a monotonically increasing submodular function. Consequently, the negative of the objective function of problem (3.29), i.e., $-\rho_1 \mathcal{T}_t^*(\boldsymbol{S}_t) - \rho_2 \mathcal{H}(\boldsymbol{S}_t)$ is a non-monotone submodular function. Thus, problem (3.29) is an unconstrained non-monotone submodular maximization problem, which is NP-Hard in general.*

Base on Remark 2, we use the double greedy algorithm [96] to find a suboptimal solution for problem (3.29). With regards to the implementation of the proposed algorithm, the edge server requires to collect devices channel information for computing their optimal bandwidth allocation policies and latency. After that, the edge server starts by initializing two device sets, i.e., $\boldsymbol{S}_1 = \emptyset$ and $\boldsymbol{S}_2 = \mathcal{K}$, and then serially passes through the devices in $\mathcal{K}$. When the algorithm passes device k ($k \in \mathcal{K}$), it determines online whether to add k into $\boldsymbol{S}_1$ or remove k from $\boldsymbol{S}_2$. This decision is based on a probability that trades off the gains of adding device k to $\boldsymbol{S}_1$ and removing k from $\boldsymbol{S}_2$. For clarity, we summarize the detailed steps of the double greedy algorithm for solving problem (3.29) in Algorithm

Algorithm 3 Double Greedy Algorithm for Latency and Representativity-aware Device Scheduling

```
Initialize $\boldsymbol{S}_1 \leftarrow \emptyset$ and $\boldsymbol{S}_2 \leftarrow \mathcal{K}$
for $k \in \mathcal{K}$ do
    Let $a_k \leftarrow (\max \mathcal{R}(\boldsymbol{S}_1) - \mathcal{R}(\boldsymbol{S}_1 \cup \{k\}), 0)$
    Let $b_k \leftarrow (\max \mathcal{R}(\boldsymbol{S}_2) - \mathcal{R}(\boldsymbol{S}_2 \backslash \{k\}), 0)$
    If $a_k = b_k = 0$, let $\frac{a_k}{a_k+b_k} = 1$
    With probability $\frac{a_k}{a_k+b_k}$ do $\boldsymbol{S}_1 \leftarrow \boldsymbol{S}_1 \cup \{k\}$ and $\boldsymbol{S}_2 \leftarrow \boldsymbol{S}_2$
    Otherwise $\boldsymbol{S}_1 \leftarrow \boldsymbol{S}_1$ and $\boldsymbol{S}_2 \leftarrow \boldsymbol{S}_2 \backslash \{k\}$
end for
Let $\boldsymbol{S}_t = \boldsymbol{S}_1$ (or $\boldsymbol{S}_t = \boldsymbol{S}_2$).
return The device scheduling set $\boldsymbol{S}_t$.
```

3, which requires solving $2K$ times bandwidth allocation problem for finding the device scheduling set. Thus, the time complexity of Algorithm 3 is $\mathcal{O}(2K \log_2 \frac{\mathcal{T}_{t,\mathrm{ub}}(\boldsymbol{S}_t) - \mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t)}{\varepsilon})$. In addition, for any device ordering, many existing works, e.g., [96, 97], have proved that the double greedy algorithm can achieve a tight 1/2 approximation of the optimal solution. Note that, the achieved approximation ratio of the double greedy algorithm is lower than the approximation ratio of Algorithm 2 (i.e., $1 - \frac{1}{e}$) for the optimal solution of the two special-case problems, i.e., latency-aware device scheduling problem and representativity-aware scheduling problem.

3.5 Numerical Results

In this section, we evaluate the proposed device scheduling algorithms for image classification tasks. All codes are implemented in python 3.8 and Pytorch, running on a Linux server.

3.5.1 Experiment Setting

In this subsection, we present the simulation settings. Unless otherwise specified, the default parameter settings are listed in Table 3-A.

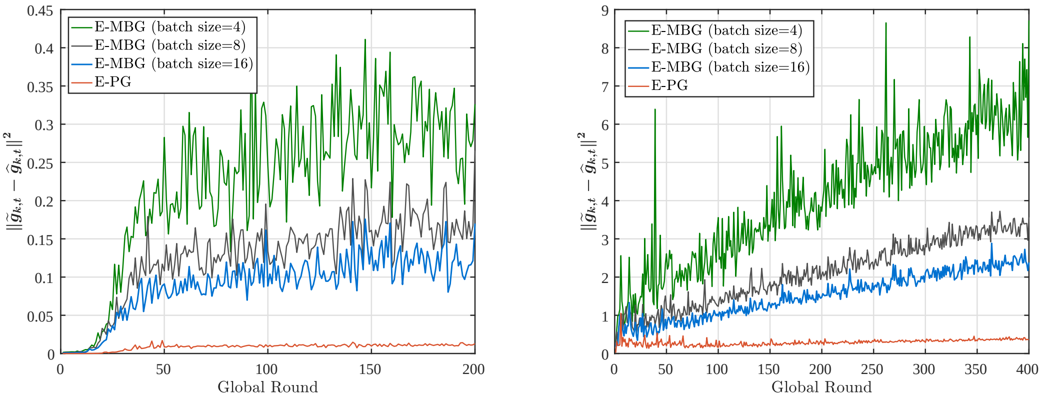
Wireless setting: Unless specified, the default system settings are given as follows:

Table 3-A: System Parameter Settings

Parameter	Value	Parameter	Value
K	100	p_k ($\forall k \in \mathcal{K}$)	10 dBm
B	10 MHz	σ^2	10^{-12} W
L_b	64	h_0	-30 dBm
q	16 bits	η	0.05
τ	8	m	2
Q(MLP)	550,256	C_k(MLP)	550,256
Q(CNN)	307,842	C_k(CNN)	28,206,904

We consider that K devices are randomly distributed within a 500m × 500m cell, and the edge server is located at the centre of this cell. The channel gain of device k ($\forall k \in \mathcal{K}$) is modelled as $h_{k,t} = h_0 \rho_k(t) d_k^{-2}$ [98], where h_0 dBm is the path loss constant; d_k is the distance between device k and the PS; $\rho_k(t) \sim \text{Exp}(1)$ is exponentially distributed with unit mean, which represents the small-scale fading channel power gain from the device k to the PS in round t. The CPU frequency for all devices are random selected from $\{0.8, 1.0, 1.2, 1.4, 1.6\}$ GHz.

Datasets and Models: For the exposition, we evaluate the proposed device scheduling policies under two classification learning tasks, i.e., the handwritten digits classification task on the MNIST dataset and the image classification task on the CIFAR-10 dataset. For the MNIST dataset, we train a multi-layer perceptron (MLP) model with a 784-units input layer, three hidden layers with 512, 256, and 64 units, and a 10-unit softmax output layer. The input layer and three hidden layers are all activated by the ReLU function. The MLP possesses 550346 parameters, which equals the number of FLOPs required for one data sample for gradient calculation. For the CIFAR-10 dataset, we train a CNN with the following structure: two 5×5 convolution layers each with 64 channels and followed by a 2×2 max-pooling layer; three fully connected layers with 1600, 120, and 64 units, respectively; and a 10-unit softmax output layer. Each convolution or fully connected layer is activated by the ReLU function. The CNN possesses 307842 parameters, and the number of FLOPs required for dealing one data sample is 28206904. For both MLP and CNN, the learning rate η is set to 0.05, a momentum of 0.9 is adopted, and cross entropy is adopted as the loss function. Besides, we first classify the training data samples



(a)

(b)

Figure 3.2: The ℓ_2 norm of the difference between the estimated gradient and the true gradient of an arbitrary device k ($k \in \mathcal{K}$): (a) on MNIST dataset, (b) on CIFAR-10 dataset.

according to their labels, then randomly split each class of data samples into $mK/10$ shards, finally randomly distribute m shards of data samples to each device.

3.5.2 Gradient Continuity

In Fig. 3.2, we evaluate the E-MBG and E-PG methods proposed in Section 3.4.3 that estimate the gradient information of arbitrary device k ($k \in \mathcal{K}$). Fig. 3.2 provides the squared norm of the difference between the estimated gradient ($\widehat{\boldsymbol{g}}_{k,t}$) by E-MBG/E-PG and the true gradient of device k on MNIST and CIFAR-10 datasets, respectively. The batch size used to compute the gradient for device k is $L_b = 64$. In each round, device k further computes and records its local gradient with smaller batch sizes $L_b = 4$, 8, and 16, which is used for E-MBG to estimate its local gradient that is computed by $L_b = 64$. The E-PG method adapts the most recently received gradient at the edge server from device k to estimate the current gradient information of device k. For both MNIST and CIFAR-10 datasets, it is observed that E-PG outperforms the E-MBG method. In addition, the gradient estimation errors of E-MBG with different batch sizes are highly varying, and a smaller batch size produces a larger estimation error. Compared to E-MBG, the

E-PG is able to achieve more accurate estimation, as well as no extra computation and communication cost. Thus, the E-PG method is embedded in the device scheduling algorithms in this work.

3.5.3 Performance of Representativity-aware Device Scheduling

To verify the effectiveness of the device representativity-aware scheduling policy proposed in Section 3.4.3, we compare its performance with the following three benchmark device scheduling schemes. Note that, we do not consider the computation latency and communication latency in this subsection.

1) *Random scheduling* (RS): The edge server uniformly selects a subset of devices from all devices to participate in the training in each round.

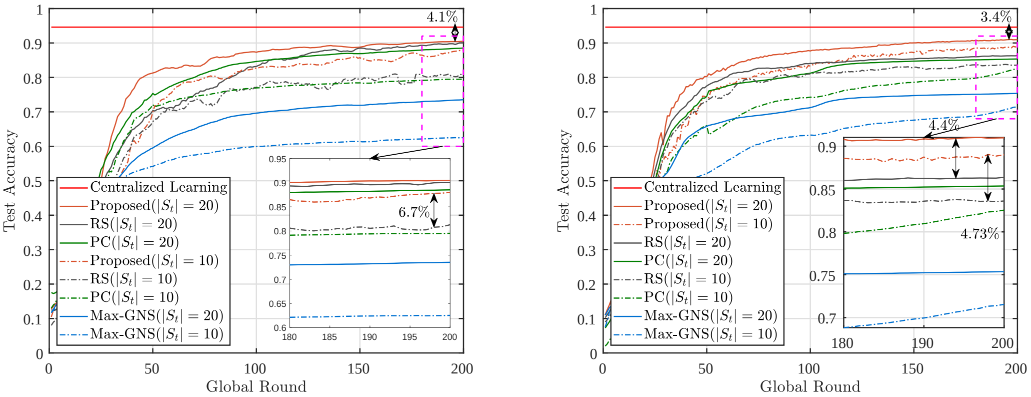
2) *Power-of-Choice scheduling* (PC) [28]: The edge server schedules a subset of devices with larger local losses each round. Note that this scheme requires devices to compute the local loss functions and upload them to the edge server in each round, thus may result in extra computation and transmission costs.

3) *Maximum gradient norm scheduling* (Max-GNS): The edge server schedules a subset of devices with the maximum gradient norm in each round. The ℓ_2-norm of the gradients have been widely used in existing works, e.g., [31, 90], to represent the significance of local gradients. However, the existing works require all devices to perform local training and then upload their gradient norm to the edge server for device scheduling. This may result in the unnecessary energy consumption of the unscheduled devices. Based on the above analysis, we use the past gradient norms of devices in this baseline to decide which devices are scheduled.

Based on the MNIST dataset, Fig. 3.3 compares the learning performance of the proposed algorithm with the above-listed three scheduling schemes under different data heterogeneity and scheduling ratios. In Fig. 3.3(a), we distribute at most two classes of data samples to each device. The results show that our proposed algorithm outperforms the three benchmarks, converges faster, and obtains higher accuracy. Specifically, when 10 devices participate in the learning process in each round, the proposed algorithm achieves a 6.7% accuracy improvement compared with the random scheduling policy. Although the proposed algorithm obtains a similar accuracy to the random scheduling policy when 20 devices participate in each round, it has a faster convergence speed.

Fig. 3.3(b) distributes at most three classes of data samples to each device, in which the data heterogeneity between devices is lower than Fig. 3.3(a). It is observed that the learning accuracies of all the scheduling schemes improved compared with Fig. 3.3(a). This is because high data heterogeneity can weak the generalisation ability of the learned global model, further resulting in poor learning performance. In addition, it is also observed that the proposed algorithm obtains high accuracies than the three benchmarks. Compared with the random scheduling policy, the proposed algorithm obtains a 4.73% accuracy improvement when $|\boldsymbol{S}_t| = 10$ and a 4.4% accuracy improvement when $|\boldsymbol{S}_t| = 20$. In addition, compared to the centralized learning scheme, the proposed approach achieves lower learning accuracy. The performance drop of the proposed approach is induced by the partial device participation and data heterogeneity. Specifically, the performance gap between the proposed approach and centralized learning is reduced from 4.1% ($m = 2$ in Fig. 3.3(a)) to 3.4% ($m = 3$ in Fig. 3.3(b)) along with the decrease of data heterogeneity degree. Thus, in practice, by increasing the device participation number in each round and reducing the data heterogeneity among devices, the learning accuracy of the proposed method can gradually approach centralized learning.

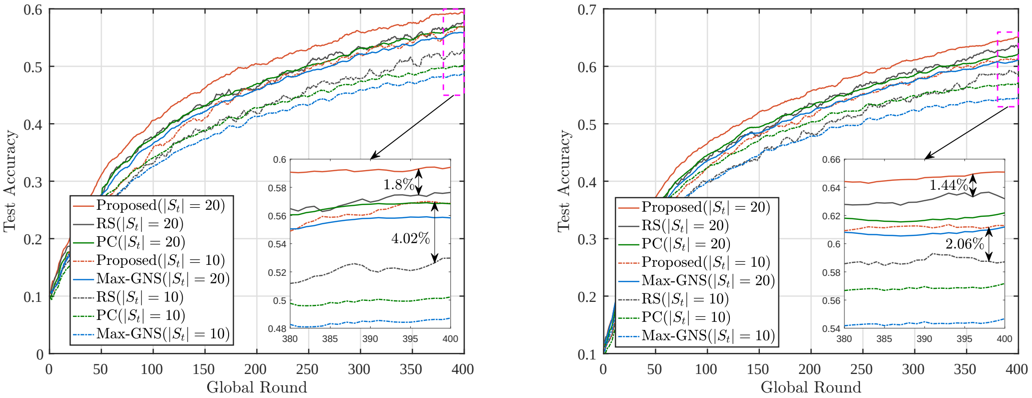
A similar evaluation is conducted on the CIFAR-10 dataset, as shown in Fig. 3.4. In Fig. 3.4(a), we set the data heterogeneity related control parameter as $m = 2$. Compared with the random scheduling policy, the proposed algorithm boosts 4.02% accuracy when $|\boldsymbol{S}_t| = 10$ and improves 1.8% accuracy when $|\boldsymbol{S}_t| = 20$. In Fig. 3.4(b), we set $m = 3$. A distinct accuracy improvement for all scheduling schemes is observed compared with $m = 2$ on this more complicated dataset. In addition, the proposed



(a)

(b)

Figure 3.3: Learning performance of different device scheduling algorithms on MNIST dataset: (a) $m = 2$, (b) $m = 3$.

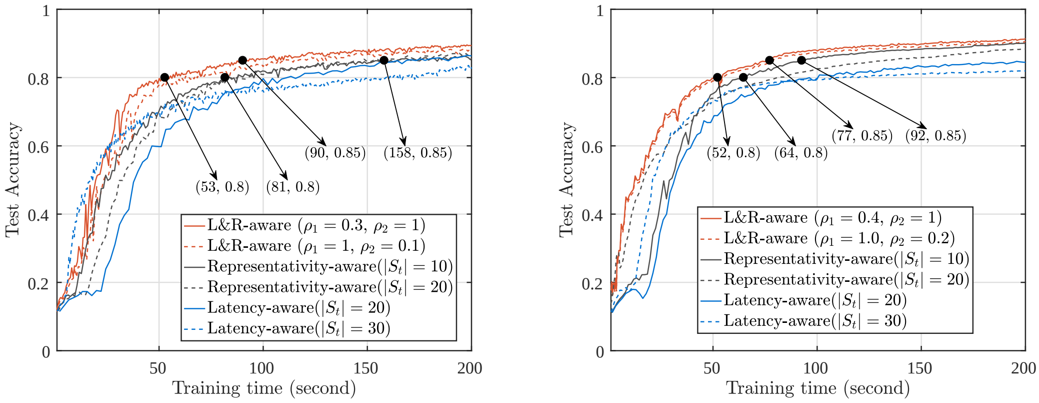


(a)

(b)

Figure 3.4: Learning performance of different device scheduling algorithms on CIFAR-10 dataset: (a) $m = 2$, (b) $m = 3$.

algorithm performs well compared to the three benchmarks, obtaining 2.06% and 1.44% accuracy improvement when $|\boldsymbol{S}_t| = 10$ and $|\boldsymbol{S}_t| = 20$, respectively.



(a) (b)

Figure 3.5: Learning performance of the latency and representativity-aware device scheduling algorithms on MNIST dataset, (a) $m = 2$, (b) $m = 3$.

3.5.4 Performance of Latency and Representativity-aware Device Scheduling

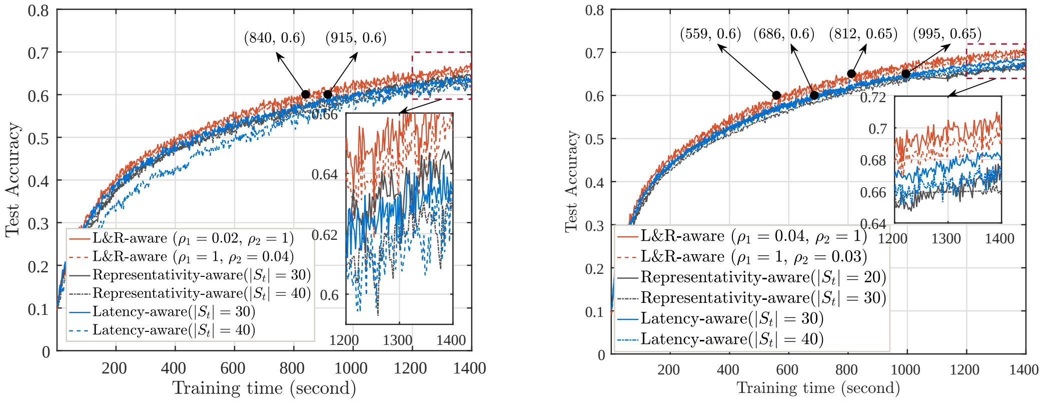
In this subsection, we evaluate the performance of the proposed device scheduling policies, i.e., 1) latency-aware device scheduling (in Section 3.4.2), 2) representativity-aware device scheduling (in Section 3.4.3), 3) latency- and representativity-aware device scheduling (L&R-aware) (in Section 3.4.4). Note that for both latency-aware scheduling and representativity-aware scheduling policies, we test their performance on $|\boldsymbol{S}_t| = 0.1K, 0.2K, \cdots, 1.0K$ and then plot the best two results. For the latency- and representativity-aware device scheduling scheme, the number of participants is automatically decided by the algorithm to adapt the parameters ρ_1 and ρ_2. When ρ_1 is large and ρ_2 is relatively small, $|S_t|$ will increase to reduce the gradient approximation error ($\mathcal{H}(S_t)$) as much as possible. In contrast, when ρ_1 is small while ρ_2 is large, $|S_t|$ would decrease to reduce the latency ($\mathcal{T}(S_t)$). In addition, setting the values of ρ_1 and ρ_2 will not make the L&R-aware-aware solution converges to the representativity-aware solution or latency-aware solution since the constraints are different.

Fig. 3.5 shows the performance of the proposed three device scheduling algorithms on the MNIST dataset. For both $m = 2$ and $m = 3$, we serially select ρ_1 and ρ_2 from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ and plot the best and worst results. It is observed that the proposed latency- and representativity-aware device algorithm always performs better than the other two device scheduling algorithms. In addition, the proposed latency- and representativity-aware device algorithm achieves better performance by setting $\rho_1 < \rho_2$.

Fig. 3.5 shows the performance of the proposed three device scheduling algorithms on the MNIST dataset. For both $m = 2$ and $m = 3$, we set $\rho_1 = 0.3$ and $\rho_2 = 1$. In Fig. 3.5(a), we evaluate the test accuracy with $m = 2$ which indicates that each device possesses at most two classes of the data samples. Specifically, given the target accuracy is 80%, the latency- and representativity-aware device scheduling algorithm only spends 53 seconds for achieving the target, while the representativity-aware scheduling algorithm takes 81 seconds. That is, compared with the representativity-aware scheduling algorithm, the latency- and representativity-aware algorithm is able to save 34.5% training time to obtain 80% test accuracy. In addition, when the target accuracy is 85%, the latency- and representativity-aware algorithm is able to save at least 43% training time in comparison with other device scheduling algorithms.

Fig. 3.5(b) evaluates the performance of the proposed device scheduling algorithms in a less heterogeneous scenario, i.e., $m = 3$. It is observed that all the algorithms perform well in this situation compared to that in $m = 2$. Similar to the evaluation in $m = 2$, the latency- and representativity-aware algorithm obtains the best learning performance. Compared to other device scheduling algorithms, the latency- and representativity-aware algorithm saves 18.8% and 16.3% training time when the target accuracy is 80% and 85%, respectively.

Fig. 3.6 presents the learning performance of the proposed three device scheduling algorithms on the CIFAR-10 dataset. For the latency- and representativity-aware scheduling algorithm, we evaluate its performance by selecting ρ_1 from $\{0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1\}$ and ρ_2 from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,$



(a) (b)

Figure 3.6: Learning performance of the latency and representativity-aware device scheduling algorithms on CIFAR-10 dataset: (a) $m = 2$, (b) $m = 3$.

0.9, 1.0}, then plot the best and worst results. Similar to the results on the MNIST dataset, the latency- and representativity-aware scheduling algorithm provides a better learning performance than the other two scheduling policies based only on either of the two metrics individually. Fig. 3.6(a) presents the learning performance of the device scheduling algorithms with $m = 2$. Specifically, when the target accuracy is 60%, the latency- and representativity-aware scheduling algorithm require at most 88% training time of the other two scheduling schemes. In Fig. 3.6(b), the data heterogeneity parameter is set to $m = 3$. It is observed that all the scheduling algorithms converge faster in this situation than that in $m = 2$. When the target accuracy is 60%, the latency- and representativity-aware scheduling algorithm is able to reduce 18.5% of the training time compared to the other two benchmarks and save 18.4% time when the target accuracy is 65%.

3.6 Summary

In this chapter, we proposed a novel latency- and representativity-aware device scheduling algorithm to accelerate the learning process for FL. We first revealed that the device scheduling policies affect learning convergence through the error between the scheduled devices' aggregated gradient and full participation aggregated gradient. Then, by proving the submodularity of both latency and representativity of the scheduled device set, we developed a double greedy algorithm to capture the trade-off between latency and representativity in each round. To mitigate the extra costs produced by local training of unscheduled devices, we utilized the past gradient information to guide the device scheduling policy design in each round. The experiments verified the effectiveness of the proposed device scheduling algorithm and the use of past gradient information to schedule devices.

Chapter 4

Knowledge-aided Federated Learning over Wireless Networks

4.1 Introduction

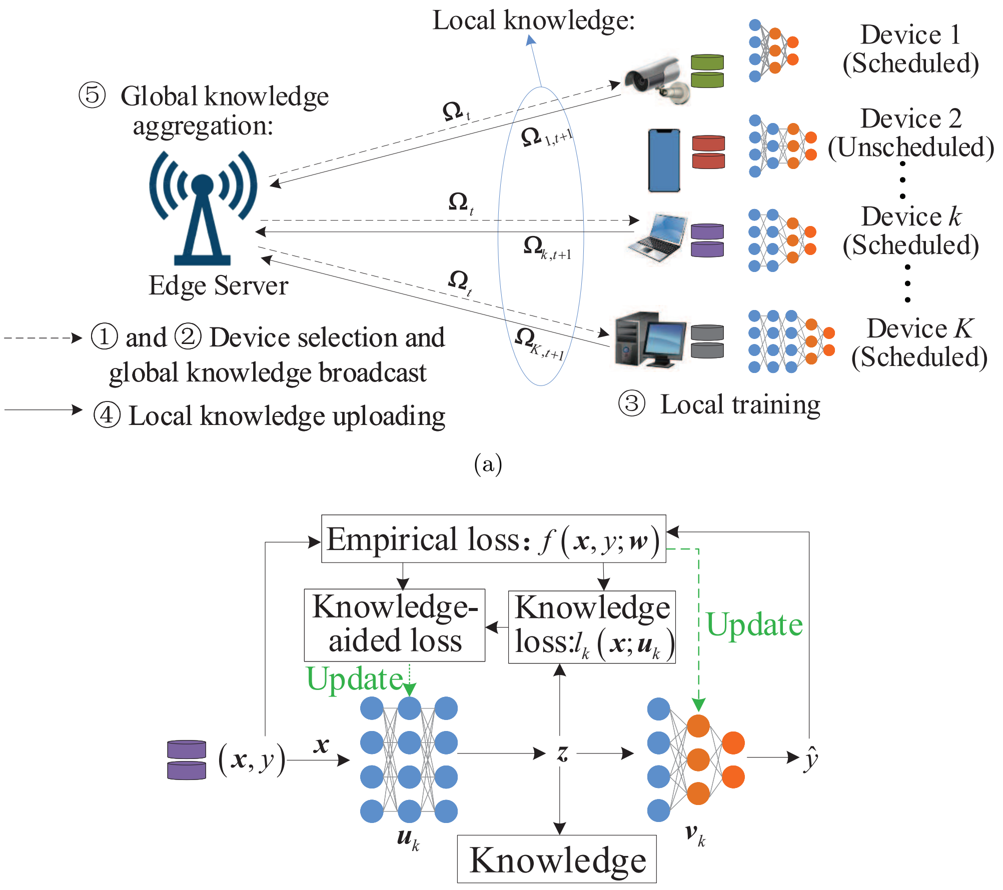
The conventional model aggregation-based FL approach requires all local models to have the same architecture, which fails to support practical scenarios with heterogeneous local models. Moreover, the frequent model exchange is costly for resource-limited wireless networks since modern deep neural networks usually have over a million parameters. Existing works to address the model heterogeneity challenge mainly rely on knowledge distillation, as illustrated in 2.1.2. Nevertheless, the knowledge distillation-based Federated Learning (FL) approaches usually necessitate that the edge server and all devices possess an extra public dataset, which may not be practical for many scenarios. To tackle these challenges, this chapter proposes a novel knowledge-aided FL (KFL) framework, which aggregates light high-level data features, namely knowledge, in the per-round learning process. This framework allows devices to design their learning models independently and reduces the communication overhead in the training process. The main contributions of this paper are summarized as follows:

- We propose a novel KFL framework in which devices collaboratively train models by uploading their knowledge of different data classes to the edge server for aggregation. This design reduces the transmitted data volume in the wireless channels, allowing devices to design their machine-learning models independently according to their computation capabilities and communication conditions.

- We theoretically analyze the convergence bound of the proposed KFL framework under the general non-convex loss function setting, which indicates that scheduling more data samples in each round is able to improve the learning performance. In addition, when the total number of scheduled data volume during the entire learning course is fixed, more data volume should be scheduled in the early rounds.

- We formulate a long-term device scheduling, bandwidth allocation, and power control problem under limited devices' energy budgets with the aid of the convergence bound. To deal with unpredicted time-varying wireless channels and enable online device scheduling, we first transform the original problem into a deterministic problem in each round with the assistance of the Lyapunov optimization framework. Then, we derive the optimal bandwidth allocation and power control through convex optimization techniques. Finally, we develop an efficient polynomial-time algorithm to solve the device scheduling policy with $\mathcal{O}(\sqrt{V}, 1/V)$ energy-learning trade-off guarantee, where V is an algorithm-specific parameter.

- We experimentally verify the correctness of our theoretical results, i.e., more data samples should be scheduled in the early rounds when the total scheduled data volume in the entire learning course are fixed. Compared with benchmark FL algorithms, the proposed KFL framework saves 99% communication overhead and boosts 2.1% and 6.65% accuracy on MNIST and CIFAR-10 datasets, respectively. In addition, The proposed online device scheduling algorithm achieves a faster convergence speed than benchmark scheduling approaches.



(a)

(b)

Figure 4.1: The illustrated KFL over wireless networks: (a) Federated learning with knowledge aggregation mechanism, where devices have different local models, (b) Local training process with the proposed knowledge-aided loss.

The remainder of this chapter is organized as follows: Section 4.2 introduces the proposed KFL system, learning cost and the global loss minimization problem. The convergence analysis and problem transformation are illustrated in Section 4.3. The joint device scheduling, bandwidth allocation, and power control algorithm are developed in 4.4. Section 4.5 verifies the effectiveness of the proposed scheme by simulation. The conclusion is drawn in Section 4.6.

4.2 System Model and Learning Mechanism

In the considered KFL system, as shown in Fig. 4.1(a), an edge server coordinates K different devices to train machine learning models for classification or recognition tasks. Unlike the conventional FL that requires all devices' models to be of the same architecture, the KFL in this work allows devices to be equipped with heterogeneous models. The devices are indexed by $\mathcal{K} = \{1, 2, \cdots, K\}$. For the dataset at devices, the number of data classes in the classification or recognition task is C, indexed by $\mathcal{C} = \{1, 2, \cdots, C\}$. Each device k ($k \in \mathcal{K}$) has a local dataset $\mathcal{D}_k$ with $D_k = |\mathcal{D}_k|$ data samples, in which the data samples belong to c-th class ($c \in \mathcal{C}$) is denoted as $\mathcal{D}_{k,c}$ with $D_{k,c} = |\mathcal{D}_{k,c}|$ data samples. Thus, $\mathcal{D}_k = \cup \{\mathcal{D}_{k,c}\}_{c=1}^{C}$. The entire dataset, $\mathcal{D} = \cup \{\mathcal{D}_k\}_{k=1}^{K}$, is with total number of samples $D = \sum_{k=1}^{K} D_k$. For ease of presentation, we use $\mathcal{D}_c$ to represent all data samples belonging to class c in $\mathcal{D}$. That is, $\mathcal{D}_c = \cup \{\mathcal{D}_{k,c}\}_{k=1}^{K}$ with $D_c = \sum_{k=1}^{K} D_{k,c}$ data samples.

4.2.1 Knowledge-aided Loss Function for Local Training

Let $\zeta = (\boldsymbol{x}, y)$ denote a data sample in $\mathcal{D}$, where $\boldsymbol{x} \in \mathbb{R}^d$ is the d-dimensional input feature vector, $y \in \mathbb{R}$ is the corresponding ground-truth label. Let $\boldsymbol{z} \in \mathbb{R}^p$ be the latent feature vector. As shown in Fig. 4.1(b), the machine learning model parameterized by $\boldsymbol{w} = [\boldsymbol{u}, \boldsymbol{v}]$ consists of two components: a feature extractor $h : \boldsymbol{x} \to \boldsymbol{z}$ parameterized by $\boldsymbol{u}$, and a label predictor $g : \boldsymbol{z} \to \hat{y}$ parameterized by $\boldsymbol{v}$. Before discussing the knowledge-aided loss function, we introduce two fundamental loss functions, i.e., empirical loss and knowledge loss. The empirical loss supervises the local models' training to minimize the prediction error, while the knowledge loss achieves knowledge sharing among devices.

1) **Empirical loss function for local model update**: Let $f(\boldsymbol{x}, y; \boldsymbol{w})$ denote the sample-wise empirical loss function, which quantifies the error between the ground-truth label, y, and the predicted output, $\hat{y}$, based on model $\boldsymbol{w}$. Thus, the local empirical loss function at device k, which measures the model error on its local dataset $\mathcal{D}_k$, is defined

as

$$F_k(\boldsymbol{w}_k) = F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) \triangleq \frac{1}{D_k} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_k} f(\boldsymbol{x}, y; \boldsymbol{w}_k), \tag{4.1}$$

where $\boldsymbol{w}_k$ denotes the machine learning model of device k; $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ correspond to its feature extractor and label predictor parts, respectively. For ease of presentation, we use $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_K)$ to denote all the devices' models throughout this paper. The global loss function associated with all distributed local datasets is given by

$$F(\boldsymbol{W}) = F(\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_K) \triangleq \frac{1}{D} \sum_{k=1}^{K} D_k F_k(\boldsymbol{w}_k). \tag{4.2}$$

The federated learning process is done by solving the following problem:

$$\min_{\boldsymbol{W}=(\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_K)} F(\boldsymbol{W}). \tag{4.3}$$

To preserve the data privacy of devices, the devices collaboratively learn $\boldsymbol{W}$ without transmitting the raw training data. Note that the conventional FL algorithms, e.g., FedAvg [7], aim to find an optimal shared global model $\boldsymbol{w}^* = \boldsymbol{w}_1^* = \cdots = \boldsymbol{w}_K^*$ to minimize the global loss $F(\boldsymbol{W})$. However, this work aims to develop a personalized FL algorithm which trains personalized models for each device to solve the problem (4.3), where different local models are used to fit user-specific data and capture the common knowledge distilled from data of other devices.

2) **Knowledge loss function for local feature extractor update**: When devices are equipped with heterogeneous models, the conventional FL algorithms fail to coordinate devices to train models collaboratively. To tackle this issue, we introduce the knowledge loss function to regularize devices' feature extractors in the training process, achieving knowledge sharing between devices. It is worth mentioning that the knowledge of different devices and classes has the same dimensionality that equals the dimension of feature extractors' output, i.e., p. Let $\boldsymbol{\Omega}_{k,c}$ denote device k's knowledge about data class c, which is defined as the average output of its feature extractor based on the data samples in $\mathcal{D}_{k,c}$, that is

$$\boldsymbol{\Omega}_{k,c} = \frac{1}{D_{k,c}} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} h_k(\boldsymbol{x}; \boldsymbol{u}_{k,t}), \tag{4.4}$$

where $h_k(\cdot)$ denote the feature extractor of device k. Let $\boldsymbol{\Omega}_c$ denote the global knowledge about class c that aggregates all devices' knowledge of class c, i.e.,

$$\boldsymbol{\Omega}_c = \frac{1}{D_c}\sum_{k=1}^{K} D_{k,c}\boldsymbol{\Omega}_{k,c}. \tag{4.5}$$

We use $\boldsymbol{\Omega} = (\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \cdots, \boldsymbol{\Omega}_C)$ to denote the aggregated global knowledge. For each data sample $(\boldsymbol{x}, y) \in \mathcal{D}_{k,c}$ $(\forall k \in \mathcal{K}, c \in \mathcal{C})$, we define the knowledge loss of device k's feature extractor as $l_k(\boldsymbol{x};\boldsymbol{u}_k) = \frac{1}{2}\|h_k(\boldsymbol{x};\boldsymbol{u}_k) - \boldsymbol{\Omega}_c\|^2$, which quantifies the difference between the extracted feature of device k on data sample $(\boldsymbol{x}, y)$ and the global feature of class c. Thus, the knowledge loss of device k is

$$L_k(\boldsymbol{u}_k) = \frac{1}{D_k}\sum_{c=1}^{C}\sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \frac{1}{2}\|h_k(\boldsymbol{x};\boldsymbol{u}_{k,t}) - \boldsymbol{\Omega}_c\|^2, \tag{4.6}$$

which measures the difference between local knowledge and global knowledge. According to (4.6), devices only learn the knowledge of their local data types instead of all the data types. However, it fits devices' local models to their specific data and improves the learning performance on heterogeneous local data scenarios. In addition, devices can use global knowledge to regularize the local training process when new data classes are generated and rapidly adapt their local models to these new class data.

In this work, we define a **knowledge-aided loss function** based on the empirical and knowledge loss functions, i.e., $F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) + \lambda L_k(\boldsymbol{u}_k)$, to guide the feature extractor training for device k $(\forall k \in \mathcal{K})$, where λ is a hyperparameter to balance the empirical loss and knowledge loss for device k. For the label predictor, we still use the conventional empirical loss function.

4.2.2 Knowledge-aided Federated Learning Mechanism

The conventional FL approaches rely on aggregating devices' model/gradient parameters in each round, which induces remarkable communication overhead for wireless networks and requires all the local models to be of the same architecture. To tackle these issues, we propose a novel KFL algorithm to enable collaborative training between heterogeneous local models. Specifically, devices upload their lightweight *knowledge* to the server for aggregation in the per-round training process instead of the heavy model/gradient parameters. The learning process repeats the following steps until the devices' models converge, as shown in Fig. 4.1(a).

1) **Device selection**: The edge server selects a subset of devices from $\mathcal{K}$ to participate in the training process in the current round. Let $\alpha_{k,t} \in \{0,1\}$ denote the scheduling indicator of device k in round t, where $\alpha_{k,t} = 1$ indicates that device k is scheduled in round t, $\alpha_{k,t} = 0$ otherwise. Thus, the scheduled device set in round t is $\boldsymbol{S}_t = \{k : \alpha_{k,t} = 1, \forall k \in \mathcal{K}\}$.

2) **Knowledge broadcast**: In each round t, the edge server broadcasts the latest global knowledge, i.e., $\boldsymbol{\Omega}_t = (\boldsymbol{\Omega}_{1,t}, \boldsymbol{\Omega}_{2,t}, \cdots, \boldsymbol{\Omega}_{C,t})$, to all scheduled devices to regularize their local training process, where $\boldsymbol{\Omega}_{c,t}$ is the c-th class knowledge in round t that is computed in (4.5).

3) **Local training**: All scheduled devices update their local models after receiving the global knowledge, $\boldsymbol{\Omega}_t$, by performing τ steps gradient descent on its local dataset, as shown in Fig. 4.1(b). For device k, its local feature extractor in t-th round is updated as

$$\boldsymbol{u}_{k,t,l+1} = \boldsymbol{u}_{k,t,l} - \eta_u \Big(\nabla_u F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,l}) \Big), \forall l \in \{0, 1, \cdots, \tau - 1\}, \tag{4.7}$$

and its predictor is updated by

$$\boldsymbol{v}_{k,t,l+1} = \boldsymbol{v}_{k,t,l} - \eta_v \nabla_v F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}), \forall l \in \{0, 1, \cdots, \tau - 1\}, \tag{4.8}$$

where η_u and η_v are the learning rate of feature extractor and predictor, respectively, λ is a hyperparameter to balance the empirical loss and knowledge loss for devices k.

4) **Knowledge computing**: After finishing the local iterations, all scheduled devices compute their knowledge for each class c ($c \in \mathcal{C}$) as $\boldsymbol{\Omega}_{k,c,t+1} = \frac{1}{D_{k,c}} \sum_{(\boldsymbol{x},y) \in \mathcal{D}_{k,c}} h_k(\boldsymbol{u}_{k,t+1}; \boldsymbol{x})$. The knowledge of device k for all classes is denoted by $\boldsymbol{\Omega}_{k,t+1} = (\boldsymbol{\Omega}_{k,1,t+1}, \cdots, \boldsymbol{\Omega}_{k,C,t+1})$.

5) **Knowledge aggregation**: After finishing the local knowledge computing, all

Algorithm 4 Knowledge-aided Federated Learning Algorithm

```
Initialization: $t = 0$, training round $T$, and each device initials its local model $\boldsymbol{w}_{k,t}$;
Server side:
for $t = 0, 1, \cdots, T-1$ do
    Select a subset of devices ($\boldsymbol{S}_t$) and broadcasts the latest global knowledge, i.e., $\boldsymbol{\Omega}_t$, to them.
    if Receive the knowledge from the selected devices then
        Aggregate the global knowledge according to (4.9).
    end if
end for
Device side:
if Device $k$ is scheduled then
    Receive the global knowledge, $\boldsymbol{\Omega}_t$, from the edge server;
    for $l = 0, 1, \cdots, \tau - 1$ do
        Update the local feature extractor, $\boldsymbol{u}_{k,t,l+1}$, based on (4.7);
        Update the local predictor, $\boldsymbol{v}_{k,t,l+1}$, based on (4.8);
    end for
    Compute their knowledge for each class $c$ ($c \in \mathcal{C}$) as $\boldsymbol{\Omega}_{k,c,t+1} = \frac{1}{D_{k,c}} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} h_k(\boldsymbol{u}_{k,t+1}; \boldsymbol{x})$.
    Upload the local knowledge $\boldsymbol{\Omega}_{k,t+1} = (\boldsymbol{\Omega}_{k,1,t+1}, \boldsymbol{\Omega}_{k,2,t+1}, \cdots, \boldsymbol{\Omega}_{k,C,t+1})$ to the edge server.
end if
```

scheduled devices upload their knowledge to the edge server through wireless channels for aggregation. Specifically, the edge server computes the global shared knowledge of c-th class as

$$\boldsymbol{\Omega}_{c,t+1} = \frac{\sum\limits_{k\in\boldsymbol{S}_t} D_{k,c}\boldsymbol{\Omega}_{k,c,t+1}}{\sum\limits_{k\in\boldsymbol{S}_t} D_{k,c}}. \tag{4.9}$$

The aggregated global knowledge in round $(t+1)$ is $\boldsymbol{\Omega}_{t+1} = (\boldsymbol{\Omega}_{1,t+1}, \boldsymbol{\Omega}_{2,t+1}, \cdots, \boldsymbol{\Omega}_{C,t+1})$.

To better illustrate the proposed KFL, we summarize the detailed steps of its training process in Algorithm 4. It is worth mentioning that the proposed KFL requires devices to upload the knowledge to the edge server for aggregation instead of the entire local models. Devices' knowledge is generated by averaging the output of their local feature extractor on the data samples from the same class, and the process is irreversible [99]. Thus, KFL is more beneficial for privacy preservation than the model aggregation-based FL algorithms exchanging local models between devices and the edge server. The reason is that the local models are updated according to the devices' private data, whose pattern is encoded into the model parameters. Therefore, if a corresponding decoder could be constructed, the private data or statistics would be recovered inversely [100].

4.2.3 Knowledge-aided Federated Learning Cost Model

In the following, we characterize the learning cost model in each KFL round, including computation cost and communication cost.

1) **Computation Cost**: We consider the CPU adopted to perform training on each device. Denote the CPU clock frequency of device k by f_k (cycles per second). The number of FLOPs per cycle is represented by n_k. Let C_k denote the required number of FLOPs to process one data sample at device k. Consequently, the local training latency of device k is given by

$$\mathcal{T}_k^{\mathrm{L}} = \frac{\tau D_k C_k}{f_k n_k}. \tag{4.10}$$

The corresponding energy consumption of device k is

$$E_k^{\mathrm{L}} = \frac{\kappa \tau D_k C_k f_k^2}{n_k}, \tag{4.11}$$

where κ is the power coefficient, depending on the chip architecture.

2) **Communication Cost**: We consider that the frequency division multiple access is employed for devices to upload their knowledge. The total available wireless bandwidth is BHz. Let $p_{k,t}$ denote the transmit power of device k, its maximum value is $p_{k,\max}$. The channel gain between device k and the edge server is represented by $h_{k,t}$, which considers the path loss and Rayleigh fading. In addition, the channel remains unchangeable within one round but varies independently over rounds. Let $\theta_{k,t} \in [0,1]$ denote the proportion of the overall bandwidth allocated to device k in round t, and $\boldsymbol{\theta}_t = (\theta_{1,t}, \theta_{2,t}, \cdots, \theta_{K,t})$. The uplink rate of device k can be described as $r_{k,t} = \theta_{k,t} B \log_2(1 + \frac{p_{k,t} h_{k,t}}{\theta_{k,t} B N_0})$, where N_0 is the power density of noise. Note that the proposed KFL requires that the knowledge of different devices and classes has the same dimensionality. Thus, the number of parameters in the knowledge of different devices is the same, denoted as Q. Each parameter is quantized by q bits. Thus, the local knowledge uploading latency of device k is

$$\mathcal{T}_{k,t}^{\mathrm{U}} = \frac{Qq}{r_{k,t}} = \frac{Qq}{\theta_{k,t} B \log_2 \left(1 + \frac{p_{k,t} h_{k,t}}{\theta_{k,t} B N_0}\right)}. \tag{4.12}$$

The corresponding energy consumption is

$$E_{k,t}^{\mathrm{U}} = p_{k,t}\mathcal{T}_{k,t}^{\mathrm{U}} = \frac{\theta_{k,t}B\mathcal{T}_{k,t}^{\mathrm{U}}N_0}{h_{k,t}}\left(2^{\frac{Qq}{\theta_{k,t}B\mathcal{T}_{k,t}^{\mathrm{U}}}} - 1\right). \tag{4.13}$$

According to above modes, the energy consumption of device k in round t is $E_{k,t} = E_{k,t}^{\mathrm{L}} + E_{k,t}^{\mathrm{U}}$. Note that we ignore the global knowledge broadcasting and aggregation latency in the above discussion because the broadcasting process occupies the entire bandwidth. The edge server has large transmit power, so the broadcasting latency is negligible. Moreover, the edge server is usually computationally powerful, and the global knowledge aggregation latency can be ignored compared to the above computation and communication latencies.

4.2.4 Problem Formulation

In this work, we aim to improve the learning performance by minimizing the global loss after T rounds, i.e., $F(\boldsymbol{W}_T)$, under the energy budget constraint of devices, where $\boldsymbol{W}_T$ denote the local models in T-th round. Towards this end, we jointly optimize the device scheduling, bandwidth allocation, and power control policies. The optimization problem is given by

$$\min_{\{\boldsymbol{S}_t,\boldsymbol{\theta}_t,\boldsymbol{p}_t\}_{t=0}^{T-1}} \quad F(\boldsymbol{W}_T) \tag{4.14}$$

$$\text{s. t.} \quad \sum_{t=0}^{T-1} E_{k,t} \leq E_k, \forall k \in \mathcal{K}, \tag{4.14a}$$

$$\mathcal{T}_{k,t}^{\mathrm{L}} + \mathcal{T}_{k,t}^{\mathrm{U}} \leq \mathcal{T}_{\max}, \forall k \in \mathcal{K}, \forall t, \tag{4.14b}$$

$$\sum_{k=1}^{K} \theta_{k,t} \leq 1, \forall t, \tag{4.14c}$$

$$0 \leq \theta_{k,t} \leq 1, \forall k \in \mathcal{K}, \forall t, \tag{4.14d}$$

$$\alpha_{k,t} \in \{0,1\}, \forall k \in \mathcal{K}, \forall t, \tag{4.14e}$$

$$0 \leq p_k \leq p_{k,\max}, \forall k \in \mathcal{K}. \tag{4.14f}$$

In problem (4.14), (4.14a) imposes restrictions on the energy consumption of each device k cannot exceed its budget E_k. (4.14b) stipulates that the completion time of each round cannot exceed its maximum allowable delay. (4.14c) indicates that the wireless bandwidth allocated to all devices cannot exceed the total available bandwidth resource. (4.14d) restricts the wireless bandwidth resource allocated to each device. (4.14e) indicates which devices are scheduled in each round.

Solving problem (4.14) requires the explicit form about how device scheduling policy affects the final global loss function. Since it is almost impossible to find an exact analytical expression of $F(\boldsymbol{W}_T)$ with respect to $\boldsymbol{S}_t$ $(t \in \{0, 1, \cdots, T-1\})$, we turn to find an upper bound of $F(\boldsymbol{W}_T)$ and minimize it for the global loss minimization in Section 4.3.1. Moreover, the optimal solution to problem (4.14) requires the system state information of all rounds at the beginning of training. However, such information is unavailable in the practical systems due to the unpredictable time-varying channel condition. To enable online device scheduling, the device scheduling decision should be made at the beginning of each round with only the current state. To this end, we transform the long-term decision problem into a deterministic one with the assistance of the Lyapunov optimization approach in Section 4.3.2.

4.3 Convergence Analysis and Problem Formulation

In this section, we theoretically analyze the convergence bound of the proposed KFL under a non-convex loss function setting. The convergence bound reveals that the scheduled data volume in each round and different learning rounds significantly affect the learning performance. Motivated by this, we define a new metric, i.e., the weighted scheduled data volume, to guide the device scheduling design. Then, we transfer the original problem to maximize this metric for minimizing the gap between the global loss function and the optimal loss. To enable the online dynamic device scheduling under long-term energy budgets constraint, we further transform the problem into a deterministic problem in each round with the assistance of the Lyapunov optimization approach.

4.3.1 Convergence Analysis

In this subsection, we investigate the convergence behavior of the proposed KFL algorithm. To facilitate the analysis, we make the following assumptions on each local loss function $F_k(\cdot)$.

Assumption 4. *All empirical loss functions $F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ ($k \in \mathcal{K}$) are continuously differentiable with respect to $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$, and there exist constants L_u, L_v, L_{uv}, and L_{vu} such that for each $F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$:*

- *$\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ is L_u-Lipschitz continuous with $\boldsymbol{u}_k$ and L_{uv}-Lipschitz continuous with $\boldsymbol{v}_k$, that is,*

$$\left\|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}'_k, \boldsymbol{v}_k)\right\| \leq L_u \left\|\boldsymbol{u}_k - \boldsymbol{u}'_k\right\|, \tag{4.15}$$

and

$$\left\|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}'_k)\right\| \leq L_{uv} \left\|\boldsymbol{v}_k - \boldsymbol{v}'_k\right\|. \tag{4.16}$$

- *$\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ is L_v-Lipschitz continuous with $\boldsymbol{v}_k$ and L_{vu}-Lipschitz continuous with $\boldsymbol{u}$.*

Assumption 5. *The squared norm of gradients is uniformly bounded, i.e., $\|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \leq G_1^2$ and $\|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \leq G_2^2$.*

Assumption 6. *For each local feature extractor $h_k(\cdot)$ ($\forall k \in \mathcal{K}$), its gradient norm is bounded by ϑ^2, i.e., $\|\nabla h_k(\boldsymbol{u}_k)\|^2 \leq \vartheta^2$, and the squared norm of its output vector is bounded by $\|h_k(\boldsymbol{u}_k; x)\|^2 \leq \varsigma^2$.*

Assumption 4 is satisfied by most deep NNs. The modern NNs are usually composed of multiple layers. Based on [87], a deep NN defined by a composition of functions is a Lipschitz NN if the functions in all layers are Lipschitz. It has been proved in [87, 88] that the convolution layer, linear layer, and some nonlinear activation functions (e.g., Sigmoid and tanh) are Lipschitz functions. Thus, most deep NNs have Lipschitz continuous gradients. For a Lipschitz NN in which all layers are Lipschitz functions, both the feature extractor and predictor composed of Lipschitz layers are Lipschitz functions. Thus, Assumption 4 is satisfied by assuming the whole NN to be Lipschitz continuous. In addition, according to Proposition 1 in [87], one can derive that $F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ is $(L_u \times L_v)$-smooth based on Assumption 4. Assumption 5 is widely used in the existing convergence analysis works, e.g., [23, 59, 61, 101]. Assumption 6 is inherently satisfied by Assumption 5 since the gradient of a NN is a function of its output vector. To begin with, we first derive a key lemma to assist our analysis as follows:

Lemma 6. *Let Assumption 4 holds, we have*

$$\begin{aligned} F_k(\boldsymbol{u}'_k, \boldsymbol{v}'_k) - F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) \leq \left\langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{u}'_k - \boldsymbol{u}_k \right\rangle + \frac{1+\chi}{2} L_u \left\| \boldsymbol{u}'_k - \boldsymbol{u}_k \right\|^2 \\ + \left\langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{v}'_k - \boldsymbol{v}_k \right\rangle + \frac{1+\chi}{2} L_v \left\| \boldsymbol{v}'_k - \boldsymbol{v}_k \right\|^2, \quad (4.17) \end{aligned}$$

where $\chi = \max\{L_{uv}, L_{vu}\} / \sqrt{L_u L_v}$*, which measures the relative cross-sensitivity of* $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ *with respect to* $\boldsymbol{v}_k$ *and* $\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ *with respect to* $\boldsymbol{u}_k$*.*

Proof. See Appendix B.1. □

Lemma 6 reveals the gradient relationships of a NN between its feature extractor and label predictor part. According to Lemma 6, we derive the one-round convergence bound of any device k $(k \in \mathcal{K})$ in Lemma 7, in which devices utilize the proposed knowledge-aided loss to update their local models.

Lemma 7. *Let Assumption 4, 5, and 6 hold. The learning rates satisfy* $\eta_u \leq \frac{1}{4\tau(1+\chi)L_u}$ *and* $\eta_v \leq \frac{1}{2\tau(1+\chi)L_v}$*, the one-round convergence bound of device* k $(k \in \mathcal{K})$ *is given by*

$$\begin{aligned} F_k(\boldsymbol{u}_{k,t+1}, \boldsymbol{v}_{k,t+1}) - F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) \leq \left(2(1+\chi) L_u \eta_u^2 \tau^2 - \frac{1}{2}\eta_u \tau \right) \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \\ + \left((1+\chi) L_v \eta_v^2 \tau^2 - \frac{1}{2}\eta_v \tau \right) \|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 + A_1 + \frac{5}{4}\eta_u \lambda^2 \sum_{l=0}^{\tau-1} \|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2 \\ + 2\eta_u^2 \lambda^2 (3\eta_u L_u^2 + 2\eta_v \chi^2 L_u L_v) \sum_{l=0}^{\tau-1} (\tau - l) \left\|\nabla L_k(\boldsymbol{u}_{k,t,l})\right\|^2, \quad (4.18) \end{aligned}$$

where $A_1 = \tau(\tau+1)(2\tau+1) \left(\eta_u^3 G_1^2 L_u^2 + \frac{1}{3}\eta_v^3 G_2^2 L_v^2 + (\frac{2}{3}\eta_u G_1^2 + \frac{1}{2}\eta_v G_2^2)\eta_u \eta_v \chi^2 L_u L_v \right)$.

Proof. See Appendix B.2. □

Based on Lemma 7, we further analyze the convergence behaviour of the proposed KFL algorithm after T rounds in Theorem 2, which takes into account the knowledge aggregation between devices.

Theorem 2. *Let Assumption 4, 5, and 6 hold, $\eta_u \leq \frac{1}{4\tau(1+\chi)L_{\boldsymbol{u}}}$ and $\eta_v \leq \frac{1}{2\tau(1+\chi)L_v}$, the gap between the global loss function after T rounds and the optimal loss is bounded by*

$$\begin{aligned} F(\boldsymbol{W}_T) - F(\boldsymbol{W}^*) \leq{}& A_3^T(F(\boldsymbol{W}_0) - F(\boldsymbol{W}^*)) \\ &+ \frac{1 - A_3^T}{1 - A_3}(A_1 + A_2) + A_2 \frac{CK}{D} \sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} \sum_{c=1}^{C} \frac{D_{k,c}^2}{D_c^2 D_k} \sum_{k=1}^{K} D_{k,c}^2 \\ &- A_2 \frac{1}{DK(T-1)} \frac{1}{\max_{1\leq k\leq K} D_k} \left(\sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} \alpha_{k,t} D_k \right)^2, \end{aligned} \tag{4.19}$$

where $A_2 = 10\eta_u \lambda^2 \tau \vartheta^2 \varsigma^2 + 8\eta_u^2 \lambda^2 \vartheta^2 \varsigma^2 \left(3\eta_u L_u^2 + 2\eta_v \chi^2 L_u L_v\right) \tau(\tau+1)$, $A_3 = 1 + (4L_u^2\eta_u^2 + 2L_v^2\eta_v^2)(1+\chi)\tau^2 - (\eta_u L_u + \eta_v L_v)\tau$.

Proof. See Appendix B.3. □

Theorem 2 reveals how the device scheduling policy affects the convergence bound of KFL without characterizing the impact of non-IID degrees on the convergence bound. In general, the non-IID degree is characterized by the difference between the optimal global loss and the weighted summation of optimal local losses [102]. However, the proposed KFL is a personalized FL algorithm which trains a personalized model for each device. Thus, one cannot characterize the impacts of non-IID degree on the convergence bound in this way due to $F(\boldsymbol{W}^*) - \frac{1}{D}\sum_{k=1}^{K} D_k F_k(\boldsymbol{w}_k^*) = 0$. However, how to characterize non-IID degrees' effects on the convergence bound of personalized FL algorithms is a promising research direction, which will be studied in our future works.

According to Theorem 2, the gap between the global loss after T rounds and the optimal loss is bounded by four terms, 1) the gap in the initial round, 2) two terms related to hyperparameters of the learning system, 3) the scheduled data volume in all rounds. It is noted that $A_3 \leq 1$ due to $\eta_u \leq \frac{1}{4\tau(1+\chi)L_u}$ and $\eta_v \leq \frac{1}{2\tau(1+\chi)L_v}$. As T increases, A_3^T approaches to 0. Hence, the first term converges to 0, and the second and the third terms converge to a constant. The first three terms decided by the system hyperparameters and initial models of devices are not related to the device scheduling policies. The last term is an explicit form related to device scheduling. For the last term, we have the following remark:

Remark 3. *Increasing the scheduled data samples in each round is able to narrow the gap between global loss and optimal loss. In addition, as t increases, A_3^{T-1-t} also increases due to $A_3 < 1$. This indicates that more devices should be scheduled in early rounds when the total number of scheduled devices in the learning process is fixed.*

Note that, it has been experimentally observed in [24] that scheduling more devices in the later rounds is beneficial for the learning performance of the federated averaging algorithm. However, the proposed KFL that only aggregates devices' knowledge in each round achieves better learning performance when scheduling more devices in the earlier rounds, which is verified by the theoretical analysis in Remark 3 and experimental results in Section 4.5.

4.3.2 Problem Transformation via Lyapunov Optimization Framework

According to Theorem 2, the gap between the global loss and the optimal loss can be narrowed by minimizing the last term on the right-hand-side (RHS) of (4.19). However, it is tractable to directly minimize this term since it involves some unknown parameters, e.g., the Lipschitz constant L_u and L_v. Based on [88], the exact computation of the Lipschitz constant of deep learning architectures is intractable, even for two-layer NNs. Inspired by Remark 3, to enable tractable device scheduling design, we introduce a variable γ_t $(t = 0, 1, \cdots, T-1)$ as the weight of scheduled data samples in round t to capture the varying significance of scheduling devices in different rounds. Based on this, we define the weighted scheduled data volume as $\sum_{t=0}^{T-1} \gamma_t \sum_{k=1}^{K} \alpha_{k,t} D_k$ and maximize it for the global loss minimization. Thus, we transform problem (4.14) as the following problem:

$$\max_{\{\boldsymbol{S}_t,\boldsymbol{\theta}_t,\boldsymbol{p}_t\}_{t=0}^{T-1}} \quad \sum_{t=0}^{T-1}\gamma_t\sum_{k=1}^{K}\alpha_{k,t}D_k \tag{4.20}$$
$$\text{s. t.} \quad (4.14a),(4.14b),(4.14c),(4.14d),(4.14e).$$

Problem (4.20) involves multi-dimension discrete and continuous variables is a typical mixed-integer programming problem, which is generally NP-Hard. In addition, solving the optimal solution of problem (4.20) offline requires optimally dividing the energy of all devices in each round due to the long-term energy constraints, which is intractable. The most critical challenge of directly solving problem (4.20) is that it requires channel information of all devices over all rounds at the beginning of the FL process, which may unfeasible in practical systems. To enable the online dynamic device scheduling, we utilize the Lyapunov optimization framework to deal with the correlations among rounds. To this end, we construct a virtual queue $q_k(t)$ for each device k $(k \in \mathcal{K})$, which evolves as

$$q_k(t+1) = \max\left\{q_k(t) + \alpha_{k,t}E_{k,t} - \frac{E_k}{T}, 0\right\}, \tag{4.21}$$

with the initial value $q_k(t) = 0$ for all devices. Inspired by the drift-plus-penalty algorithm in [103], we transform problem (4.20) as the following problem to enable online device scheduling

$$\min_{\{\boldsymbol{S}_t,\boldsymbol{\theta}_t,\boldsymbol{p}_t\}_{t=0}^{T-1}} \quad -V\gamma_t\sum_{k=1}^{K}\alpha_{k,t}D_k + \sum_{k=1}^{K}q_k(t)\alpha_{k,t}E_{k,t} \tag{4.22}$$
$$\text{s. t.} \quad (4.14b),(4.14c),(4.14d),(4.14e).$$

In problem (4.22), $V \geq 0$ is a weight factor that balances the energy consumption of devices and learning performance. A large V emphasises the learning performance improvement by sacrificing the devices' energy and vice versa. In addition, from the objective function (4.22), the unscheduled devices in the former rounds have smaller $q_k(t)$. These devices are encouraged to participate in the current round of training for minimizing (4.22). Thus, problem (4.22) contributes to a fair scheduling scheme between devices.

4.4 Online Device Scheduling and Wireless Resource Allocation

In this section, we propose an energy-aware device scheduling, bandwidth allocation, and power control algorithm that solves problem (4.22) in an online fashion. We first derive the optimal bandwidth allocation and power control policies using convex optimization techniques. Then, we propose a polynomial-time algorithm to solve the device scheduling decision with a $\mathcal{O}(\sqrt{V}, 1/V)$ energy-learning trade-off guarantee, where V is an algorithm-related parameter.

4.4.1 Optimal Power Control and Bandwidth Allocation

For any given scheduled device set $\boldsymbol{S}_t \in \mathcal{K}$, we decompose the bandwidth allocation and power control problem from (4.22) as follows:

$$\begin{aligned} \min_{\{\boldsymbol{\theta}_t, \boldsymbol{p}_t\}} \quad & \sum_{k \in \boldsymbol{S}_t} q_k(t) E_{k,t} \\ \text{s. t.} \quad & (4.14\text{b}), (4.14\text{c}), (4.14\text{d}). \end{aligned} \tag{4.23}$$

For problem (4.23), we have the following proposition:

Proposition 1. *The optimal solution of problem* (4.23) *satisfies* $\mathcal{T}_{k,t}^{\mathrm{U}} = \mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}}$, *and the optimal transmit power of device k satisfies*

$$p_{k,t} = \frac{\theta_{k,t} B N_0}{h_{k,t}} \left(2^{\frac{Qq}{(\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}})\theta_{k,t} B}} - 1 \right). \tag{4.24}$$

Proof. See Appendix B.4. □

According to Proposition 1, we substitute (4.24) into problem (4.23), the optimal bandwidth allocation problem can be formulated as

$$\begin{aligned} \min_{\boldsymbol{\theta}_t} \quad & \sum_{k \in \boldsymbol{S}_t} \frac{\theta_{k,t} B N_0 q_k(t) (\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}})}{h_{k,t}} \mathcal{I}(\theta_{k,t}) \\ \text{s. t.} \quad & (4.14\text{c}), (4.14\text{d}), \end{aligned} \tag{4.25}$$

$$\theta_{k,t} B \log\left(1+\frac{p_{k,\max} h_{k,t}}{\theta_{k,t} B N_0}\right) \geq \frac{Qq}{(\mathcal{T}_{\max}-\mathcal{T}_k^{\mathrm{L}})}, \tag{4.25a}$$

where

$$\mathcal{I}(\theta_{k,t}) = \exp\left(\frac{Qq\ln 2}{(\mathcal{T}_{\max}-\mathcal{T}_k^{\mathrm{L}})\theta_{k,t} B}\right) - 1. \tag{4.26}$$

For problem (4.25), we obtain its optimal solution by using the following lemma.

Lemma 8. *The optimal bandwidth allocation of problem (4.25) satisfies*

$$\theta_{k,t}^{*} = \max\left\{\theta_{k,t}(\mu), \theta_{k,t}^{\min}\right\}, \tag{4.27}$$

where

$$\theta_{k,t}(\mu) = \frac{Qq\ln 2}{(\mathcal{T}_{\max}-\mathcal{T}_k^{\mathrm{L}})B\left(\mathcal{W}\left(\frac{\mu h_{k,t}}{eBN_0 q_k(t)(\mathcal{T}_{\max}-\mathcal{T}_k^{\mathrm{L}})}-\frac{1}{e}\right)+1\right)}, \tag{4.28}$$

and $\theta_{k,t}^{\min}$ *satisfies constraint (4.25a),* μ *is the Lagrange multiplier which satisfies* $\sum\limits_{k=1}^{K}\theta_{k,t}(\mu^{*}) = 1$*.* $\mathcal{W}(\cdot)$ *is the principal branch of the Lambert function, defined as the solution of* $\mathcal{W}(x)e^{\mathcal{W}(x)} = x$*, in which* e *is the Euler's number.*

Proof. See Appendix B.5. □

Although Lemma 8 provides the optimal condition of bandwidth allocation, there is still an unknown variable μ. Below we develop a binary search method to solve the optimal μ. Since the Lagrange multiplier $\mu \geq 0$, we have $\frac{\mu h_{k,t}}{eBN_0 q_k(t)\left(\mathcal{T}_{\max}-\mathcal{T}_k^{\mathrm{L}}\right)} - \frac{1}{e} \geq -\frac{1}{e}$. Moreover, $\mathcal{W}(x)$ is a monotonically increasing function when $x \geq -\frac{1}{e}$. Thus, $\theta_{k,t}(\mu)$ is a monotonically decreasing function with respect to μ. To deploy the binary search method, we derive the lower and upper bound of μ. Since $\mu \geq 0$, the lower bound of μ is $\mu_{\mathrm{lb}} = 0$. For the upper bound, we have $\max_{\boldsymbol{S}_t}\{\theta_{k,t}(\mu)\} \geq \frac{1}{|\boldsymbol{S}_t|}$. Let $\varphi_k = \frac{Qq|\boldsymbol{S}_t|\ln 2}{\left(\mathcal{T}_{\max}-\mathcal{T}_k^{\mathrm{L}}\right)B}$. Based on the definition of Lambert function, we have

$$\mu \leq \mu_{\mathrm{ub}} = \max_{k\in\boldsymbol{S}_t}\left\{\frac{BN_0 q_k(t)(\mathcal{T}_{\max}-\mathcal{T}_k^{\mathrm{L}})\left((\varphi_k-1)e^{\varphi_k}+1\right)}{h_{k,t}}\right\}. \tag{4.29}$$

Based on the lower bound μ_{lb} and upper bound μ_{ub}, the optimal Lagrange multiplier can be obtained by the binary search method. For clarity, we summarize the detailed steps for solving the optimal bandwidth allocation policy in Algorithm 5. The binary search method halves the search region at every iteration and terminate when the given

Algorithm 5 Optimal Bandwidth Allocation Approach

```
Initialize $\boldsymbol{S}_t$, the precision requirement $\varepsilon > 0$.
Initialize the upper bound of Lagrange multiplier $\mu_{\text{ub}}$ based on (4.29), set the lower bound
to $\mu_{\text{lb}} = 0$.
repeat
    Set $\mu = (\mu_{\text{lb}} + \mu_{\text{ub}})/2$.
    For each device $k \in \boldsymbol{S}_t$, compute the required bandwidth allocation ratio $\theta_{k,t}(\mu)$ based on
    (4.28).
    Compute the summation of required bandwidth allocation ratio $\sum_{k\in\boldsymbol{S}_t} \theta_{k,t}(\mu)$.
    if $\sum_{k\in\boldsymbol{S}_t} \theta_{k,t}(\mu) > 1$ then
        Halve the searching region by setting $\mu_{\text{lb}} = \mu$ and $\mu_{\text{ub}} = \mu_{\text{ub}}$.
    else if $0 < \sum_{k\in\boldsymbol{S}_t} \theta_{k,t}(\mu) < 1 - \varepsilon$ then
        Halve the searching region by setting $\mu_{\text{lb}} = \mu_{\text{lb}}$ and $\mu_{\text{ub}} = \mu$.
    else
        Break the circulation.
    end if
until $|\mu_{\text{ub}} - \mu_{\text{lb}}| < \varepsilon$
Substituting $\mu$ into (4.28) for get $\theta_{k,t}(\mu)$, then compute the optimal bandwidth allocation
policy based on (4.27).
Substitute the optimal bandwidth allocation policy into (4.24) for obtaining the optimal
power control policy.
return The optimal bandwith allocation policy $\boldsymbol{\theta}_t$, the optimal power control policy $\boldsymbol{p}_t$.
```

precision (i.e., ε) requirement is satisfied. Thus, the time complexity of this method is $\mathcal{O}\left(\log_2 \frac{\mu_{\text{ub}}-\mu_{\text{lb}}}{\varepsilon}\right)$.

4.4.2 Device Scheduling

Based on the above analysis, the optimal bandwidth allocation and power control policy for any device scheduling set $\boldsymbol{S}_t$ can be obtained by using Algorithm 5. For device scheduling design, an intuitive method is to compute the objective function value for all possible device scheduling decisions, and select the one with minimal objective function as the final scheduling decision. However, this intuitive method is infeasible in its implementation since there are $\sum_{n=0}^{K} C_K^n = 2^K$ possible scheduling decisions, inducing an exponential time complexity with $\mathcal{O}\left(2^K \log_2 \frac{\mu_{\text{ub}}-\mu_{\text{lb}}}{\varepsilon}\right)$. In the following part, we develop an efficient algorithm to solve the device scheduling policy.

According to the objective function (4.22), it is desirable to select devices with small $q_k(t)$ and $E_{k,t}$, as well as large data samples. The small $E_{k,t}$ is achieved by strong channels and low computation energy consumption. To identify these devices, we first allocate equal bandwidth to all devices (i.e., $\theta = 1/K$), and then substitute $\mathcal{T}_{k,t}^{\mathrm{U}} = \mathcal{T}_{\max} - \mathcal{T}_{k}^{\mathrm{L}}$ into (4.13) to compute the estimated energy consumption $\bar{E}_{k,t} = E_{k,t}^{\mathrm{L}} + E_{k,t}^{\mathrm{U}}$. Based on the estimated energy consumption for all devices, we sort devices based on $\Delta_{k,t} = -V\gamma_t D_{k,c} + q_k(t)\bar{E}_{k,t}$ $(\forall k \in \mathcal{K})$ in the ascending order. Denote $\widetilde{\mathcal{K}}$ as the sorted device set. Many sorting algorithms, such as Heapsort or Mergesort, can be used, with a worst-case complexity $\mathcal{O}(K \log K)$. Then, we solve the device scheduling policy by incrementally adding devices into the selection set $\boldsymbol{S}$ from the sorted device set $\widetilde{\mathcal{K}}$. For each possible device scheduling set $\boldsymbol{S}$, we perform Algorithm 5 to obtain the optimal wireless bandwidth allocation $\boldsymbol{\theta}_t^*(\boldsymbol{S})$, power control decisions $\boldsymbol{p}_t^*(\boldsymbol{S})$, as well as the optimal energy consumption $E_t^*(\boldsymbol{S})$. Substituting $E_t^*(\boldsymbol{S})$ into (4.22), the drift-plus-penalty value of device scheduling set $\boldsymbol{S}$ can be obtained, denoted as $\mathcal{Y}(\boldsymbol{S})$. Let $\mathcal{H}$ denote the set of all possible device scheduling set $\boldsymbol{S}$. Finally, we obtain the optimal device scheduling policy through comparing the drift-plus-penalty value of all possible device scheduling set $\boldsymbol{S} \in \mathcal{H}$, i.e., $\boldsymbol{S}_t^* = \arg\min_{\boldsymbol{S}\in\mathcal{H}} \mathcal{Y}(\boldsymbol{S})$. Note that, the energy consumption of devices with $q_k(t) = 0$ does not affect the objective function value, the minimal required bandwidth should be allocated to them for saving more bandwidth for other users with $q_k(t) > 0$. For clarity, we summarize the detail steps of device scheduling algorithm in Algorithm 6, which obtains the device scheduling policy by performing at most K times Algorithm 5 and has a polynomial time complexity $\mathcal{O}\left(K \log_2 \frac{\mu_{\mathrm{ub}}-\mu_{\mathrm{lb}}}{\varepsilon}\right)$.

For Algorithm 6, we analyze its performance by comparing it with its optimal offline counterpart which is in fact problem (4.20). The offline algorithm has the channel information of all rounds. Let $\alpha_{k,t}^*$ be the offline optimal device scheduling decision obtained by solving the problem (4.20) with pre-known device information. The performance guarantee of the proposed device scheduling algorithm is shown in Proposition 2.

Proposition 2. *Compared to the offline optimal solution, the cumulative loss of Algorithm 6 is bounded by*

$$\sum_{t=0}^{T-1} \gamma_t \sum_{k=1}^{K} \alpha_{k,t} D_{k,c} \geq -\frac{T\zeta_0}{V} - \frac{T(T-1)}{2V} \sum_{k=1}^{K} \zeta_k^2 + \sum_{t=0}^{T-1} \gamma_t \sum_{k=1}^{K} \alpha_{k,t}^* D_{k,c}, \tag{4.30}$$

Algorithm 6 Energy-aware online Device scheduling

```
Input the virtual queue length $q_k(t)$ $(k \in \mathcal{K})$ and $\gamma_t$, initialize $V$.
Substituting $\theta_{k,t} = 1/K$ and $\mathcal{T}^{\mathrm{U}}_{k,t} = \mathcal{T}_{\max} - \mathcal{T}^{\mathrm{L}}_k$ into (4.13) to compute the estimated energy consumption of device $k$ $(\forall k \in \mathcal{K})$, i.e., $\bar{E}_{k,t} = E^{\mathrm{L}}_{k,t} + E^{\mathrm{U}}_{k,t}$.
Sort devices based on $\Delta_{k,t} = -V\gamma_t D_{k,c} + q_k(t)\bar{E}_{k,t}$ in the ascending order to obtain the sorted device set $\widetilde{\mathcal{K}}$.
for $k = \widetilde{\mathcal{K}}(1), \widetilde{\mathcal{K}}(2), \cdots, \widetilde{\mathcal{K}}(K)$ do
    Update $\boldsymbol{S} = \boldsymbol{S} \cup \{k\}$
    Solve the optimal bandwidth allocation and power control policy by Algorithm 5, i.e., $\boldsymbol{\theta}^*_t(\boldsymbol{S})$ and $\boldsymbol{p}^*_t(\boldsymbol{S})$.
    Compute the drift-plus-penalty of $\boldsymbol{S}$, i.e., $\mathcal{Y}(\boldsymbol{S}) = -V\gamma_t \sum_{k\in\boldsymbol{S}} D_k + \sum_{k\in\boldsymbol{S}} q_k(t)E_{k,t}$
    if $-VD_k + q_k(t)E_{k,t} > 0$ then
        Break the circulation
    else
        Add $\boldsymbol{S}$ into $\mathcal{H}$, i.e., $\mathcal{H} = \mathcal{H} \cup \boldsymbol{S}$
    end if
end for
Find the optimal device scheduling set $\boldsymbol{S}^*_t = \arg\min_{\boldsymbol{S}\in\mathcal{H}} \mathcal{Y}(\boldsymbol{S})$.
return The device scheduling set $\boldsymbol{S}^*_t$, wireless bandwidth allocation $\boldsymbol{\theta}^*_t(\boldsymbol{S}^*_t)$, and power control policy $\boldsymbol{p}^*_t(\boldsymbol{S}^*_t)$.
```

and the total energy consumption of Algorithm 6 is bounded by

$$\sum_{t=0}^{T-1}\sum_{k=1}^{K} \alpha_{k,t}E_{k,t} \leq \sum_{k=1}^{K} E_k + \sqrt{2K\left(T\zeta_0 + V\sum_{t=0}^{T-1}\gamma_t D\right)}, \tag{4.31}$$

where $\zeta_0 = \frac{1}{2}\sum_{k=1}^{K}\zeta_k^2$ *and* $\zeta_k = \max_t\left\{\left|\alpha_{k,t}E_{k,t} - \frac{E_k}{T}\right|\right\}$.

Proof. See Appendix B.6. □

Proposition 2 characterizes the performance of the proposed device scheduling algorithm, which shows that 1) the energy constraints of devices are approximately satisfied with the $\mathcal{O}(\sqrt{V})$-bounded factor, and 2) the proposed device algorithm is $\mathcal{O}(1/V)$-optimal with respect to the performance of its optimal offline counterpart solution. Thus, the proposed device scheduling algorithm demonstrates an $\mathcal{O}(\sqrt{V}, 1/V)$ energy-learning trade-off. The worst-case performance of Algorithm 6 can be improved by reducing the upper bound of energy usage bias ζ_0. In addition, adjusting the weight parameter V is able to achieve the balance between the learning performance and energy consumption of devices. Specifically, with larger V, more emphasis is put on the scheduled data samples

Table 4-A: Experimental Settings

Parameter	Value	Parameter	Value
K	100	$p_{k,\max}$ $(\forall k \in \mathcal{K})$	30 dBm
B	5 MHz	N_0	-174 dBm
q	32 bits	h_0	-30 dBm
η_u	0.05	η_v	0.05
τ	5	m	2
d_0	1 m	v	2
d_{MLP}	256	d_{CNN}	128
E_k (MLP)	$0.1 \times T$ J	$T_{\max}$ (MLP)	1 s
E_k (CNN)	$0.5 \times T$ J	$T_{\max}$ (CNN)	2 s

to improve the learning performance while more energy is consumed at devices, and vice versa. In practical systems, one should carefully select the value of V to optimize the learning performance with energy limits and use the energy in a balanced manner to avoid large ζ_k.

4.5 Numerical Results

In this section, we verify the effectiveness of the proposed KFL algorithm. If not specified, the simulation settings are listed in Table 4-A. We consider K devices randomly distributed in a cell with a radius of 500m, and the server is located at the centre of the cell. Similar to [98], the channel gain is modelled as $h_{k,t} = h_0 \rho_k(t)(d_0/d_k)^v$, where h_0dBm is the path loss constant; d_k is the distance between device k and the edge server; d_0m is the reference distance; $\rho_k(t) \sim \text{Exp}(1)$ is exponentially distributed with unit mean, which represents the small-scale fading channel power gain from the device k to the edge server in round t; d_0/d_k represents the large-scale path loss with v being the path loss exponent. For all devices in the system, their CPU frequency are randomly selected from $\{0.85, 1.12, 1.2, 1.3\}$ GHz [104, 105].

We evaluate the proposed KFL algorithm on two image classification tasks using MNIST and CIFAR-10 datasets, both of them have 10 classes of data samples. For the MNIST dataset, we train five-layer MLP models with the following architecture: four fully connected layers with 784, 512, d_{MLP}, 64 units, each of these layers is activated by the ReLU function; and a 10-unit softmax output layer. For the CIFAR-10 dataset, we train five-layer CNN models with the following structure: two 5×5 convolution layers followed by a 2×2 max-pooling layer, in which the first convolution layer possesses 6 channels and the second layer with 16 channels; three fully connected layers with 400, d_{CNN}, and 64 units, respectively; and a 10-unit softmax output layer. The ReLU function activates each convolution or fully connected layer. When devices are equipped with homogeneous models, we set $d_{\text{MLP}} = 256$ and $d_{\text{CNN}} = 128$. The number of FLOPs and parameters of these machine learning models can be estimated using the method in [106]. Specifically, the MLP with $d_{\text{MLP}} = 256$ possesses 553406 parameters which equal the FLOPs required to process one data sample. The CNN with $d_{\text{CNN}} = 128$ has 63106 parameters and requires 1245834 FLOPs to process one data sample. When devices have heterogeneous models, the value of d_{MLP} and d_{CNN} for all devices are randomly selected from $\{128, 192, 256, 320, 384\}$. For both MLP and CNN, a momentum of 0.9 is adopted, and cross-entropy is adopted as the loss function. In addition, we first classify the training data samples according to their labels, then split each class of data samples into $mK/10$ shards, and finally randomly distribute two shards of data samples to each device. That is, each client has a data distribution corresponding to at most m classes. The data distributions among devices are more skewed for smaller m. In the simulations, each device first computes the number of correct predicted data samples on its test dataset by its local model. Note that the deployed trained models on devices are the same in FedAvg, while each device has a personalized local model in the proposed KFL. Then, the test accuracy is computed as the total number of correct predicted test data samples on all devices divided by the total number of test data samples on all devices.

In the following sections, we verify the theoretical results in Remark 3 on MNIST and CIFAR-10 datasets by comparing the learning performance of the following three temporal device scheduling patterns. 1) Uniform Scheduling: Ten devices are randomly scheduled in each round to participate in the learning process. 2) Ascend Scheduling:

The number of scheduled devices increases from 1 to 20, with an average number of 10 devices scheduled in each round. 3) Descend Scheduling: The number of scheduled devices decreases from 20 to 1, with an average number of 10 devices per round.

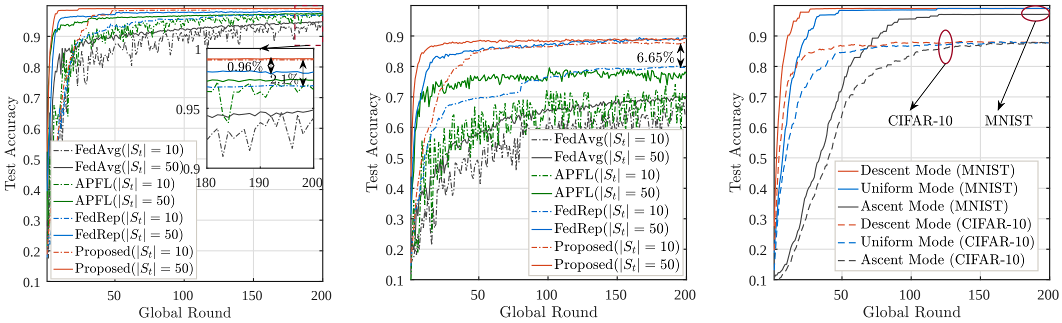
4.5.1 Performance Evaluation with Homogeneous Models

We evaluate the performance of the proposed KFL algorithm by comparing it with the following benchmarks. Note that, devices are equipped with homogeneous local models, and we do not consider the energy and bandwidth limitation in this subsection. 1) FedAvg [7]: In each round, the scheduled devices upload their model parameters to the edge server for aggregation. 2) FedRep [107]: The scheduled devices sequentially train the feature extractor and label predictor parts of their models in each round. Particularly, the scheduled devices only upload feature extractor part of their models to the edge server for aggregation. 3) APFL [108]: In each round, each scheduled device trains its own local model and the received global model from the edge server. Then APFL incorporates the devices' locally trained model and the updated global model to achieve a device-specific model. It is worth mentioning that the proposed KFL algorithm requires fewer parameters transmission in each round than the benchmarks and thus reduces the communication cost, as shown in Table 4-B. Specifically, for the MNIST dataset, the proposed algorithm only requires devices to upload the knowledge of 10 classes, including $64 \times 10 = 640$ parameters, accounting for 0.1% of the transmitted parameters by FedAvg or by APFL. For the CIFAR-10 dataset, the KFL algorithm only requires devices to upload 640 parameters in each round, comprising 1.0% of the total model parameters.

Fig. 4.2 shows the learning performance of the proposed KFL algorithm and two benchmarks on MNIST and CIFAR-10 datasets. From Fig. 4.2(a), compared to the state-of-art FedRep, it is observed that the proposed algorithm achieves 0.96% accuracy improvement when 50 devices participate in each round learning process and obtains a 2.1% accuracy gain when scheduling 10 devices in each round. In addition, the proposed

Table 4-B: Communication Overhead of Different FL Algorithms

Dataset	Algorithm	Number of Transmitted Parameters	Saved Communication Overhead
MNIST	Proposed	640	**99.9%**
	FedRep [107]	533248	3.6%
	FedAvg [7]	553406	0%
	APFL [108]	553406	0%
CIFAR-10	Proposed	640	**99%**
	FedRep [107]	54200	14.1%
	FedAvg [7]	63106	0%
	APFL [108]	63106	0%



(a) (b) (c)

Figure 4.2: Comparison of learning performance under homogeneous models: (a) different algorithms on the MNIST dataset; (b) different algorithms on the CIFAR-10 dataset; (c) different scheduling patterns on MNIST and CIFAR-10 datasets.

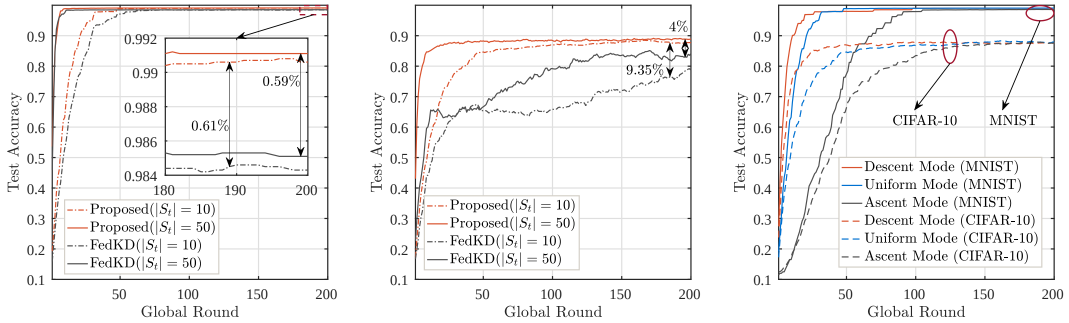
algorithm converges faster than the benchmarks. A similar experiment conducted on the CIFAR-10 dataset is shown in Fig. 4.2(b). Similar to the results on the MINIST dataset, the proposed algorithm outperforms the benchmarks. Specifically, it improves 6.65% accuracy when 10 devices are scheduled in each round. Although the proposed algorithm has similar accuracy to the FedRep when scheduling 50 devices in each round, it converges faster than the latter.

Fig. 4.2(c) presents the test accuracy of the proposed KFL algorithm with different device scheduling patterns on MNIST and CIFAR-10 datasets. It is observed that the descend scheduling pattern converges faster than the other two scheduling patterns on these two datasets. This experimental result verifies the theoretical results in Remark 3, which indicates that more scheduled data volume should bias to the early rounds if the entire scheduled data volume are fixed.

4.5.2 Performance Evaluation with Heterogeneous Models

In this subsection, we verify the effectiveness of the proposed KFL algorithm by comparing it with the FedKD [109], which is a knowledge distillation-based FL algorithm. Note that in this subsection, devices are equipped with heterogeneous local models, and do not consider the energy and bandwidth limitations. Since the knowledge distillation process requires aggregating devices' model output logits on an additional proxy dataset, we sample 50 data samples from each class (for both MNIST and CIFAR-10) to construct the proxy dataset with 500 data samples. Note that as stated in the experimental setting, our proposed algorithm only requires devices to upload 640 parameters in each round, reducing 87% of transmission costs compared with the knowledge distillation-based algorithm because the latter requires devices to upload $500 \times 10 = 5000$ parameters in each round.

Fig. 4.3 presents the test accuracy of the proposed KFL algorithm and the knowledge distillation-based FL algorithm under heterogeneous devices' models. Fig. 4.3(a) shows the results of the MNIST dataset. Compared to the FedKD algorithm, the proposed KFL algorithm achieves a slight test accuracy improvement, i.e., 0.61% when 10 devices and 0.59% when 50 devices participate in the per-round training. In addition, the proposed KFL algorithm convergences faster than the FedKD algorithm. Fig. 4.3(b) presents the results of the CIFAR-10 dataset which is more complex than the MNIST dataset. The proposed KFL algorithm obtains a more distinct learning performance gain on the CIFAR-10 dataset, i.e., compared to FedKD, improving 4% and 9.35% accuracy when 10 and 50 devices participate in per-round training, respectively. In fact, the learning performance of FedKD or other knowledge distillation-based FL algorithms heavily relies on the quality of the proxy dataset. In practical applications, the additional proxy dataset may not always be available, and its quality is usually not very high. Thus, the proposed KFL algorithm is flexible for practical scenarios. Fig. 4.3(c) shows a similar



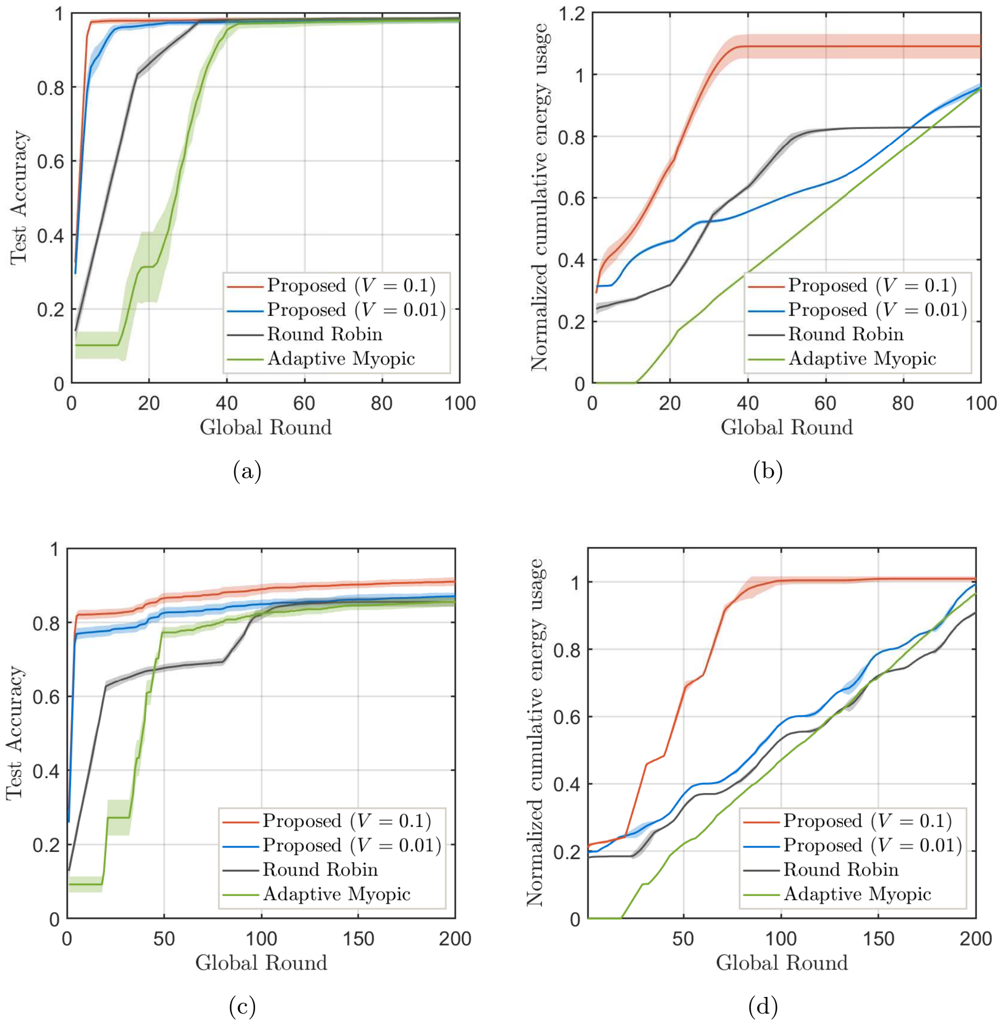
(a) (b) (c)

Figure 4.3: Comparison of learning performance under heterogeneous models (a) different algorithms on the MNIST dataset; (b) different algorithms on the CIFAR-10 dataset; (c) different scheduling patterns on MNIST and CIFAR-10 datasets.

result to the experiment under homogeneous models, indicating that more data sample volume should be scheduled in the earlier rounds when the total scheduled data volume in the entire learning course are fixed.

4.5.3 Performance of the Proposed Device Scheduling Algorithm

This subsection evaluates the proposed device scheduling algorithm in the wireless network by comparing it with two benchmarks. Note that devices in this subsection are equipped with heterogeneous models. 1) Round Robin Scheduling Policy [110]: In each round, the round robin policy selects a set of devices with the size of 5 (for both MNIST and CIFAR-10 dataset) that have sufficient energy to support its current local training and knowledge uploading to participate in the training process. This policy contributes a fairness scheduling among devices. The size of the scheduled device set of the round robin is determined by the maximum overall scheduled devices of other scheduling algorithms divided by the number of rounds. 2) Myopic Scheduling Policy [111]: For each device k, the available energy in round t is given by the remaining energy divided by the remaining number of rounds, i.e., $\frac{E_k - \sum_{t'=0}^{t-1} \alpha_{k,t'} E_{k,t'}}{T-t+1}$. Note that, in this subsection, devices are equipped with heterogeneous models. For the proposed algorithm, we set



(a)

(b)

(c)

(d)

Figure 4.4: Comparison of learning performance in different device scheduling algorithms on the MNIST and CIFAR-10 datasets.

$\gamma_t = \frac{1}{t}$ $(\forall t \in \{0, 1, \cdots, T-1\})$. In addition, to compare the energy usage in the training process, we define the normalized cumulative energy usage as the maximal value of the proportion of consumed energy to the overall energy across devices till t-th round, i.e., $\max_{k\in\mathcal{K}} \frac{\sum_{t'=1}^{t} \alpha_{k,t'} E_{k,t'}}{E_k}$.

Fig. 4.4(a) and Fig. 4.4(b) compare the test accuracy and normalized cumulative energy usage of the scheduling algorithms on the MNIST dataset. It is observed from Fig. 4.4(a) that the proposed algorithm obtains a faster convergence speed and higher test accuracy than the benchmarks. From Fig. 4.4(b), we can see that the proposed scheduling algorithm with $V = 0.01$ and $V = 0.1$ have higher energy usage in the beginning 30 rounds than benchmarks. This induces a faster convergence speed of the proposed algorithm. Particularly, the proposed algorithm with $V = 0.01$ has the same energy usage as the Adaptive Myopic algorithm. Both satisfy the energy constraints of devices (at the end of the training process, the normalized cumulative energy usage is smaller than 1). However, the proposed algorithm with $V = 0.01$ achieves better learning performance. This performance gain comes from the proposed algorithm enabling devices to use energy more flexibly, thus improving the training performance.

Similar comparison is made on CIFAR-10 dataset in Fig. 4.4(c) and Fig. 4.4(d). It is also observed that the proposed online device scheduling algorithm outperforms the baselines in accuracy and convergence speed. From Fig. 4.4(d), we can see that the proposed algorithm enables devices to consume more energy in the earlier rounds compared to the baselines, which indicates that the proposed algorithm schedules more data samples in the early rounds. Thus, based on Remark 3, the proposed algorithm obtains better learning performance than the baselines. Particularly, the proposed algorithm with $V = 0.1$ enables devices to exhaust their energy in the former 100 rounds and achieve the best learning performance. The round robin algorithm enables devices to consume energy uniformly throughout the process. While for the Adaptive Myopic algorithm, the energy consumption at the former rounds exceeds the budget, and thus no devices are scheduled. In fact, the proposed algorithm schedules devices in the descend scheduling pattern, while Adaptive Myopic schedules devices in the ascend scheduling pattern and Round Robin schedules devices in the uniform scheduling pattern. Thus, the result in Fig. 4.4(a) and 4.4(c) also verified the correctness of our theoretical analysis in Remark 3, i.e., more data samples should be scheduled in the early rounds under restricted resources budgets.

4.6 Summary

In this chapter, we have developed a novel KFL framework which aggregates devices' knowledge to enable collaborative training between devices. The benefits of this framework are three folds: 1) Allowing devices with heterogeneous models to train machine learning models collaboratively. 2) Significantly reducing the communication overhead of devices compared to conventional model aggregation-based FL approaches. 3) Mitigating the impact of non-IID data distribution among devices on learning performance. Experimental results show that compared to conventional model aggregation-based FL algorithms, the proposed KFL framework is able to reduce 99% communication load while boosting 2.1% and 6.65% accuracy on MNIST and CIFAR-10 datasets, respectively. In addition, we have theoretically and experimentally revealed that more scheduled data samples should be biased to the early rounds if the scheduled data samples of the entire learning process are fixed. With this insight, we have developed an efficient online device scheduling and resource allocation algorithm to improve learning performance under devices' limited energy budgets. Experimental results show that the proposed online device scheduling algorithm converges faster than the benchmark device scheduling algorithms. In the future work, we will optimize the local models' design according to the devices' computing capabilities and datasets for further improving the learning performance of KFL.

Chapter 5

Robust Federated Learning for Unreliable and Resource-limited Networks

5.1 Introduction

Most existing FL algorithms assume an error-free wireless channel and ignore the unreliable nature of wireless communications [21]. Due to devices' constrained transmit power and bandwidth, it is hard to guarantee all the scheduled devices successfully transmit their parameters to the edge server [22]. This brings a new challenge for FL to enhance the robustness of the training process and mitigate the impact of erroneous transmission. An intuitive solution [23] is to discard the devices' parameter with errors, but it further reduces the number of participating devices and exacerbates the performance loss of FL. Thus, it is essential to develop innovative approaches for FL to address the scarcity of radio resources and the unreliability of wireless transmissions.

To mitigate the adverse impact of unreliable wireless channels and limited resources on FL, this work aims to jointly design the wireless network and learning mechanism to enhance the robustness of the training process and improve the learning performance of FL. Inspired by the success of using stale model parameters to accelerate the training process in asynchronous FL [104], we propose a novel FL framework which recycles the latest historical local gradients received at the edge server to update the global model in each round. It is worth mentioning that unlike [57, 58] that utilize the past local models to replace the transmission-failure devices' model for global aggregation, this work recycles devices' gradients to update the global model and achieves a better learning performance which verified in our simulations. In addition, we investigate the effect of partial device participation and the staleness of local gradients on the convergence bound. The main contributions of this work are summarized as follows:

- To cope with limited resources and unreliable channels in wireless networks, we propose a novel FL framework, i.e., FL with gradient recycling (FL-GR), which recycles the historical gradients of unscheduled and transmission-failure devices for global model updates. This framework can achieve faster convergence speed and higher accuracy than the conventional FL that only aggregates the successfully received local models. In addition, we formulate a joint device scheduling, resource block (RB) allocation, and power control problem to minimize the global loss, in which training latency and devices' energy consumption are considered.

- For the convenience of implementation in practical wireless networks, we propose a memory-friendly FL-GR that is equivalent to FL-FR, but with low memory space requirement of the edge server. Then, we theoretically analyze how the wireless network parameters affect the convergence bound of FL-GR. Based on the convergence bound, we define a new objective function, i.e., the average staleness of local gradients, and transform the global loss minimization problem into an explicit one for device scheduling, RB allocation, and power control.

- To solve the transformed problem, we first find the devices' optimal transmit power control policy under any given RB allocation policy. Then, we transform the orig-

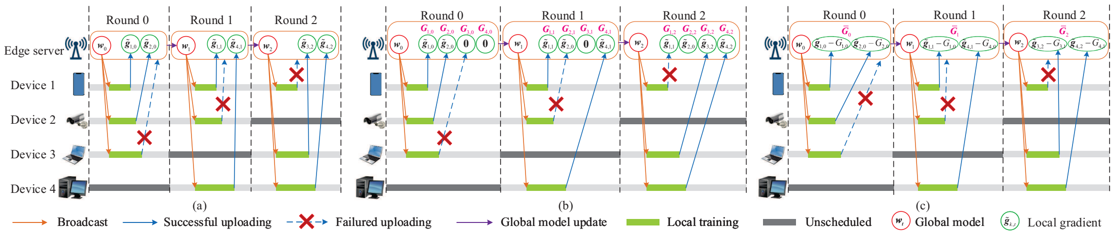


Figure 5.1: Illustration of conventional FL framework and the proposed FL-GR: (a) Conventional FL only uses the current successfully received gradients from scheduled devices to update the global model. (b) The proposed FL-GR recycles the latest historical successful transmitted gradients of unscheduled and transmission-failure devices for the global model update. (c) Memory-friendly FL-GR.

inal global loss minimization problem into a perfect bipartite matching problem. Through detailed analysis, we further transform the bipartite matching problem into equivalent linear programming whose optimal solution can be effectively solved with polynomial time complexity.

- We provide extensive experimental results on real-world datasets (i.e., MNIST, CIFAR-10, and CIFAR-100) to demonstrate the effectiveness of the proposed FL-GR and device scheduling algorithm. Compared to the FL algorithm without gradient recycling, FL-GR achieves over 4% accuracy improvement. In addition, the proposed device scheduling algorithm outperforms the benchmarks in convergence speed and test accuracy.

This rest of this chapter is organized as follows: Section 5.2 introduces the proposed FL-GR and system model, then formulate a global loss minimization problem. A memory-friendly implementation of the proposed FL-GR and the convergence analysis are illustrated in Section 5.3. Section 5.4 illustrates the proposed device scheduling, RB allocation, and power control algorithm that solves the global loss minimization problem. Section 5.5 verifies the effectiveness of FL-GR and the proposed device scheduling algorithm by simulations. The conclusion is drawn in Section 5.6.

5.2 System Model and Learning Mechanism

In this work, we investigate an FL system under a noisy and resource-limited wireless network, where the unreliable property of the wireless uplinks, i.e., transmission error, is considered. To tackle the transmission error effect on FL performance, we propose a new FL framework in which the edge server recycles the historical latest received gradients of unscheduled and transmission-failure devices to accelerate the learning process. In addition, we characterize the learning costs of the proposed FL framework and formulate an optimization problem to minimize the global loss function.

5.2.1 Federated Learning System

The considered FL system consisting of one edge server and K devices indexed by $\mathcal{K} = \{1, 2, \cdots, K\}$. Each device k ($k \in \mathcal{K}$) has a local dataset $\mathcal{D}_k$ with $D_k = |\mathcal{D}_k|$ data samples. Without loss of generality, we assume that there is no overlapping between local datasets from different devices, i.e., $\mathcal{D}_k \cap \mathcal{D}_h = \emptyset$ ($\forall k, h \in \mathcal{K}$). Thus, the entire dataset is denoted by $\mathcal{D} = \cup\{\mathcal{D}_k\}_{k=1}^{K}$ with a total number of samples $D = \sum_{k=1}^{K} D_k$. Given a data sample $(\boldsymbol{x}, y) \in \mathcal{D}$, where $\boldsymbol{x} \in \mathbb{R}^d$ is the d-dimensional input data vector, $y \in \mathbb{R}$ is the corresponding ground-truth label. Let $f(\boldsymbol{x}, y; \boldsymbol{w})$ denote the sample-wise loss function, which captures the error of the model parameter $\boldsymbol{w}$ on the input-output data pair $(\boldsymbol{x}, y)$. Thus, the local loss function of device k that measures the model error on its local dataset is given by

$$F_k(\boldsymbol{w}) = \frac{1}{D_k} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_k} f\left(\boldsymbol{x}, y; \boldsymbol{w}\right). \tag{5.1}$$

Accordingly, the global loss function associated with all distributed local datasets is given by

$$F(\boldsymbol{w}) = \sum_{k=1}^{K} p_k F_k(\boldsymbol{w}), \tag{5.2}$$

where p_k is the weight of device k such that $p_k \geq 0$ and $\sum_{k=1}^{K} p_k = 1$. Similar to many existing works, e.g., [25] and [34], we consider a balance size for local datasets and set

$p_k = \frac{1}{K}, \forall k \in \mathcal{K}$.

The objective of the FL system is to train a global model, $\boldsymbol{w}$, so as to minimize the global loss, $F(\boldsymbol{w})$, on the whole dataset, $\mathcal{D}$. The optimization objective of FL can be expressed as $\min\limits_{\boldsymbol{w}} F(\boldsymbol{w})$. To preserve the data privacy of devices, the devices collaboratively train $\boldsymbol{w}$ by periodically uploading local models or gradient information to the edge server for aggregation instead of transmitting the raw training data.

5.2.2 FL with Gradient Recycling

To address the unreliable transmissions and limited resources in the FL system, we propose a new FL framework, namely FL with gradient recycling (FL-GR), in which the edge server maintains a *gradient array* $\{\boldsymbol{G}_{k,t} : \forall k \in \mathcal{K}\}$ that caches the latest successfully received gradients for all devices and uses them for the global model update. Note that, at the beginning of The FL process, the gradient array is initialized as $\boldsymbol{G}_{k,t} = \mathbf{0}$ $(\forall k \in \mathcal{K})$. The learning process consists of T global rounds and performs the following steps in each round t $(t \in \{0, 1, \cdots, T-1\}$.

- **Step 1 (Global model broadcast)**: The edge server selects a subset of devices to participate in the current round training process and then broadcasts the latest global model, $\boldsymbol{w}_t$, to the selected devices. Let $\alpha_{k,t} \in \{0, 1\}$ to denote the scheduling indicator of device k, where $\alpha_{k,t} = 1$ indicates that device k is scheduled in round t, $\alpha_{k,t} = 0$ otherwise. We use $\boldsymbol{S}_t = \{k : \alpha_{k,t} = 1, \forall k \in \mathcal{K}\}$ to represent the device scheduling decision in round t.

- **Step 2 (Local model training)**: After receiving the global model from the edge server, each scheduled device updates its local model by running λ steps SGD on its local dataset, according to

$$\boldsymbol{w}_{k,t}^{(l+1)} = \boldsymbol{w}_{k,t}^{(l)} - \eta \tilde{\nabla} F_k(\boldsymbol{w}_{k,t}^{(l)}), \forall l = 0, \cdots, \lambda - 1, \tag{5.3}$$

where $\boldsymbol{w}_{k,t}^{(l)}$ is the local model of device k in the l-th local iteration in round t with

$\boldsymbol{w}_{k,t}^{(0)} = \boldsymbol{w}_t$, and $\eta > 0$ is the learning rate. In (5.3), the stochastic gradient $\tilde{\nabla} F_k(\boldsymbol{w}_{k,t}^{(l)})$ is given by

$$\tilde{\nabla} F_k(\boldsymbol{w}_{k,t}^{(l)}) = \frac{1}{L_b} \sum_{(\boldsymbol{x},y)\in\mathcal{B}_{k,t}^{(l)}} \nabla f(\boldsymbol{x}, y; \boldsymbol{w}_{k,t}^{(l)}), \tag{5.4}$$

where $\mathcal{B}_{k,t}^{(l)}$ is a mini-batch data uniformly sampled from $\mathcal{D}_k$ with $L_b = |\mathcal{B}_{k,t}^{(l)}|$ data samples.

- **Step 3 (Local gradient uploading)**: After accomplishing local model training, each scheduled device k $(k \in \mathcal{K})$ uploads its cumulative local stochastic gradient $\tilde{\boldsymbol{g}}_{k,t}$ to the edge server. $\tilde{\boldsymbol{g}}_{k,t}$ is given by

$$\tilde{\boldsymbol{g}}_{k,t} = \sum_{l=0}^{\lambda-1} \tilde{\nabla} F_k(\boldsymbol{w}_{k,t}^{(l)}) = \frac{1}{\eta}\left(\boldsymbol{w}_t - \boldsymbol{w}_{k,t}^{(\lambda)}\right). \tag{5.5}$$

Due to the unreliable wireless channels, the local gradient may not be successfully transmitted to the edge server. Let $s_{k,t} \in \{0, 1\}$ denote the successful transmission indicator of device k in round t, where $s_{k,t} = 1$ represents the uploaded information of device k is successfully received at the edge server, $s_{k,t} = 0$ otherwise.

- **Step 4 (Global model update)**: After the edge server receives the local gradients from the scheduled devices, the edge server updates the gradient array as

$$\boldsymbol{G}_{k,t} = \begin{cases} \tilde{\boldsymbol{g}}_{k,t}, & \text{if } \ \alpha_{k,t}s_{k,t} = 1, \\ \boldsymbol{G}_{k,t-1}, & \text{otherwise}, \end{cases} \quad \forall k \in \mathcal{K}. \tag{5.6}$$

In (5.6), the edge server only refreshes the scheduled and successfully transmitted devices' gradient and maintains the latest historical successfully received gradients for unscheduled or transmission-failure devices. Then, the edge server updates the global model as

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \boldsymbol{w}_t - \eta\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,t} \\ &= \boldsymbol{w}_t - \eta\frac{1}{K}\sum_{k=1}^{K}\left(\alpha_{k,t}s_{k,t}\tilde{\boldsymbol{g}}_{k,t} + (1-\alpha_{k,t}s_{k,t})\boldsymbol{G}_{k,t-1}\right). \end{aligned} \tag{5.7}$$

Note that, in (5.7), the edge server utilizes the successfully received gradient from scheduled devices in the current round and the historical latest received gradients of unscheduled or transmission-failure devices to update the global model. This differs from the existing works in [35, 51–55] that only aggregate the scheduled and successfully transmitted devices' gradient to update the global model, i.e., $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \frac{\sum_{k=1}^{K} \alpha_{k,t} s_{k,t} \tilde{\boldsymbol{g}}_{k,t}}{\sum_{k=1}^{K} \alpha_{k,t} s_{k,t}}$.

For the proposed FL-GR, we have the following remark:

Remark 4. *The recycling of historical local gradients in FL-GR has lower model aggregation error than the approaches in [57, 58] that reusing of historical models. For ease of comparison, we define the perfect updated global model based on all devices' local models as* $\boldsymbol{w}^*_{t+1} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{w}_{k,t+1} = \boldsymbol{w}_t - \eta\frac{1}{K}\sum_{k=1}^{K} \tilde{\boldsymbol{g}}_{k,t}$*. Note that, in [57, 58], the updated global model in round* $(t+1)$ *is* $\boldsymbol{w}^{\mathrm{m}}_{t+1} = \frac{1}{K}\sum_{k=1}^{K}(\alpha_{k,t}s_{k,t}\boldsymbol{w}_{k,t+1} + (1-\alpha_{k,t}s_{k,t})\boldsymbol{w}_{k,t-\tau_{k,t}+1})$*, where* $\tau_{k,t}$ *is the interval between the current round* t *and the last round that device* k *received global model. Thus, the aggregation model error of reusing local models in [57, 58] is given by*

$$\begin{aligned}\Delta_m &= \left\|\boldsymbol{w}^*_{t+1} - \boldsymbol{w}^{\mathrm{m}}_{t+1}\right\|^2 \\ &= \left\|\frac{1}{K}\sum\nolimits_{k=1}^{K}(1-\alpha_{k,t}s_{k,t})\left(\boldsymbol{w}_{k,t+1} - \boldsymbol{w}_{k,t-\tau_{k,t}+1}\right)\right\|^2 \\ &= \left\|\frac{1}{K}\sum\nolimits_{k=1}^{K}(1-\alpha_{k,t}s_{k,t})(\boldsymbol{w}_t - \eta\tilde{\boldsymbol{g}}_{k,t} - \boldsymbol{w}_{k,t-\tau_{k,t}} + \eta\tilde{\boldsymbol{g}}_{k,t-\tau_{k,t}})\right\|^2.\end{aligned}$$

The aggregation model error of reusing gradients is given by

$$\Delta_g = \left\|\eta\frac{1}{K}\sum\nolimits_{k=1}^{K}(1-\alpha_{k,t}s_{k,t})\left(\tilde{\boldsymbol{g}}_{k,t-\tau_{k,t}} - \tilde{\boldsymbol{g}}_{k,t}\right)\right\|^2.$$

Based on the triangle inequality, we have $\Delta_m \geq \Delta_g$*, i.e., the proposed approach that recycles historical local gradients has a smaller model aggregation error than the approaches in [57, 58] that reuse past local models. Thus, the proposed FL-GR outperforms the approaches in [57, 58], which is also verified in our simulations in Section 5.5.*

In addition, it is worth mentioning that without gradient recycling, our FL-GR degrades to the FedAvg algorithm [7]. For illustrating this, we rearrange the model update rule in (5.7) as

$$\begin{aligned}\boldsymbol{w}_{t+1} &= \boldsymbol{w}_t - \eta\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{G}_{k,t} = \frac{1}{K}\sum_{k=1}^{K}(\boldsymbol{w}_t - \eta\boldsymbol{G}_{k,t}) \\ &= \frac{1}{K}\left(\sum_{k=1}^{K}\alpha_{k,t}s_{k,t}\left(\boldsymbol{w}_t - \eta\boldsymbol{G}_{k,t}\right) + \sum_{k=1}^{K}(1-\alpha_{k,t}s_{k,t})(\boldsymbol{w}_t - \eta\boldsymbol{G}_{k,t})\right) \\ &\overset{(a)}{=} \frac{1}{K}\left(\sum_{k=1}^{K}\alpha_{k,t}s_{k,t}\boldsymbol{w}_{k,t+1} + \sum_{k=1}^{K}(1-\alpha_{k,t}s_{k,t})\boldsymbol{w}_t\right),\end{aligned} \tag{5.8}$$

where (a) is due to without gradient recycling, the unsuccessful participating devices' gradients are $\mathbf{0}$. From (5.8), without gradient recycling, the global model is updated as averaging all devices' models, which includes the successful participating devices' updated models and the unsuccessful participating devices' models that are replaced with the current global model. Thus, without gradient recycling, FL-GR will degrade to FedAvg.

To better explain FL-GR, we illustrate the conventional FL framework and FL-GR in Fig. 5.1. Assume that one edge server and four devices are in the system to perform three rounds of the FL process. At the beginning of the FL, the edge server initials the global model $\boldsymbol{w}_0$ and the gradient array for all devices to $\mathbf{0}$. Take round 2 as an example, in which devices 1, 3, and 4 are scheduled to participate in the learning process, and device 1 cannot successfully transmit its gradient. The conventional FL in Fig. 5.1(a) only aggregates the successfully transmitted devices' gradients ($\tilde{\boldsymbol{g}}_{3,2}$ and $\tilde{\boldsymbol{g}}_{4,2}$) to update the global model, i.e., $\boldsymbol{w}_3 = \boldsymbol{w}_2 - \frac{1}{2}\eta(\tilde{\boldsymbol{g}}_{3,2} + \tilde{\boldsymbol{g}}_{4,2})$. However, the FL-GR in Fig. 5.1(b) utilizes the successfully received gradients ($\tilde{\boldsymbol{g}}_{3,2}$ and $\tilde{\boldsymbol{g}}_{4,2}$) and the historical received gradient of unscheduled or transmission failure devices ($\boldsymbol{G}_{1,2} = \tilde{\boldsymbol{g}}_{1,1}$ and $\boldsymbol{G}_{2,2} = \tilde{\boldsymbol{g}}_{2,0}$) to update the global model, i.e., $\boldsymbol{w}_2 = \boldsymbol{w}_1 - \frac{1}{4}\eta(\tilde{\boldsymbol{g}}_{1,1} + \tilde{\boldsymbol{g}}_{2,0} + \tilde{\boldsymbol{g}}_{3,2} + \tilde{\boldsymbol{g}}_{4,2})$. It is worth mentioning that although FL-GR recycles the historical local gradients to update the global model, it differs from the asynchronous FL [104]. The asynchronous FL broadcasts the global model to all devices at the beginning of FL, while FL-GR only broadcasts the global to the scheduled devices. In addition, the global model update rule in (5.7) also differs from the asynchronous FL [104].

5.2.3 Computation model

Let C_k denotes the number of CPU cycles required for device k ($k \in \mathcal{K}$) to process one data sample, which can be measured offline as a priori knowledge. Let f_k represents the computation capability (CPU cycles per second) of device k. Thus, the computational time of local training is given by

$$T_{k,t}^{C} = \frac{\lambda L_b C_k}{f_k}. \tag{5.9}$$

The corresponding energy consumption of device k is

$$E_{k,t}^{C} = \kappa \lambda L_b C_k (f_k)^2, \tag{5.10}$$

where κ is the energy coefficient of devices, which depends on the chip architecture.

Note that we have ignored the computation cost of global model update at the edge server and focused on resource-limited edge devices since the edge server usually has strong computation capabilities and is supplied by the grid power.

5.2.4 Communication Model

In this work, we consider the orthogonal frequency division multiple access (OFDMA) with R RBs indexed by $\mathcal{R} = \{1, 2, \cdots, R\}$ for devices to upload their local gradients. Each device can occupy one uplink RB in a communication round to upload its local gradient. Let $\boldsymbol{z}_{k,t} = (z_{k,t}^{(1)}, z_{k,t}^{(2)}, \cdots, z_{k,t}^{(R)})$ denote the RB allocation vector for device k in round t, where $z_{k,t}^{(r)} \in \{0, 1\}$, $z_{k,t}^{(r)} = 1$ indicates that the m-th resource block is allocated to device k, and $z_{k,t}^{(r)} = 0$ otherwise. For ease of representation, we use $\boldsymbol{Z}_t = (\boldsymbol{z}_{1,t}, \boldsymbol{z}_{2,t}, \cdots, \boldsymbol{z}_{K,t})$ denote the RB allocation decision for all devices in round t. Denote $p_{k,t}$ as the transmit power of device k in round t, its maximum value is $p_{k,\max}$. The channel gain from device k to the edge server is modelled as $h_{k,t} = \rho_k(t) d_k^{-v}$, where $\rho_k(t)$ is the small-scale fading gain between device k and the edge server, d_k is the distance between device k and the edge server, and v being the path loss exponent. We consider Rayleigh fading, i.e., $\rho_k(t) \sim \exp(1)$, and it is independent and identically distributed across devices and rounds. Thus, the achievable transmit rate of device k in round t is

$$r_{k,t}(\boldsymbol{z}_{k,t}, p_{k,t}) = \sum_{r=1}^{R} z_{k,t}^{(r)} B \log_2 \left(1 + \frac{p_{k,t} h_{k,t}}{I_r + BN_0}\right), \tag{5.11}$$

where B is the bandwidth of each resource block, N_0 is the noise power spectral density, I_r is the interference caused by the devices that are located in other service areas and use the same RB [35]. It is noted that each device can only occupy at most one resource block, and each resource block can be accessed by at most one device. Thus, the RB allocation policy for devices should satisfy $\sum_{r=1}^{R} z_{k,t}^{(r)} \leq 1$ and $\sum_{k=1}^{K} z_{k,t}^{(r)} \leq 1$.

Let Q denote the size of each gradient, i.e., the number of bits used to quantify the gradients. If device k is scheduled to participate in the training process of round t, its transmission time is given by

$$T_{k,t}^{\mathrm{U}} = \frac{Q}{r_{k,t}(\boldsymbol{z}_{k,t}, p_{k,t})}. \tag{5.12}$$

The corresponding energy consumption of device k for transmission is

$$E_{k,t}^{\mathrm{U}} = p_{k,t} T_{k,t}^{\mathrm{U}}. \tag{5.13}$$

5.2.5 Successful Transmission Probability

In this work, we consider characterizing the unreliability of uplink transmissions of devices by the successful transmission probability. Before studying the uplink success probability, we assume the downlink transmission is always successful, i.e., devices successfully receive the global model. It is worth mentioning that this assumption is valid since the edge server usually has more transmit power and can occupy more RBs for the global model broadcasting compared to devices.

Let γ_{th} denotes the signal to interference plus noise ratio (SINR) threshold for successful data decoding. The successful transmission indicator of device k in round t is $s_{k,t} = \sum_{r=1}^{R} z_{k,t}^{(r)} \mathbb{1}\left(\mathrm{SINR}_{k,t}^{(r)} \geq \gamma_{\mathrm{th}}\right)$, where $\mathrm{SINR}_{k,t}^{(r)} = \frac{p_{k,t} h_{k,t}}{I_r + BN_0}$ is the SINR of device k in m-th RB. The successful transmission probability of device k through m-th channel in round t is given by

$$\begin{aligned}\Pr\left(\mathrm{SINR}_{k,t}^{(r)} \geq \gamma_{\mathrm{th}}\right) &= \Pr\left(\frac{p_{k,t}h_{k,t}}{I_r + BN_0} \geq \gamma_{\mathrm{th}}\right) \\ &= \Pr\left(\rho_k(t) \geq \frac{\gamma_{\mathrm{th}}(I_r + BN_0)}{p_{k,t}d_k^{-v}}\right) \\ &= e^{-\frac{\gamma_{\mathrm{th}}(I_r+BN_0)}{p_{k,t}d_k^{-v}}}, \end{aligned} \tag{5.14}$$

where e refers to the Euler's number.

5.2.6 Problem Formulation

In this work, we aim to minimize the global loss function after T training rounds in the resource-limited and unreliable wireless network. To this end, we formulate an optimization problem to jointly optimize device scheduling, RB allocation, and power control as follows:

$$\min_{\{\boldsymbol{S}_t\boldsymbol{Z}_t,\boldsymbol{p}_t\}_{t=0}^{T-1}} \quad \mathbb{E}[F(\boldsymbol{w}_T)] \tag{5.15}$$

$$\text{s. t.} \quad E_{k,t}^{\mathrm{C}} + E_{k,t}^{\mathrm{U}} \leq E_{k,\max}, \forall k \in \mathcal{K}, \forall t, \tag{5.15a}$$

$$T_{k,t}^{\mathrm{C}} + T_{k,t}^{\mathrm{U}} \leq T_{\max}, \forall k \in \mathcal{K}, \forall t, \tag{5.15b}$$

$$z_{k,t}^{(r)} \in \{0,1\}, \forall k \in \mathcal{K}, \forall m \in \mathcal{R}, \forall t, \tag{5.15c}$$

$$\sum_{r=1}^{R} z_{k,t}^{(r)} \leq 1, \forall k \in \mathcal{K}, \forall t, \tag{5.15d}$$

$$\sum_{k=1}^{K} z_{k,t}^{(r)} \leq 1, \forall r \in \mathcal{R}, \forall t, \tag{5.15e}$$

$$0 \leq p_{k,t} \leq p_{k,\max}, \forall k \in \mathcal{K}, \forall t, \tag{5.15f}$$

$$\alpha_{k,t} \in \{0,1\}, \forall k \in \mathcal{K}, \forall t, \tag{5.15g}$$

where (5.15a) stipulates that the energy consumption for each participating device k ($k \in \mathcal{K}$) in each round cannot exceed its budget $E_{k,\max}$. $T_{\max}$ in (5.15b) is the maximum delay of one-round FL training. (5.15c), (5.15d) and (5.15e) correspond to the RB allocation restrictions, indicating that one device can occupy at most one RB for uplink transmission, and one RB can only be allocated to one device. (5.15f) is the devices' transmit power constraint. (5.15g) indicates which devices are scheduled in each round.

Solving problem (5.15) requires the explicit form of the global loss function related to the device scheduling, power control, and RB allocation policy. However, the evolution of machine learning models in the learning process is very complex. It is almost impossible to find an exact analytical expression of $\mathbb{E}[F(\boldsymbol{w}_T)]$ with respect to $\boldsymbol{S}_t$, $\boldsymbol{Z}_t$, and $\boldsymbol{p}_t$. Thus, we turn to find an upper bound of $\mathbb{E}[F(\boldsymbol{w}_T)]$ in Section 5.3.2 and minimize it for the global loss minimization.

5.3 Memory-friendly FL-GR and Convergence Analysis

In this section, to improve the implementation feasibility of the proposed FL-GR in practical wireless networks, we first propose a memory-friendly FL-GR that is equivalent to the proposed FL-GR in Section 5.2.2 but with low memory requirements of the edge server. Then, we theoretically analyze the convergence bound of FL-GR to reveal how the device scheduling, RB allocation, and power control policies affect its learning performance. Motivated by this, we define a new objective function, i.e., the average staleness of local gradients, to transform problem (5.15) into a tractable one for guiding the wireless network design.

5.3.1 Memory-friendly FL-GR

It is worth mentioning that implementing the proposed FL-GR in Section 5.2.2 requires the edge server to maintain a huge array to cache the latest gradient information for each device. Thus, the cache size requirement of the edge server in FL-GR scales with the model size and the number of devices. This may restrict the scale of the wireless FL system since the server's memory may be exhausted when the number of devices is very large. To address this issue, we propose a **memory-friendly FL-GR** in which each device k maintains a gradient array $\boldsymbol{G}_{k,t}$ to cache its previous latest gradient, and the edge server maintains a gradient array $\bar{\boldsymbol{G}}_t$ to cache local gradients' aggregation information. Then we replace step 3 and step 4 in Section 5.2.2 with the following steps:

- Replace Step 3 in Section 5.2.2 with: After all selected devices accomplish local model training, they upload the difference between their current and the previous latest cumulative gradient, i.e., $\tilde{\boldsymbol{g}}_{k,t} - \boldsymbol{G}_{k,t-1}$, to the edge server.

- Replace Step 4 in Section 5.2.2 with: The edge server updates $\bar{\boldsymbol{G}}_t$ as $\bar{\boldsymbol{G}}_t = \bar{\boldsymbol{G}}_{t-1} + \frac{1}{K}\sum_{k=1}^{K} \alpha_{k,t} s_{k,t} (\tilde{\boldsymbol{g}}_{k,t} - \boldsymbol{G}_{k,t-1})$, and all devices update their gradient array $\boldsymbol{G}_{k,t}$ according to (5.6). Then, the edge server updates the global model as $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \bar{\boldsymbol{G}}_t$.

By replacing step 3 and step 4 in Section 5.2.2 with the above two steps, the edge server distribute the memory requirement to the devices and form a memory-friendly FL-GR algorithm, as shown in Fig. 5.1(c). For this memory-friendly FL-GR algorithm, we have the following theorem.

Theorem 3. *The memory-friendly FL-GR which formed by replacing step 3 and step 4 in Section 5.2.2 with the above two steps is equivalent to the proposed FL-GR in Section 5.2.2.*

Proof. We prove Theorem 3 by Mathematical induction. Firstly, the maintained gradient array $\bar{\boldsymbol{G}}_t$ at the edge server satisfies:

$$\begin{aligned}\bar{\boldsymbol{G}}_t &= \bar{\boldsymbol{G}}_{t-1} + \frac{1}{K}\sum_{k=1}^{K} \alpha_{k,t} s_{k,t} (\tilde{\boldsymbol{g}}_{k,t} - \boldsymbol{G}_{k,t-1}) \\ &= \bar{\boldsymbol{G}}_{t-1} + \frac{1}{K}\sum_{k=1}^{K} \left(\boldsymbol{G}_{k,t} - \boldsymbol{G}_{k,t-1}\right). \end{aligned} \tag{5.16}$$

Note that at the beginning of the learning process, the devices' gradient array $\boldsymbol{G}_{k,-1}$ and the server's gradient array $\bar{\boldsymbol{G}}_{-1}$ are all initialized with $\mathbf{0}$. Thus, when $t = 0$, we have

$$\begin{aligned}\bar{\boldsymbol{G}}_0 &= \bar{\boldsymbol{G}}_{-1} + \frac{1}{K}\sum_{k=1}^{K} (\boldsymbol{G}_{k,0} - \boldsymbol{G}_{k,-1}) \\ &= \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,0}. \end{aligned} \tag{5.17}$$

When $t = 1$,

$$\bar{\boldsymbol{G}}_1 = \bar{\boldsymbol{G}}_0 + \frac{1}{K}\sum_{k=1}^{K} (\boldsymbol{G}_{k,1} - \boldsymbol{G}_{k,0})$$

$$
\begin{aligned}
&= \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,1} + \bar{\boldsymbol{G}}_0 - \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,0} \\
&= \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,1}.
\end{aligned} \tag{5.18}
$$

Similarly, we can conclude that for any t, $\bar{\boldsymbol{G}}_t = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,t}$ is established. Thus, the updated global model by this memory-friendly FL-GR is $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\bar{\boldsymbol{G}}_t = \boldsymbol{w}_t - \eta\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,t}$, which is equivalent to update rule of the global model in (5.7) in Section 5.2.2. □

According to Theorem 3, one can implement the above memory-friendly FL-GR to achieve an equivalent learning process with the proposed FL-GR in Section 5.2 in practical wireless networks. It is worth mentioning that the computation costs, communication costs, and the learned global model of these two algorithms are the same. The memory-friendly FL-GR only reduces the edge server's memory requirement compared to FL-GR. For clarity, we summarize the detailed steps of this memory-friendly implementation of FL-GR in Algorithm 7. In the following, we focus on analyzing the convergence performance of FL-GR and transform problem (5.15) into an tractable one for device scheduling, RB allocation, and power control.

5.3.2 Convergence Analysis

For the simplicity of notation, we define the local full gradient on device k in the l-th local iteration of the t-th round as $\nabla F_k(\boldsymbol{w}_{k,t}^{(l)}) = \frac{1}{D_k}\sum_{\boldsymbol{x}\in\mathcal{D}_k} \nabla f(\boldsymbol{x}, y; \boldsymbol{w}_{k,t}^{(l)})$. Let $F(\boldsymbol{w}^*)$ denote the loss function of the optimal global model $\boldsymbol{w}^*$, and $\tilde{\eta} = \eta\lambda$ as an auxiliary variable. In addition, it is worth mentioning that we recycle the latest historical gradients of the unscheduled and transmission-failure devices to update the global model. To identify the time information of devices' gradients, we define the staleness of device k's local gradient as $\tau_{k,t}$, which evolves as

$$
\tau_{k,t} = \begin{cases} \tau_{k,t-1} + 1, & \text{if} \ \ \alpha_{k,t} s_{k,t} = 0, \\ 0, & \text{if} \ \ \alpha_{k,t} s_{k,t} = 1, \end{cases} \quad \forall k \in \mathcal{K}. \tag{5.19}
$$

Algorithm 7 Memory-friendly Implementation of FL-GR

```
Initialization: The edge server initials its gradient array $\bar{\boldsymbol{G}}_{-1} = \boldsymbol{0}$ and the global model $\boldsymbol{w}_0$, each device $k$ ($k \in \mathcal{K}$) initial their gradient array as $\boldsymbol{G}_{k,-1} = \boldsymbol{0}$
Server side:
for $t = 0, 1, \cdots, T-1$ do
    Select a subset of devices and broadcast the latest global model $\boldsymbol{w}_t$ to them.
    if Receive the gradient information from the selected devices then
        Update the gradient array $\bar{\boldsymbol{G}}_t$ as $\bar{\boldsymbol{G}}_t = \bar{\boldsymbol{G}}_{t-1} + \frac{1}{K}\sum_{k=1}^{K} \alpha_{k,t} s_{k,t} (\tilde{\boldsymbol{g}}_{k,t} - \boldsymbol{G}_{k,t-1})$
        Update the global model according to $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \bar{\boldsymbol{G}}_t$.
    else
        $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t$
    end if
end for
Device side:
if Device $k$ is scheduled then
    Receive the global model $\boldsymbol{w}_t$ from the edge server and initial $\boldsymbol{w}_{k,t}^{(0)} = \boldsymbol{w}_t$;
    for $l = 0, 1, \cdots, \lambda - 1$ do
        Update the local model according (5.3)
    end for
    Compute the cumulative stochastic gradient $\tilde{\boldsymbol{g}}_{k,t} = \frac{1}{\eta}\left(\boldsymbol{w}_t - \boldsymbol{w}_{k,t}^{(\lambda)}\right)$
    Upload the $\tilde{\boldsymbol{g}}_{k,t} - \boldsymbol{G}_{k,t-1}$ to the edge server.
    Update the gradient array $\boldsymbol{G}_{k,t}$ according to (5.6).
end if
```

Before starting the convergence analysis of FL-GR, we make the following standard assumptions for the local loss functions, i.e., $F_1(\boldsymbol{w}), F_2(\boldsymbol{w}), \cdots, F_K(\boldsymbol{w})$.

Assumption 7. *All the local loss functions, $F_k(\boldsymbol{w})$ ($\forall k \in \mathcal{K}$), are L-smooth. That is, for all $\boldsymbol{v}$ and $\boldsymbol{w}$,*

$$F_k(\boldsymbol{v}) \leq F_k(\boldsymbol{w}) + \langle F_k(\boldsymbol{w}), \boldsymbol{v} - \boldsymbol{w} \rangle + \frac{L}{2}\left\|\boldsymbol{v} - \boldsymbol{w}\right\|^2. \tag{5.20}$$

Assumption 8. *The stochastic gradient $\tilde{\nabla} F_k(\boldsymbol{w}_t)$ ($\forall k \in \mathcal{K}$) is an unbiased estimator of the full gradient $\nabla F_k(\boldsymbol{w}_t)$, i.e., $\mathbb{E}[\tilde{\nabla} F_k(\boldsymbol{w}_t)] = \nabla F_k(\boldsymbol{w}_t)$, and its variance is upper bounded by a constant σ^2, i.e., $\mathbb{E}\|\tilde{\nabla} F_k(\boldsymbol{w}_t) - \nabla F_k(\boldsymbol{w}_t)\|^2 \leq \sigma^2$.*

Assumption 9. *The expected squared norm of devices' gradients is uniformly bounded by G^2, i.e., $\left\|\nabla F_k(\boldsymbol{w}_t)\right\|^2 \leq G^2$, for all $k = 1, 2, \cdots, K$ and $t = 0, 1, \cdots, T-1$.*

Assumption 7, 8, and 9 are standard and widely used in the FL literature for convergence analysis, e.g., [27, 31, 112]. These assumptions are satisfied by the loss functions of widely used learning models, e.g., SVM, Logistic regression, and most neural networks [89]. Particularly, a deep neural network defined by a composition of functions is a Lipschitz neural network if the functions in all layers are Lipschitz [87]. It has been proved in [87] and [88] that the convolution layer, linear layer, some nonlinear activation functions (e.g., Sigmoid, tanh, Leaky ReLU, and SoftPlus), and the widely used cross-entropy function have Lipschitz smooth gradients. That is, the loss functions of most neural networks that are consisted of Lipschitz layers are Lipschitz continuous.

Before illustrating the details of convergence bound, we introduce two lemmas based on the above assumptions to assist our convergence analysis.

Lemma 9. *Let Assumption 7, 8, and 9 hold, the learning rate satisfy $\eta \leq \frac{1}{2\lambda L}$, the drift of the local model from the global model after l iterations is bounded as*

$$\mathbb{E}\left\|\boldsymbol{w}_{k,t}^{(l)} - \boldsymbol{w}_t\right\|^2 \leq \frac{4(\lambda-1)\tilde{\eta}^2}{\lambda}\left(2G^2 + \frac{\sigma^2}{\lambda}\right). \tag{5.21}$$

Proof. See Appendix C.1. □

Lemma 10. *Let Assumption 7, 8, and 9 hold, the learning rate satisfy $\eta \leq \frac{1}{2\lambda L}$, the difference between the global models in two different rounds, i.e., t and t' $(t \geq t')$, is bounded as*

$$\mathbb{E}\left\|\boldsymbol{w}_t - \boldsymbol{w}_{t'}\right\|^2 \leq 3\tilde{\eta}^2(t-t')^2\left(\left(1+\frac{8\tilde{\eta}^2L^2(\lambda-1)}{\lambda}\right)G^2 + \left(1+\frac{4\tilde{\eta}^2L^2(\lambda-1)}{\lambda^2}\right)\sigma^2\right). \tag{5.22}$$

Proof. See Appendix C.2. □

Based on Lemma 9 and Lemma 10, we derive the one-round convergence bound of the proposed FL-GR in Theorem 4 as follows:

Theorem 4. *Let Assumption 7, 8, and 9 hold, the learning rate satisfy $\eta \leq \frac{1}{2\lambda L}$, the one-round convergence bound is given by*

$$\begin{aligned}&\mathbb{E}\left[F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}_t)\right]\\&\leq \left(-\frac{1}{2}\tilde{\eta} + 3L\tilde{\eta}\right)\left\|\nabla F(\boldsymbol{w}_t)\right\|^2 + \frac{(\tilde{\eta}+3\tilde{\eta}L)(\lambda-1)}{\lambda}\left(2G^2+\frac{\sigma^2}{\lambda}\right) + \frac{(\tilde{\eta}+1)\sigma^2}{2}\\&+ c\frac{1}{K}\sum_{k=1}^{K}(\tau_{k,t-1}+1)^2\left(1-\alpha_{k,t}\sum_{r=1}^{R} z_{k,t}^{(r)}\Pr(\mathrm{SINR}_{k,t}^{(r)} \geq \gamma_{\mathrm{th}})\right),\end{aligned} \tag{5.23}$$

where $c = \frac{9}{8}(\tilde{\eta}+1)\left((1+\frac{2(\lambda-1)}{\lambda})G^2 + (1+\frac{(\lambda-1)}{\lambda^2})\sigma^2\right)$.

Proof. See Appendix C.3. □

According to Theorem 4, the summation of the square of local gradients' staleness, i.e., the last term on the RHS of (5.23), is a critical factor that negatively affects the learning convergence rate. By increasing the number of scheduled devices and their successful transmission probabilities, the expected staleness of local gradients would be reduced and thus accelerate the learning process. Due to the limited wireless resources, one should carefully design the device scheduling, RB allocation, and power control to improve the number of devices with successful transmission while satisfying their energy and delay constraints.

Based on Theorem 4, the convergence performance of FL-GR after T training rounds is given by the following corollary.

Corollary 2. *Let the assumptions in Theorem 4 hold, the expected gap between the global loss after T training rounds and the optimal loss is bounded by*

$$\mathbb{E}\left[F(\boldsymbol{w}_T)-F(\boldsymbol{w}^*)\right] \leq \underbrace{(1-\tilde{\eta}L+6\tilde{\eta}L^2)^T\mathbb{E}\left[F(\boldsymbol{w}_0)-F(\boldsymbol{w}^*)\right]}_{\text{Initial gap}} + \underbrace{c_1\frac{1-(1-\tilde{\eta}L+6\tilde{\eta}L^2)^T}{\tilde{\eta}L-6\tilde{\eta}L^2}}_{\text{Constant term}}$$
$$\underbrace{+c\sum_{t=1}^{T-1}(1-\tilde{\eta}L+6\tilde{\eta}L^2)^{T-1-t}\frac{1}{K}\sum_{k=1}^{K}(\tau_{k,t-1}+1)^2\left(1-\alpha_{k,t}\sum_{r=1}^{R}z_{k,t}^{(r)}\Pr(\mathrm{SINR}_{k,t}^{(r)}\geq\gamma_{\mathrm{th}})\right)}_{\text{Cumulative staleness of local gradients}}, \tag{5.24}$$

where $c_1=\frac{(\tilde{\eta}+3\tilde{\eta}L)(\lambda-1)}{\lambda}(2G^2+\frac{\sigma^2}{\lambda})+\frac{(\tilde{\eta}+1)\sigma^2}{2}$.

Proof. See Appendix C.4. □

From Corollary 2, the expected gap between the global loss after T rounds and the optimal loss is bounded by three terms: 1) the initial gap between the global loss and the optimal loss. 2) a constant term related to the system hyperparameters caused by multiple local iterations ($\lambda > 1$) and stochastic gradient error. 3) the cumulative staleness of local gradients over T training rounds. The first two terms determined by the system hyperparameters and the initial global model are unrelated to device scheduling, RB allocation, and power control policies. The last term is highly related to the wireless network design, which indicates that an out-of-date local gradient may degrade the learning performance. To minimize the global loss function and improve the learning performance, one should carefully design the device scheduling, RB allocation, and power control policy to minimize the average staleness of local gradients (last term on the RHS of (5.24)) for preventing the over stale local gradients. For the global loss optimization, we have the following remark:

Remark 5. *It is worth mentioning that similar to many existing works, e.g., [36, 92], the available devices and wireless resources in problem* (5.15) *are independent across different rounds. Thus, the convergence bound in (24) can be minimized by directly minimizing the average staleness of local gradients in each round, i.e.,* $\frac{1}{K}\sum\limits_{k=1}^{K}(\tau_{k,t-1}+1)^2\bigg(1-\alpha_{k,t}\sum\limits_{r=1}^{R}z_{k,t}^{(r)}\Pr(\mathrm{SINR}_{k,t}^{(r)}\geq\gamma_{\mathrm{th}})\bigg)$*. Inspired by this, we define a new objective function based on Theorem 2 and Corollary 1, i.e.,* $\frac{1}{K}\sum\limits_{k=1}^{K}(\tau_{k,t-1}+1)^2\bigg(1-\alpha_{k,t}\sum\limits_{r=1}^{R}z_{k,t}^{(r)}\Pr(\mathrm{SINR}_{k,t}^{(r)}\geq\gamma_{\mathrm{th}})\bigg)$*, which directly minimizes the upper bound on* $\mathbb{E}[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)]$ *in each round and achieves the minimization of the* T*-rounds convergence bound in* (5.24)*.*

5.4 Optimal Device Scheduling, Resource Allocation, and Power Control

In this section, we propose an effective device scheduling, RB allocation, and power control algorithm that solves problem (5.15). Towards this end, we first transform problem (5.15) into a tractable one based on the convergence analysis in Section 5.3.2. Then, we solve the optimal power control and RB allocation policies in an effective manner.

5.4.1 Problem Transformation

The convergence analysis results in Theorem 4 and Corollary 2 reveal how the wireless network design affects the learning performance of FL-GR. According to Remark 5, we transform problem (5.15) into minimize $\frac{1}{K}\sum\limits_{k=1}^{K}(\tau_{k,t-1}+1)^2\bigg(1-\alpha_{k,t}\sum\limits_{r=1}^{R}z_{k,t}^{(r)}\Pr(\mathrm{SINR}_{k,t}^{(r)}\geq$

$\gamma_{\text{th}})\Big)$ in each round through device scheduling, RB allocation, and power control policies. Since $\alpha_{k,t} = \sum_{r=1}^{R} z_{k,t}^{(r)} \in \{0,1\}$, we have $\alpha_{k,t} \sum_{r=1}^{R} z_{k,t}^{(r)} \Pr(\text{SINR}_{k,t}^{(r)} \geq \gamma_{\text{th}}) = \sum_{r=1}^{R} z_{k,t}^{(r)} \Pr(\text{SINR}_{k,t}^{(r)} \geq \gamma_{\text{th}})$. That is, when the RB allocation decision is given, the device scheduling policy can be directly computed by $\alpha_{k,t} = \sum_{r=1}^{R} z_{k,t}^{(r)}$ $(\forall k \in \mathcal{K})$. Therefore, we transform problem (5.15) into minimizing the average square of local gradients' staleness in each round as follows:

$$\min_{\boldsymbol{Z}_t, \boldsymbol{p}_t} \quad \frac{1}{K} \sum_{k=1}^{K} (\tau_{k,t-1}+1)^2 \Bigg(1 - \sum_{r=1}^{R} z_{k,t}^{(r)} \Pr(\text{SINR}_{k,t}^{(r)} \geq \gamma_{\text{th}})\Bigg) \tag{5.25}$$

$$\text{s. t.} \quad (5.15\text{a}) - (5.15\text{f}).$$

Problem (5.25) is a non-convex optimization problem which is difficult to solve. In the following, we derive the optimal power control policy for each device under any given RB allocation decision and transform problem (5.25) into an equivalent linear programming problem that can be effectively addressed.

5.4.2 Optimal Power Control

For any given RB allocation policy $\boldsymbol{Z}_t$, it is straightforward to see that the power control policies of devices do not affect each other and independently contribute to the objective function. Therefore, the power control policy for each device can be solely optimized by itself. With given RB allocation policy $\boldsymbol{Z}_t$, we decompose the power control optimization problem for each device k $(k \in \mathcal{K})$ as follows:

$$\min_{p_{k,t}} \quad h_k(p_{k,t}) \tag{5.26}$$

$$\text{s. t.} \quad (5.15\text{a}), (5.15\text{f}).$$

where

$$h_k(p_{k,t}) = -(\tau_{k,t-1}+1)^2 \sum_{r=1}^{R} z_{k,t}^{(r)} e^{-\frac{\gamma_{\text{th}}(I_r + BN_0)}{p_{k,t} d_k^{-v}}}. \tag{5.27}$$

Problem (5.26) is a non-convex optimization problem. To solve the optimal power control policy, below we analyze the properties of the objective function and constraints of problem (5.26). Firstly, the first-order partial derivative of the objective function with respect to $p_{k,t}$ is given by

$$\frac{\partial h_k(p_{k,t})}{\partial p_{k,t}} = -(\tau_{k,t-1}+1)^2 \sum_{r=1}^{R} z_{k,t}^{(r)} e^{-\frac{\gamma_{\text{th}}(I_r+BN_0)}{p_{k,t}d_k^{-v}}} \frac{\gamma_{\text{th}}(I_r+BN_0)}{p_{k,t}^2 d_k^{-v}}, \forall k \in \mathcal{K}. \tag{5.28}$$

It is straightforward to see that $\frac{\partial h_k(p_{k,t})}{\partial p_{k,t}} < 0$ since $p_{k,t} > 0$. That is, the objective function $h_k(p_{k,t})$ is a monotonically decreasing function with the transmit power $p_{k,t}$ ($\forall k \in \mathcal{K}$). Thus, the optimal transmit power for each device is its maximum available power. According to constraint (5.15a), the energy consumption of gradient information uploading should satisfy $E_{k,t}^{\text{U}} \leq E_{k,\max} - E_{k,t}^{\text{C}}$. In addition, the first-order partial derivative of $E_{k,t}^{\text{U}}$ with respect to $p_{k,t}$ satisfies

$$\frac{\partial E_{k,t}^{\text{U}}}{\partial p_{k,t}} = \frac{Q\sum\limits_{r=1}^{R} z_{k,t}^{(r)} B \frac{1}{\left(1+\frac{p_{k,t}h_{k,t}}{I_r+BN_0}\right)\ln 2}\left(\left(1+\frac{p_{k,t}h_{k,t}}{I_r+BN_0}\right)\ln\left(1+\frac{p_{k,t}h_{k,t}}{I_r+BN_0}\right)-\frac{p_{k,t}h_{k,t}}{I_r+BN_0}\right)}{\left(\sum\limits_{r=1}^{R} z_{k,t}^{(r)} B\log_2\left(1+\frac{p_{k,t}h_{k,t}}{I_r+BN_0}\right)\right)^2} > 0, \tag{5.29}$$

where the inequality is because $\ln(1+x) > \frac{x}{1+x}$, for $x > 0$. Therefore, $E_{k,t}^{\text{U}}$ is monotonically increases with $p_{k,t}$. Hence, the transmit power of device k should satisfies $p_{k,t} \leq p_{k,t}^{\text{E}}$, where $p_{k,t}^{\text{E}}$ satisfy $\frac{p_{k,t}^{\text{E}}Q}{r_{k,t}(\boldsymbol{z}_{k,t},p_{k,t}^{\text{E}})} = E_{k,\max} - \kappa\lambda L_b C_k f_k^2$. Combining with constraint (5.15f), the optimal power control policy for device k is

$$p_{k,t}^{*} = \min\{p_{k,t}^{\text{E}}, p_{k,\max}\}, \forall k \in \mathcal{K}, \tag{5.30}$$

where $p_{k,t}^{\text{E}}$ satisfy $\frac{p_{k,t}^{\text{E}}Q}{r_{k,t}(\boldsymbol{z}_{k,t},p_{k,t}^{\text{E}})} = E_{k,\max} - \kappa\lambda L_b C_k f_k^2$.

5.4.3 Optimal Resource Block Allocation

Up to now, we can compute the optimal power control policy for each device k ($k \in \mathcal{K}$) with any allocated RB m ($m \in \mathcal{R}$) based on (5.30), denoted by $p_{k,t}^{*}(r)$. Thus, we compute the optimal power control policy for all devices in all RBs (i.e., $\{p_{k,t}^{*}(r) : \forall m \in \mathcal{R}, \forall k \in \mathcal{K}\}$) and substitute them into problem $\widehat{\mathcal{P}}$ to simplify it as the following RB allocation problem.

$$\max_{\boldsymbol{Z}_t} \quad \frac{1}{K}\sum_{k=1}^{K}\sum_{r=1}^{R} z_{k,t}^{(r)}(\tau_{k,t-1}+1)^2 e^{-\frac{\gamma_{\text{th}}(I_r+BN_0)}{p_{k,t}^{*}(r)d_k^{-v}}} \tag{5.31}$$

$$\text{s. t.} \quad (5.15c), (5.15d), (5.15e),$$

$$\frac{\lambda L_b C_k}{f_k} + \frac{Q}{r_{k,t}(z_{k,t}, p^*_{k,t}(r))} \leq T_{k,\max}, \forall k \in \mathcal{K}, \forall m \in \mathcal{R}. \tag{5.31a}$$

Problem (5.31) is a typical non-linear integer programming problem which is difficult to solve. Below we transform it into a maximum weight perfect bipartite matching problem and find its optimal solution within polynomial time.

To transform (5.31) into a bipartite matching problem, we construct a complete and balanced bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{K} \cup \overline{\mathcal{R}}$ is the vertex set, and $\mathcal{E}$ is the set of edges that connect the vertices in $\mathcal{K}$ and $\overline{\mathcal{R}}$. In $\mathcal{G}$, each vertex k in $\mathcal{K}$ corresponds a device k. $\overline{\mathcal{R}} = \mathcal{R} \cup \mathcal{R}_v$ is an extended set of $\mathcal{R}$, where each vertex r in $\mathcal{R}$ corresponds to RB r. $\mathcal{R}_v$ is the virtual vertex set used to construct a balanced bipartite graph $\mathcal{G}$, which makes the size of $\overline{\mathcal{R}}$ equal to the size of $\mathcal{K}$, i.e., $|\overline{\mathcal{R}}|= |\mathcal{K}|$. The weight of edges in $\mathcal{G}$ is given by:

$$\Delta_{k,r} = \begin{cases} (\tau_{k,t-1}+1)^2 e^{-\frac{\gamma_{\text{th}}(I_r+BN_0)}{p^*_{k,t}(r) d_k^{-v}}}, & \text{if } \frac{\lambda L_b C_k}{f_k} + \frac{Q}{r_{k,t}(z_{k,t}, p^*_{k,t}(r))} \leq T_{k,\max}, k \in \mathcal{K}, r \in \mathcal{R}, \\ 0, & \text{else.} \end{cases} \tag{5.32}$$

Note that this work assumes that the number of devices exceeds the number of RBs. When the number of RBs exceeds the number of devices, we can introduce a virtual device set $\mathcal{K}_v$ such that the $|\mathcal{K}|+|\mathcal{K}_v|= |\mathcal{R}|$, and construct a similar graph to the case of $|\mathcal{K}|> |\mathcal{R}|$.

According to the above-defined bipartite graph $\mathcal{G}$, we transform (5.31) into a maximum weight perfect bipartite matching problem, which aims to find a perfect matching $\mathcal{H}$ of $\mathcal{G}$ maximizing $\sum_{e\in\mathcal{H}} \Delta_{k,r}$. Let $\theta_{k,r} \in \{0, 1\}$ be the edge connecting vertex k ($k \in \mathcal{K}$) and vertex r ($r \in \overline{\mathcal{R}}$), where $\theta_{k,r} = 1$ denote that RB r is allocated to device k, and $\theta_{k,r} = 0$ otherwise. For the sake of presentation, we use $\boldsymbol{\theta}_k = \{\theta_{k,1}, \theta_{k,2}, \cdots, \theta_{k,|\overline{\mathcal{R}}|}\}$ to denote the connection indicator of device k to all the RBs. Hence, we formulate the bipartite matching problem as the following optimization problem.

$$\max_{\{\boldsymbol{\theta}_k\}_{k=1}^K} \quad \sum_{k=1}^{K}\sum_{r=1}^{|\overline{\mathcal{R}}|}\theta_{k,r}\Delta_{k,r} \tag{5.33}$$

$$\text{s. t.} \quad \sum_{r=1}^{|\overline{\mathcal{R}}|}\theta_{k,r}=1, \forall k\in\mathcal{K}, \tag{5.33a}$$

$$\sum_{k=1}^{K}\theta_{k,r}=1, \forall r\in\overline{\mathcal{R}}, \tag{5.33b}$$

$$\theta_{k,r}\in\{0,1\}\,, \forall k\in\mathcal{K}, \forall r\in\overline{\mathcal{R}}. \tag{5.33c}$$

It is worth mentioning that any solution to problem (5.33) corresponds to a perfect matching of graph $\mathcal{G}$. However, problem (5.33) is a linear integer programming, which is still difficult to solve. By relaxing the integrality constraint (5.33c), we can obtain the following linear programming problem:

$$\max_{\{\boldsymbol{\theta}_k\}_{k=1}^K} \quad \sum_{k=1}^{K}\sum_{r=1}^{|\overline{\mathcal{R}}|}\theta_{k,r}\Delta_{k,r} \tag{5.34}$$

$$\text{s. t.} \quad (5.33a), (5.33b),$$

$$0\le\theta_{k,r}\le 1, \forall k\in\mathcal{K}, \forall r\in\overline{\mathcal{R}}. \tag{5.34a}$$

Problem (5.34) is the linear programming relaxation of problem (5.33), which can be solved by using the current matrix multiplication time algorithm [113] with time complexity of $\mathcal{O}((K^{2+1/6})^2)$ since it has K^2 variables (i.e., $\theta_{k,r}: k\in\mathcal{K}, r\in\overline{\mathcal{R}}$). Note that in problem (5.34), each row in the coefficient matrix corresponding to (5.33a) and constraint (5.33b) only contains a '1'. This implements that each square submatrix of this coefficient matrix has a determinant equal to 0, +1, or -1. Thus, this coefficient matrix is a totally unimodular matrix. Based on [114], the optimal solution of problem (5.34) is an integer solution which is equal to the optimal solution of problem (5.33). That is, the optimal solution of (5.33) can be obtained by directly solving problem (5.34). In the above analysis, we first transform problem (5.25) into an equivalent maximum weight perfect bipartite matching problem, i.e., problem (5.33). Then, we further transform problem (5.33) into its equivalent linear programming (5.34). It is worth mentioning that these are two equivalent transformations and do not change the optimality of problem (5.25). Thus, the optimal solution of problem (5.25) can be addressed by first solving the optimal solution of problem (5.34). When the optimal solution of problem (5.34) is found, the optimal RB allocation is determined. Furthermore, the optimal device scheduling policy can be computed by $\alpha^*_{k,t} = \sum_{r=1}^{R} z_{k,t}^{(r,*)}$ $(\forall k \in \mathcal{K})$, and the optimal transmit power of each device can be determined by (5.30).

According to the above analysis, we can solve problem (5.25) in an effective manner to obtain the optimal device scheduling, power control, and RB allocation policies. For clarity, we summarize the detailed steps of solving problem (5.25) in Algorithm 8. Firstly, Algorithm 8 requires computing all devices' optimal power control policies in all RBs according to (5.30), which requires computing $K \times R$ times of power control policy and has a time complexity of $\mathcal{O}(KR)$. Then, we construct a bipartite graph to transform problem (5.25) into a maximum weight perfect bipartite matching problem, i.e., (5.33). This step requires calculating the successful transmission probabilities for all devices in all RBs and judging whether the devices' delay satisfies the latency constraint. The time complexity of this step is $\mathcal{O}(2KR)$. Finally, we transform problem (5.33) into equivalent linear programming (i.e., (5.34)) and utilize the current matrix multiplication time algorithm [113] to solve its optimal solution for obtaining the RB allocation policy. After that, we find the optimal power control of scheduling devices based on the RB allocation policy and compute the device scheduling policies as $\alpha^*_{k,t} = \sum_{r=1}^{R} z_{k,t}^{(r,*)}$ $(\forall k \in \mathcal{K})$. Thus, the overall time complexity of Algorithm 8 is $\mathcal{O}(3KR + (K^{2+1/6})^2)$.

Algorithm 8 requires computing the optimal power control policy and successful transmission probabilities for all devices in all RBs, which has a time complexity of $\mathcal{O}(2KR)$. Then, we construct a bipartite graph and solve the corresponding linear programming, and the time complexity is $\mathcal{O}((K^{2+1/6})^2)$. Thus, the overall time complexity of Algorithm 8 is $\mathcal{O}(2KR + (K^{2+1/6})^2)$.

Algorithm 8 Optimal Device Scheduling, Power control, and RB allocation

```
Compute the optimal power control policy for each device in all RBs according to (5.30)
Compute the successful transmission probabilities for all devices with all RB, i.e., $\Pr(\mathrm{SINR}_{k,t}^{(r)} \geq \gamma_{\mathrm{th}})$ $(\forall k \in \mathcal{K}, \forall r \in \mathcal{R})$
Construct a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and compute the weight of each edge in $\mathcal{E}$ according to (5.32)
Construct the linear programming problem (5.34)
Solve problem (5.34) and obtain the optimal bipartite perfect matching $\{\boldsymbol{\theta}_k\}_{k=1}^{K}$
Compute the optimal RB allocation policy $\boldsymbol{Z}_t^* = \{z_{k,t}^{(r,*)} : k \in \mathcal{K}, r \in \mathcal{R}\}$, where $z_{k,t}^{(r,*)} = \theta_{k,r}$
Compute the optimal device scheduling policy $\boldsymbol{S}_t^* = \{\alpha_{k,t}^* = 1 : \forall k \in \mathcal{K}\}$, where $\alpha_{k,t}^* = \sum_{r=1}^{R} z_{k,t}^{(r,*)}$ $(\forall k \in \mathcal{K})$
Return the optimal device scheduling policy $\boldsymbol{S}_t^*$, RB allocation policy $\boldsymbol{Z}_t^*$, and power control policy $\boldsymbol{p}_t^*$
```

Table 5-A: Network Architecture for the Classification Model

Dataset	Model Name	Model Architecture
MNIST	MLP	F: [784, 128, 10]
CIFAR-10	CNN	C: [6, M, 16, M]
		F: [1600, 256, 64, 10]
CIFAR-100	VGG-11	C: VGG-11 feature extractor [69]
		F: [512, 256, 100]

5.5 Numerical Results

In this section, we evaluate the performance of our proposed FL-GR and the device scheduling algorithm. All the codes in the simulation are implemented in python 3.8 and Pytorch, running on a Linux server. We first present the evaluation setup and then show experimental results.

5.5.1 Simulation Settings

For the simulations, we consider a cellular network with a coverage radius of 500m, in which one base station is located at its centre and K devices are randomly distributed. The CPU frequency of each device is randomly selected from $\{0.8, 1.0, 1.2, 1.4\}$ GHz. We evaluate the proposed algorithm under three classification learning tasks, i.e., the handwritten digits classification task on the MNIST dataset, as well as the image classification tasks on the CIFAR-10 and CIFAR-100 datasets. The network architectures used for the learning tasks on these three datasets are summarized in Table 5-A, where

Table 5-B: Simulation Parameter Settings

Parameter	Value	Parameter	Value
K	100	R	10
B	1MHz	N_0	-174dBm
v	2	γ_{th}	0dB
κ	5×10^{-27}	η	0.05
τ	5	L_b	64
$p_{k,\max}(\forall k \in \mathcal{K})$	30mW	$I_r(\forall m \in \mathcal{R})$	$\left[10^2 BN_0, 10^5 BN_0\right]$
Q(CNN)	1,962,016	C_k(CNN)	326,338,5
Q(VGG-11)	305,685,860	C_k(VGG-11)	76,421,465
$T_{\max}$(CNN)	0.3s	$T_{\max}$(VGG-11)	35s
$E_{k,\max}$(CNN)	1.0J	$E_{k,\max}$(VGG-11)	230J

'F' denotes the fully connected module, 'C' denotes the convolution module, 'M' denotes the 2 × 2 max-pooling layer, and the number indicates the number of neurons in fully connected layers or filters in convolution layers. Particularly, for the CNN used on the CIFAR-10 dataset, the size of convolution kernels are all set to be 5 × 5. The input and hidden layers in all the learning models are all activated by the ReLU function. For all three datasets, we adopt a typical heterogeneous data-splitting method that is widely used in the existing FL works, e.g., [36, 92]. We first classify the training data samples according to their labels, then split the data samples in each class into $2K/N$ shards ($N = 10$ on MNIST and CIFAR-10; $N = 100$ on CIFAR-100), and finally randomly distribute two shards of data samples to each device. That is, each device has a data distribution corresponding to at most 2 classes. That is, each device has a data distribution corresponding to at most 2 classes. For all models, a momentum of 0.9 is adopted and cross-entropy is adopted as the loss function. In addition, each parameter of these models is quantitated as 16 bits [34]. For all devices in the system, each CPU cycle can process 4 FLOPs. Thus, the required CPU cycles to process one data sample are equal to the number of FLOPs of its model divided by 4. The parameters chosen in the simulations are based on the parameter settings of a typical wireless FL system [27, 33, 35, 98, 115]. If not specified, the default system settings are listed in Table 5-B.

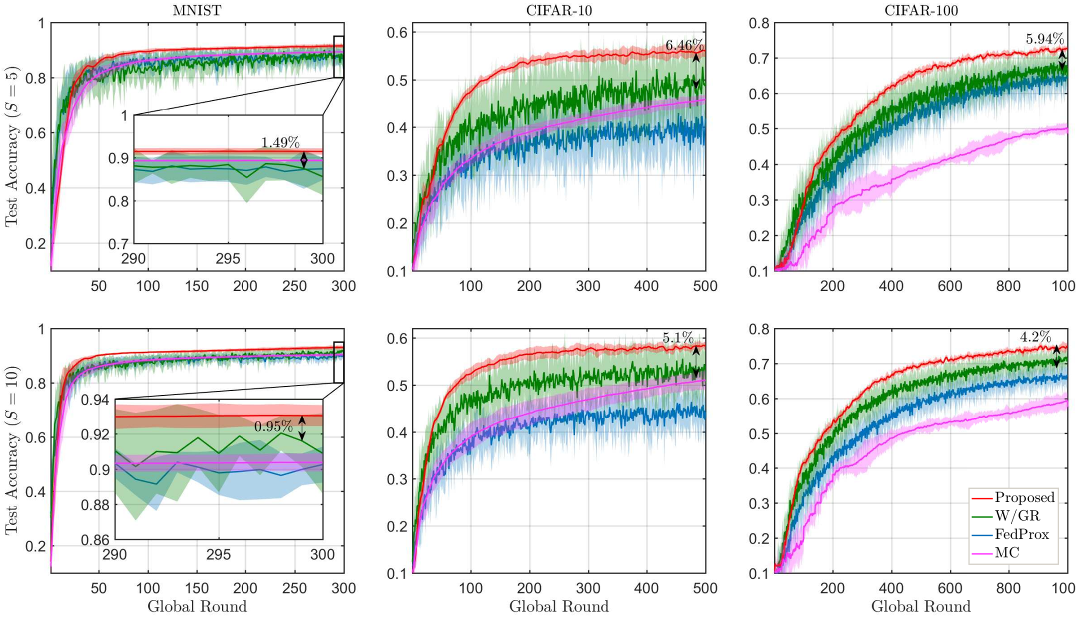
Table 5-C: Required Rounds to Reach a Target Accuracy

Dataset	S	Target Accuracy	Proposed	Best Baseline	Saved Time
MNIST	5	85%	48	80 (MC)	40%
	10	90%	50	233 (MC)	78.5%
CIFAR-10	5	50%	128	318 (W/GR)	59.7%
	10	55%	147	376 (W/GR)	60.9%
CIFAR-100	5	65%	457	640 (W/GR)	28.6%
	10	70%	509	760 (W/GR)	33%

5.5.2 Effectiveness of Gradient Recycling

To evaluate the effectiveness of the proposed FL-GR, we compare it with the following benchmarks in terms of test accuracy under different numbers of successful transmission devices (denoted as S) per round: 1) Without gradient recycling (W/GR): In each round, the edge server only aggregates the successfully received gradients from the scheduled devices to update the global model. This scheme is widely used in existing literature, e.g., [35, 52–55]. 2) FedProx [116]: FedProx utilizes a proximal term to limit the impact of local updates for improving model performance under heterogeneous data distributions among devices. 3) Model compensation (MC) [57, 58]: In each round, the edge server uses the successfully received local models and the past local models of transmission-failure or unscheduled devices for global model aggregation.

Fig. 5.2 presents the test accuracy of the proposed FL-GR and two benchmarks. It is observed that FL-GR outperforms the benchmarks on all three datasets. In addition, the learning performance of all three FL algorithms improved with the increasing number of successful-transmission devices, i.e., the test accuracy of $S = 10$ is greater than that of $S = 5$ for all three FL algorithms. Specifically, from the results on the MNIST dataset in Fig. 5.2(a), when $S = 5$ devices successfully transmitted their gradients to the edge server in each round, FL-GR achieved a 1.37% accuracy improvement compared to the FL algorithms without gradient recycling. Although FL-GR only achieves a slight performance gain (i.e., 0.89%) when $S = 10$, its learning process is more stable than the benchmarks. Fig. 5.2(b) and Fig. 5.2(c) evaluate the learning performance of FL-GR on CIFAR-10 and CIFAR-100, respectively, drawing a similar conclusion to the MNIST



(a)

(b)

(c)

Figure 5.2: Comparison of learning performance of different FL algorithms: (a) on MNIST dataset; (b) on CIFAR-10 dataset; (c) on CIFAR-100 dataset.

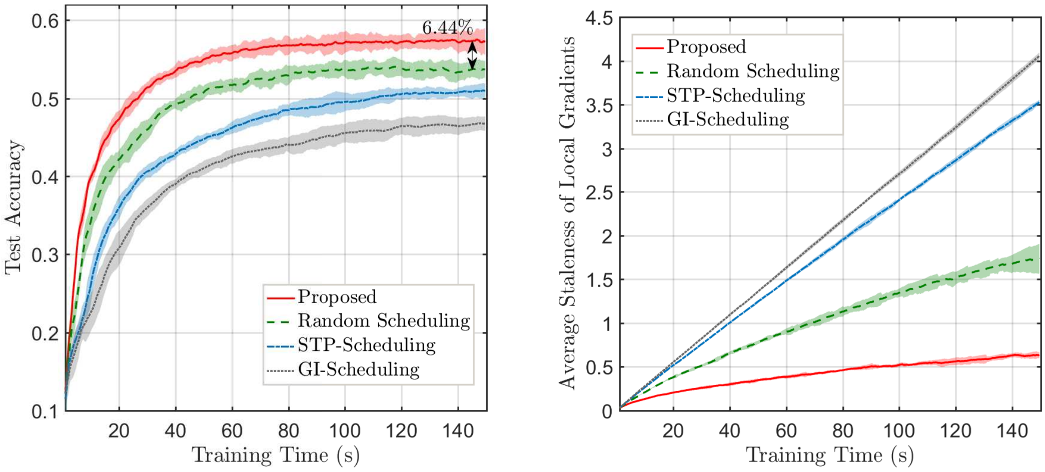
dataset. In particular, we can observe a more distinct accuracy boosting of FL-GR on these two complicated datasets than the MNIST dataset. From Fig. 5.2(b), FL-GR obtains 3.41% and 2.83% accuracy improvement when $S = 5$ and $S = 10$, respectively. Fig. 5.2(c) shows that FL-GR boosts 4.08% and 2.87% accuracy when $S = 5$ and $S = 10$, respectively. In addition, from Fig. 5.2(c), when $S = 5$, FL-GR spends only 457 rounds to achieve 65% accuracy, while wGR (the best benchmark) requires 640 rounds. That is, FL-GR can reduce by 29% training time to obtain 65% test accuracy compared to the benchmarks. When $S = 10$, FL-GR is able to save 33% training time to achieve 70% test accuracy compared to the benchmarks.

In Table 5-C, we present the number of optimization rounds necessary to achieve a target accuracy for both the proposed approach and the top-performing baseline algorithm. Specifically, on the CIFAR-100 dataset, when $S = 5$, FL-GR spends only 457 rounds to achieve 65% accuracy, while W/GR (the best benchmark) requires 640 rounds. That is, FL-GR can reduce by 28.6% training time to obtain 65% test accuracy compared to the benchmarks. When $S = 10$, FL-GR is able to save 33% training time to achieve 70% test accuracy compared to the benchmarks.

It is worth mentioning that the simulation results in Fig. 5.2 show that the proposed FL-GR outperforms the model compensation approach in [57, 58]. The latent reason is that the historical gradients of devices are able to approximate their current gradients with small approximation errors. In contrast, the past local models of devices may not match their current local models well. In fact, some existing works on FL, e.g., [92, 115], have shown that the gradients on one device have continuity. Specifically, its shown in [92] that the ℓ_2-norm of the past gradient of a device can effectively approximate the ℓ_2-norm of its current gradient. Moreover, our previous work [115] experimentally demonstrated that for any device, its past gradient is able to approximate the current gradient effectively. In addition, although the simulations in [57, 58] demonstrated that the model compensation approach outperforms the W/GR method under full participation and small transmission error rates, our simulation results on the CIFAR-10 and CIFAR-100 datasets show that it does not perform better than W/GR under small successful participation ratios (i.e., $S = 5$ and $S = 10$ correspond to 5% and 10% successful participation ratio, respectively).

5.5.3 Comparison of Device Scheduling Policies

In this subsection, we compare the proposed device scheduling algorithm to the following scheduling policies: 1) Random scheduling: In each round, the edge server randomly selects a subset of devices and their corresponding RBs that satisfy the constraint (5.15a)-(5.15f). 2) Gradient importance-aware scheduling (GI-Scheduling): The edge server selects a subset of devices with the maximum gradient norm and satisfies the constraint (5.15a)-(5.15f) in each round. 3) Successful transmission probability-aware scheduling (STP-Scheduling): The edge server selects a subset of devices with the maximum successful transmission probabilities and satisfies the constraint (5.15a)-(5.15f). Note that all the device scheduling approaches in this subsection serve the proposed FL-



(a)

(b)

Figure 5.3: Comparison of learning performance for different device scheduling algorithms on the CIFAR-10 dataset.

GR to schedule devices instead of other learning frameworks. In fact, random scheduling is equivalent to randomly selecting a perfect matching in the constructed bipartite graph of the proposed device scheduling algorithm instead of the maximum weight perfect matching to schedule devices. GI-Scheduling and STP-Scheduling both construct a similar bipartite graph to the proposed scheduling algorithm and find the maximum weight perfect matching corresponding to their device scheduling policy. In the graph of GI-Scheduling, the weight of each edge is equal to the gradient norm times the successful transmission probability. In the STP-Scheduling, the weight of each edge is the successful transmission probability.

Fig. 5.3 shows the learning performance of different device scheduling algorithms on the CIFAR-10 dataset. From Fig. 5.3(a), we can see that the proposed device scheduling algorithm performs better than the other three device scheduling approaches in terms of convergence speed and final test accuracy. Specifically, the proposed device scheduling algorithm achieves around 3.5% accuracy improvement compare to the random scheduling approach. Fig. 5.3(b) presents the average staleness of local gradients for all four device scheduling algorithms. It is observed that the proposed algorithm possesses the lowest staleness of local gradients. In addition, for the three benchmarks, the device

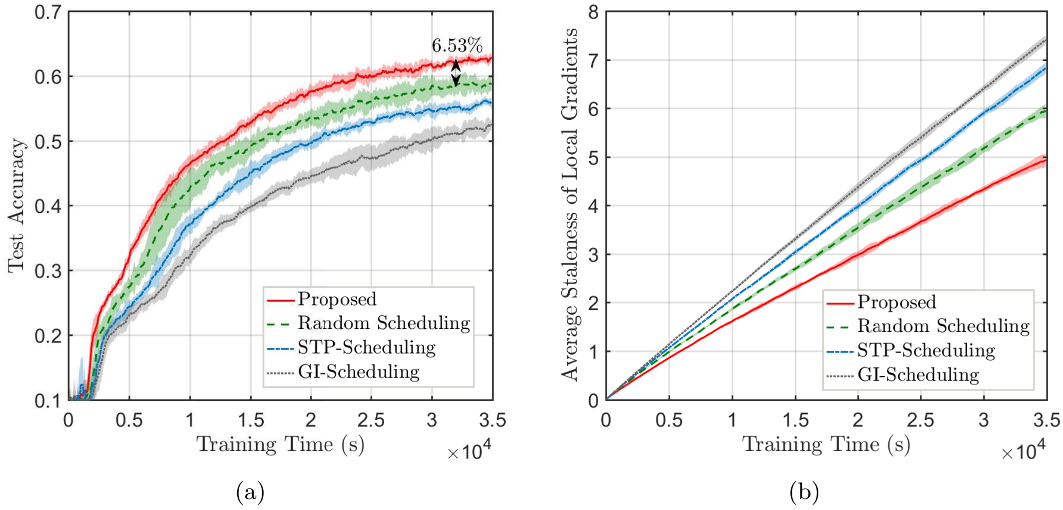
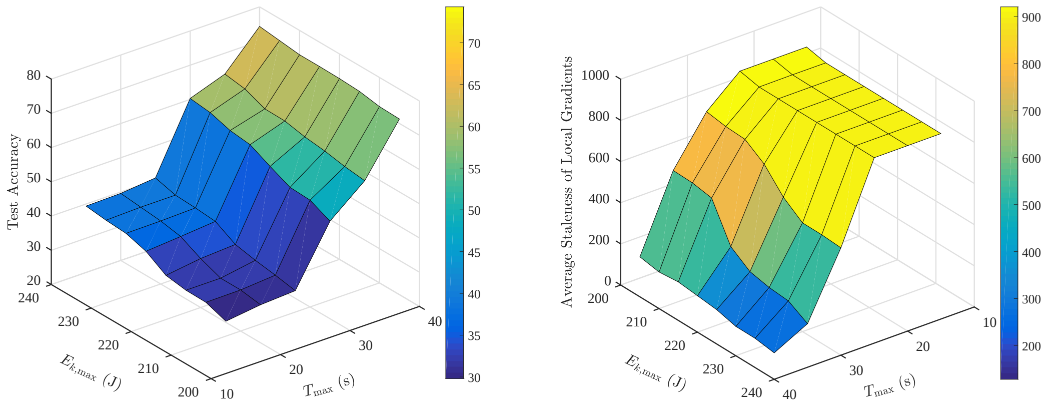


Figure 5.4: Comparison of learning performance for different device scheduling algorithms on the CIFAR-100 dataset.

scheduling algorithm with lower staleness obtains higher accuracy and faster convergence speed. This verified our theoretical analysis results in Theorem 4 and Corollary 2, which suggests scheduling the devices with large staleness to reduce the average square staleness of local gradients in each round.

A similar comparison is made on the CIFAR-100 dataset in Fig. 5.4. We can observe the same conclusion with the simulation on the CIFAR-10 dataset. Specifically, The proposed device scheduling algorithm boosts 3.85% accuracy and possesses the lowest staleness of local gradients compared to the three benchmarks. This simulation further verifies the effectiveness of our convergence analysis in Theorem 4 and Corollary 2. In addition, it is worth mentioning that the simulation results on both CIFAR-10 and CIFAR-100 datasets show that random scheduling performs better than STP-scheduling and GI-scheduling when they serve FL-GR. This is because STP-scheduling and GI-scheduling induce higher average staleness of local gradients, as shown in Fig. 5.3(b) and Fig. 5.4(b). However, when these scheduling approaches serve for FedAvg, some existing works, e.g., [31, 35], have shown that random scheduling performs worse than STP-scheduling and GI-scheduling.



(a)

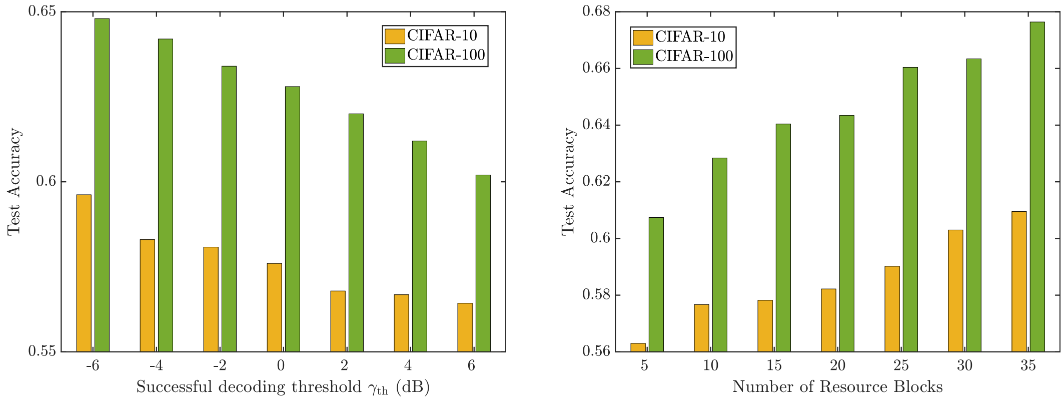
(b)

Figure 5.5: Impacts of energy and delay constraints on the learning performance on CIFAR-100 dataset.

5.5.4 Impact of Wireless Parameters

This section analyzes the impacts of wireless network parameters on the learning performance of the proposed FL-GR, including energy constraint, delay constraint, the number of RBs, and the successful decode threshold. Note that in this section, the test accuracy on CIFAR-10 and CIFAR-100 is achieved after 150 and 3.5×10^4 seconds of training, respectively. Based on our simulation results in Section 5.5.3, FL-GR can converge within the pre-setting training time.

In Fig. 5.5, we test the impacts of energy and delay constraints on the final test accuracy of the proposed FL-GR on CIFAR-100 datasets. From Fig. 5.5(a), with the increase in energy and delay budgets, we can see that FL-GR achieves higher test accuracy. The reason is that the large energy and delay budgets can increase the number of successful participating devices and reduce the average staleness of local gradients, as shown in Fig. 5.5(b). When the energy and delay budgets are small, the devices with long time delays and large energy consumption may not satisfy the delay and energy consumption constraints or cannot successfully upload their gradient information to the edge server. Thus, the number of successful participants has been restricted, which results in the high average staleness of local gradients and low test accuracy.



(a)

(b)

Figure 5.6: Impacts of wireless parameters on the learning performance: (a) Successful decoding threshold; (b) Number of RBs.

Fig. 5.6 evaluates the impacts of wireless network parameters on the learning performance of the proposed FL-GR on both CIFAR-10 and CIFAR-100 datasets. From Fig. 5.6(a), we can see the test accuracy of FL-GR on CIFAR-10 and CIFAR-100 decrease with the increase of γ_{th}. This is because the large γ_{th} reduces the successful transmission probabilities of devices, decreasing the number of successful participants. Hence, the average staleness of local gradients increased. Based on our convergence analysis results, the learning performance of FL-GR will decrease with the increase of γ_{th}. Fig. 5.6(b) shows how the number of RBs affects the learning performance of FL-GR. It is observed that the test accuracy on both CIFAR-10 and CIFAR-100 increases with the increase in the number of RBs. The reason is the rise in RBs improves the number of successful participants in each round and thus reduces the average staleness of local gradients.

5.6 Summary

In this chapter, we have developed a novel FL framework, namely FL-GR, that recycles devices' historical gradients to update the global model in the learning process. This framework efficiently copes with the scarcity of radio resources and the unreliability of wireless communications in practical wireless networks. To improve the learning performance of FL-GR, we have formulated an optimization problem to minimize global loss through device scheduling, RB allocation, and power control. To solve this problem, we have investigated the convergence bound of FL-GR and transformed the global loss minimization problem into a tractable one. Then, we derived the optimal power control for any given RB allocation policy and further transformed the global loss minimization problem into an equivalent linear programming problem, which can be solved efficiently. Simulation results on three real-world datasets (i.e., MNIST, CIFAR-10, and CIFAR-100) have shown that the proposed FL-GR achieves over 4% accuracy gain compared to the FL algorithms without gradient recycling. In addition, the proposed device scheduling algorithm outperforms the existing algorithm in accuracy and convergence speed.

Chapter 6

Efficient Wireless Federated Learning with Adaptive Model Pruning

6.1 Introduction

Most existing wireless FL studies focused on homogeneous model settings where devices train identical local models. In this setting, the devices with poor capabilities may delay the global model update and degrade the learning performance of FL since devices are usually drastically diverse in computation and communication capabilities. Moreover, in the homogenous model settings, the scale of the global model is restricted by the device with the lowest capability. To tackle these challenges, this work proposes an adaptive model pruning-based FL (AMP-FL) framework, where the edge server dynamically generates sub-models by pruning the global model for devices' local training to adapt their heterogeneous computation capabilities and time-varying channel conditions. Since the involvement of diverse structures of devices' sub-models in the global model updating may negatively affect the training convergence, we propose compensating for the gradients of pruned model regions by devices' historical gradients. The main contributions of this paper are listed as:

- We propose an adaptive model pruning-based FL (AMP-FL) framework, which dynamically prunes the global model to generate sub-models for adapting devices' communication and computation capabilities in the learning process. This framework effectively reduces communication and computation overhead for devices at the same time, enabling efficient FL over heterogeneous devices. To prevent the diverse sub-model structures from affecting the learning convergence, we propose compensating for the gradients of pruned regions by devices' historical gradients. In addition, we theoretically analyze the relationship between the pruning ratio and communication & computation load.

- We define an age of information (AoI) metric to characterize local gradients' staleness and theoretically analyze AMP-FL's convergence bound. The bound indicates that scheduling devices with large AoI and pruning the global model regions with small AoI are able to improve learning performance. Based on this, we define a new objective function, i.e., the average square of AoI of devices' gradients, and transform the inexplicit global loss minimization problem into a tractable one for guiding device scheduling, model pruning, and resource block allocation design.

- To solve the transformed problem, we first find the optimal model pruning policies for devices under a given RB allocation policy. On this basis, we transform it into an equivalent linear programming problem that can be effectively solved with polynomial time complexity. In addition, to improve the implementation feasibility of AMP-FL in practical wireless networks, we propose a memory-friendly AMP-FL equivalent to AMP-FL but with a low memory size requirement of the edge server.

- We conduct extensive simulations on two real-world datasets, i.e., MNIST and CIFAR-10, to verify the effectiveness of AMP-FL. Specifically, compared to the FL algorithms with the homogeneous local model settings, the proposed AMP-FL

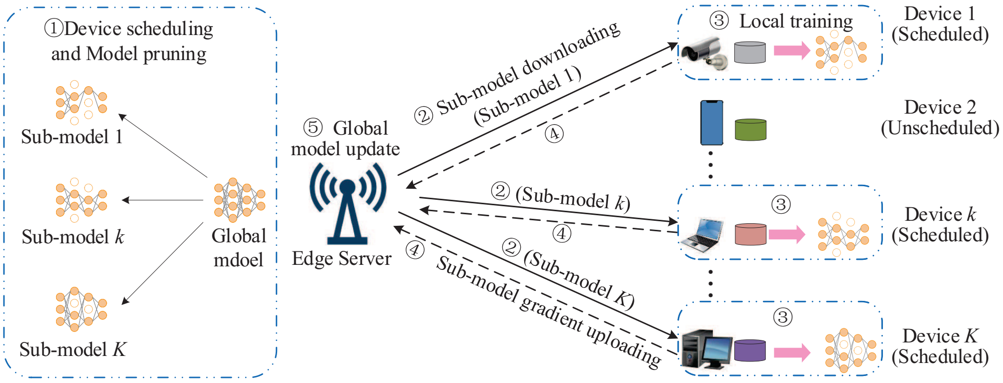


Figure 6.1: Illustration of the considered wireless FL system with adaptive model pruning.

is able to provide 1.9x and 1.6x speed up on MNIST and CIFAR-10, respectively. The proposed model pruning and device scheduling approach also obtains higher learning accuracy and faster convergence speed than the benchmark schemes.

The remainder of this chapter is organized as follow: Section 6.2 introduces the system model, the proposed AMP-FL framework, and the problem formulation. The convergence analysis and global loss minimization problem transformation are presented in Section 6.3. Section 6.4 illustrates the proposed model pruning, device scheduling, and RB allocation algorithm. Section 6.5 evaluates the effectiveness of the proposed approaches by simulations. The summary is presented in Section 6.6.

6.2 System Model and Learning Mechanism

This work considers a typical wireless FL system, as shown in Fig. 6.1, where K devices are orchestrated by an edge server to collaboratively train a shared global machine learning model, $\boldsymbol{w}$, by periodically uploading local gradient information to the edge server for global model update instead of transmitting the raw training data. To mitigate the negative effect of stragglers on learning performance, this work allows devices to train heterogeneous local models adapted to their computation and communication capabilities. The local models are obtained by pruning the global model using the proposed structured model pruning strategy (in Section 6.4.1) that dynamically adjusts the local models during the learning process with respect to devices' individual heterogeneous computation capabilities and time-varying communication conditions.

We assume that the global model can be partitioned into I disjoint regions indexed by $\mathcal{I} = \{1, 2, \cdots, I\}$, where each model region i is either one filter in convolution layers or one neuron in the fully-connected layers. Let $\boldsymbol{w}^{(i)}$ $(i \in \mathcal{I})$ denote the i-th region of the global model. The devices are indexed by $\mathcal{K} = \{1, 2, \cdots, K\}$. Each device k $(k \in \mathcal{K})$ has a local dataset $\mathcal{D}_k$ with $D_k = |\mathcal{D}_k|$ data samples. The entire dataset is denoted by $\mathcal{D} = \cup_{k=1}^{K} \mathcal{D}_k$ with $D = \sum_{k=1}^{K} D_k$ data samples. For any data sample $\zeta = (\boldsymbol{x}, y) \in \mathcal{D}$, a loss function $f(\boldsymbol{x}, y; \boldsymbol{w})$ is utilized to capture the fitting performance of model $\boldsymbol{w}$ on the input-output data pair $(\boldsymbol{x}, y)$. Thus, the local loss function of device k $(k \in \mathcal{K})$, i.e., $F_k(\boldsymbol{w})$, is given by $F_k(\boldsymbol{w}) = \frac{1}{D_k} \sum_{(\boldsymbol{x},y) \in \mathcal{D}_k} f(\boldsymbol{x}, y; \boldsymbol{w})$. The global loss function is given by $F(\boldsymbol{w}) = \sum_{k=1}^{K} a_k F_k(\boldsymbol{w})$, where a_k is the weight of device k such that $a_k \geq 0$ and $\sum_{k=1}^{K} a_k = 1$. Similar to many existing works, e.g., [59, 115, 117], we consider a balanced size of local datasets by setting $a_k = \frac{1}{K}$, $\forall k \in \mathcal{K}$. The goal of the FL system is to train a shared global model $\boldsymbol{w}$ so as to minimize the global loss $F(\boldsymbol{w})$ on the whole dataset $\mathcal{D}$, i.e., $\min_{\boldsymbol{w}} F(\boldsymbol{w})$.

6.2.1 Federated Learning with Adaptive Model Pruning

To improve the communication and computation efficiency for wireless FL, this work proposes a novel AMP-FL framework to adaptively generate sub-models for devices to train, as shown in Fig. 6.1. In addition, to alleviate the adverse effects of diverse structures of local models and partial participation in the learning performance, we propose compensating the gradients of pruned model regions and unscheduled devices by devices' historical gradients. The effectiveness of this gradient compensation mechanism is evaluated in Section 6.5. To this end, the edge server maintains a gradient array $\{\boldsymbol{G}_{k,t} : \forall k \in \mathcal{K}\}$ that caches the latest received gradients from devices. The learning process consists of T global rounds and running the following five steps in each round t

($t \in \{0, 1, \cdots, T-1\}$):

1) **Device Selection and Model Pruning**: The edge server selects a subset of devices to engage in the current round. Denote $\alpha_{k,t} \in \{0,1\}$ as the selection indicator of device k in t-th round, where $\alpha_{k,t} = 1$ represents device k is selected, $\alpha_{k,t} = 0$ otherwise. For ease of presentation, let $\boldsymbol{S}_t = \{k : \alpha_{k,t} = 1, \forall k \in \mathcal{K}\}$ denote the device scheduling decision in round t. After device selection, the edge server prunes the global model to generate sub-models for the scheduled devices according to their computing and communication capabilities. Let $\boldsymbol{m}_{k,t} = \{m_{k,t}^{(i)} : i = 1, 2, \cdots, I\}$ denote the pruning mask of device k in round t, where $m_{k,t}^{(i)} = 1$ represents that the i-th region of the global model is preserved in device k's sub-model, $m_{k,t}^{(i)} = 0$ otherwise. Thus, the sub-model of device k can be denoted as $\boldsymbol{w}_{k,t} = \boldsymbol{w}_t \odot \boldsymbol{m}_{k,t}$, where $\odot$ denote the element-wise product.

2) **Local Model Downloading**: Each selected device downloads its sub-model from the edge server.

3) **Local Model Training**: Each selected device trains its sub-model by performing λ-steps SGD. Specifically, device k ($k \in \boldsymbol{S}_t$) updates the i-th region ($\forall i \in \mathcal{I}, m_{k,t}^{(i)} = 1$) of its model as

$$\boldsymbol{w}_{k,t,l+1}^{(i)} = \boldsymbol{w}_{k,t,l}^{(i)} - \eta \nabla \tilde{F}_k(\boldsymbol{w}_{k,t,l}^{(i)}), l \in \{0, 1, 2, \cdots, \lambda - 1\}, \tag{6.1}$$

where $\boldsymbol{w}_{k,t,l}^{(i)}$ is the i-th region of device k's local model in the l-th iteration in round t with $\boldsymbol{w}_{k,t,0}^{(i)} = \boldsymbol{w}_t^{(i)}$, and η is the learning rate. In (6.1), the stochastic gradient $\nabla \tilde{F}_k(\boldsymbol{w}_{k,t,l}^{(i)})$ is given by $\nabla \tilde{F}_k(\boldsymbol{w}_{k,t,l}^{(i)}) = \frac{1}{L_b} \sum_{\zeta \in \mathcal{B}_{k,t,l}} \nabla f(\boldsymbol{w}_{k,t,l}^{(i)}, \zeta)$, where $\mathcal{B}_{k,t,l}$ is a mini-batch data uniformly sampled from $\mathcal{D}_k$ with $L_b = |\mathcal{B}_{k,t,l}|$ data samples.

4) **Local Gradient Uploading**: After finishing local training, each scheduled device k ($k \in \boldsymbol{S}_t$) uploads its cumulative local gradient, i.e., $\tilde{\boldsymbol{g}}_{k,t} = \{\tilde{\boldsymbol{g}}_{k,t}^{(i)} : \forall i \in \mathcal{I}, m_{k,t}^{(i)} = 1\}$, to the edge server, where $\tilde{\boldsymbol{g}}_{k,t}^{(i)}$ is given by $\tilde{\boldsymbol{g}}_{k,t}^{(i)} = \sum_{l=0}^{\lambda-1} \nabla \tilde{F}_k(\boldsymbol{w}_{k,t,l}^{(i)}) = \frac{1}{\eta}(\boldsymbol{w}_t^{(i)} - \boldsymbol{w}_{k,t,\lambda}^{(i)})$.

5) **Global Model Update**: After receiving the local gradients from the scheduled

devices, the edge server updates the gradient array as follows:

$$\boldsymbol{G}_{k,t}^{(i)} = \begin{cases} \tilde{\boldsymbol{g}}_{k,t}^{(i)}, & \alpha_{k,t} m_{k,t}^{(i)} = 1, \\ \boldsymbol{G}_{k,t-1}^{(i)}, & \alpha_{k,t} m_{k,t}^{(i)} = 0, \end{cases} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}. \tag{6.2}$$

Then, the edge server updates the global model as $\boldsymbol{w}_{t+1}^{(i)} = \boldsymbol{w}_t^{(i)} - \eta \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{G}_{k,t}^{(i)}, \forall i \in \mathcal{I}$.

6.2.2 Communication and Computation Load Model

Let $\mathcal{C}$ represent the number of FLOPs required to process one data sample on the global model, and $\mathcal{Q}$ indicate the number of global model parameters. In each round, devices train heterogeneous local models generated by pruning the global model to adapt to their communication and computation capabilities. For any device k ($k \in \mathcal{K}$) in round t with pruning mask $\boldsymbol{m}_{k,t}$, its pruning ratio is given by

$$\beta_{k,t} = 1 - \frac{1}{I} \sum_{i=1}^{I} m_{k,t}^{(i)}. \tag{6.3}$$

In fact, (6.3) indicates the ratio of pruned filters and neurons in the global model. To avoid introducing layer-wise hyperparameters, the proposed AMP-FL uses the same pruning ratio for every convolution or FC layer. Given the pruning ratio, AMP-FL removes a corresponding ratio of filters and neurons in each convolutional layer and FC layer to generate sub-models. In the following, we analyze the number of parameters and FLOPs for device k's sub-model from the perspective of convolution and FC layers.

1) For the l-th convolution layer in the global model with C_l filters, the number of parameters in this layer is $\mathcal{Q}_{g,l} = (K_c^2 \times C_{l-1} + 1) \times C_l$ which contains $K_c^2 \times C_{l-1} \times C_l$ weight parameters and C_l bias parameters; the number of FLOPs is $\mathcal{C}_{g,l} = 2K_c^2 HW C_{l-1} \times C_l$, where C_{l-1} is the number of filters in the $(l-1)$-th layer, K_c is the filter width (assumed to be symmetric), H and W are the height and width of the input feature maps [118]. For device k's sub-model with pruning ratio $\beta_{k,t}$, the number of parameters contained in its sub-model in this layer is $\mathcal{Q}_{k,l} = (1-\beta_{k,t})^2 K_c^2 \times C_{l-1} \times C_l + (1-\beta_{k,t}) \times C_l \approx (1-\beta_{k,t})^2 \mathcal{Q}_{g,l}$, and the number of FLOPs is $\mathcal{C}_{k,l} = 2(1-\beta_{k,t})^2 C_{l-1} C_l K_c^2 WH = (1-\beta_{k,t})^2 \mathcal{C}_{g,l}$.

2) For the l-th FC layer in the global model with N_l neurons, the number of parameters is $\mathcal{Q}_{g,l} = (N_{l-1} + 1) \times N_l$ which contains $N_{l-1} \times N_l$ weight parameters and N_l bias parameters; the number of FLOPs is $\mathcal{C}_{g,l} = 2N_{l-1} \times N_l$, where N_{l-1} is the number of neurons in the $(l-1)$-th FC layer. For device k's sub-model with pruning ratio $\beta_{k,t}$, the number of parameters contained in its sub-model is $\mathcal{Q}_{k,l} = (1-\beta_{k,t})^2 \times N_{l-1} \times N_l + (1-\beta_{k,t}) \times N_l \approx (1-\beta_{k,t})^2 \mathcal{Q}_{g,l}$, and the number of FLOPs is $\mathcal{C}_{k,l} = 2(1-\beta_{k,t})^2 \times N_{l-1} \times N_l = (1-\beta_{k,t})^2 \mathcal{C}_{g,l}$.

Note that in the above analysis, we approximate the number of parameters of both convolution and FC layers in the sub-model to be the ratio, i.e., $(1-\beta_{k,t})^2$, of that in the original global model. This is because the number of bias parameters is far less than that of weight parameters [60]. According to the above analysis, for each device k with pruning ratio β_k, the number of parameters and FLOPs for its sub-model can be approximately scaled by $(1-\beta_{k,t})^2$ of the global model. That is, the number of parameters of device k's sub-model is

$$\mathcal{Q}_k = (1-\beta_{k,t})^2 \mathcal{Q}, \tag{6.4}$$

and the corresponding number of FLOPs required to process one data sample is

$$\mathcal{C}_k = (1-\beta_{k,t})^2 \mathcal{C}. \tag{6.5}$$

6.2.3 Learning Latency Model

In the following, we characterize the per-round learning latency model for the proposed AMP-FL, including computation and communication latency.

1) **Computation Latency**: We denote f_k as the CPU frequency of device k. Each CPU cycle can process n_k FLOPs. Thus, the computation time of device k is

$$\mathcal{T}_{k,t}^{\mathrm{L}} = \frac{\lambda L_b \mathcal{C}_k}{f_k n_k} = \frac{\lambda L_b (1-\beta_{k,t})^2 \mathcal{C}}{f_k n_k}. \tag{6.6}$$

2) **Communication Latency**: This work considers the OFDMA is utilized with R RBs for devices to transmit their gradient information. The RBs are indexed by $\mathcal{R} = \{1, 2, \cdots, R\}$. Let $\boldsymbol{z}_{k,t} = (z_{k,t}^{(1)}, z_{k,t}^{(2)}, \cdots, z_{k,t}^{(R)})$ denote the RB allocation decision of device k in round t, where $z_{k,t}^{(r)} \in \{0, 1\}$, $z_{k,t}^{(r)} = 1$ represents that the r-th RB is allocated to device k, $z_{k,t}^{(r)} = 0$ otherwise. For ease of representation, we use $\boldsymbol{Z}_t = (\boldsymbol{z}_{1,t}, \boldsymbol{z}_{2,t}, \cdots, \boldsymbol{z}_{K,t})$ denote the RB allocation decisions for all devices in round t. Denote p_k as the transmit power of device k. Let $h_{k,t}$ represent the channel gain between device k and the edge server, and it remains unchangeable within one round but varies independently over rounds. Thus, the transmit rate of device k is $r_{k,t}(\boldsymbol{z}_{k,t}) = \sum_{r=1}^{R} z_{k,t}^{(r)} B \log_2(1 + \frac{p_k h_{k,t}}{I_r + BN_0})$, where B is the bandwidth of each RB, N_0 is the noise power spectral density. I_r is the interference caused by devices located in other service areas not participating in the FL process and using the same resource block [27, 35]. We consider that each device can only occupy at most one RB, and each RB can be accessed by at most one device. Thus, $\sum_{r=1}^{R} z_{k,t}^{(r)} \leq 1$ and $\sum_{k=1}^{K} z_{k,t}^{(r)} \leq 1$. Each parameter in devices' local gradients is quantized by q bits. Thus, the transmit time of device k to upload its gradient information is

$$\mathcal{T}_{k,t}^{\mathrm{U}} = \frac{\mathcal{Q}_k q}{r_{k,t}(\boldsymbol{z}_{k,t})} = \frac{(1 - \beta_{k,t})^2 \mathcal{Q} q}{r_{k,t}(\boldsymbol{z}_{k,t})}. \tag{6.7}$$

Note that the above analysis ignored the model pruning and global model updating latencies since the edge server is usually computationally powerful. The model pruning and updating latencies are negligible in comparison with the above communication and computation latency. In addition, we assume that the sub-model download latency is negligible since the edge server usually has more transmit power for the sub-model distribution compared to devices [13, 59]. The sub-model download latency is far smaller than the discussed communication and computation latency. It is worth mentioning that the proposed algorithm in Section 6.4 can be directly generalized in the case with non-negligible sub-model download latency by simply adding the sub-model download latency into the time constraint (6.8a).

6.2.4 Problem Formulation

This work focuses on improving the performance of the proposed AMP-FL by minimizing the global loss value after T global training rounds, i.e., $\mathbb{E}[F(\boldsymbol{w}_T)]$, where $\boldsymbol{w}_T$ is the global model in round T. Specifically, we jointly optimize the device scheduling, model pruning, and RB allocation strategies under latency and wireless resource restrictions. The optimizing problem is given by

$$\min_{\{\boldsymbol{S}_t,\boldsymbol{Z}_t,\boldsymbol{m}_t\}_{t=0}^{T-1}} \quad \mathbb{E}[F(\boldsymbol{w}_T)] \tag{6.8}$$

$$\text{s. t.} \quad \mathcal{T}_{k,t}^{\mathrm{L}} + \mathcal{T}_{k,t}^{\mathrm{U}} \leq \mathcal{T}_{\max}, \forall k \in \mathcal{K}, \forall t, \tag{6.8a}$$

$$\sum_{r=1}^{R} z_{k,t}^{(r)} \leq 1, \forall k \in \mathcal{K}, \forall t, \tag{6.8b}$$

$$\sum_{k=1}^{K} z_{k,t}^{(r)} \leq 1, \forall t, \tag{6.8c}$$

$$z_{k,t}^{(r)} \in \{0,1\}, \forall k \in \mathcal{K}, \forall t, \tag{6.8d}$$

$$\beta_{k,t} = 1 - \frac{1}{I}\sum_{i=1}^{I} m_{k,t}^{(i)}, \forall k \in \mathcal{K}, \forall t, \tag{6.8e}$$

$$0 \leq \beta_{k,t} \leq 1, \forall k \in \mathcal{K}, \forall t, \tag{6.8f}$$

$$m_{k,t}^{(i)} \in \{0,1\}, \forall k \in \mathcal{K}, \forall t, \tag{6.8g}$$

$$\alpha_{k,t} \in \{0,1\}, \forall k \in \mathcal{K}, \tag{6.8h}$$

where (6.8a) stipulates that the per-round latency cannot surpass its maximum allowed delay, $\mathcal{T}_{\max}$. (6.8b), (6.8c), and (6.8d) impose restrictions on the RB allocation decisions, indicating that one device can occupy at most one RB for uplink transmission and one RB can only be allocated to one device. (6.8e) characterizes the relationship between pruning policy and model pruning ratio for devices. (6.8f) prevents the model pruning ratio from exceeding 1 or lessening 0 since the edge server can prune at most the entire model or not prune for the global model. (6.8g) and (6.8h) correspond to the constraints related to the model pruning and device scheduling indicator domains, respectively.

Problem (6.8) is a typical integer programming that involves multi-dimensional discrete variables and is intractable to solve. In addition, solving problem (6.8) requires an explicit form of $\mathbb{E}[F(\boldsymbol{w}_T)]$ with respect to the device selection ($\boldsymbol{S}_t$), model pruning ($\boldsymbol{m}_t$), and RB allocation ($\boldsymbol{Z}_t$) policies, which is almost impossible since the evolution of the model vector is extremely complex during the training process. To this end, similar to many existing works, e.g., [12, 27, 33, 35], we turn to find an upper bound of the global loss function and optimize it for global loss minimization in Section 6.3.

6.3 Convergence Analysis and Problem Transformation

In this section, we theoretically characterize the convergence behaviour of AMP-FL to explore how the device schedule, model pruning, and RB allocation policies affect its learning performance. Based on the obtained convergence bound, we define a new objective function, i.e., the AoI for local gradients, to transform problem (6.8) into a tractable one for guiding the device selection, model pruning, and RB allocation design.

6.3.1 Convergence Analysis

This subsection analyzes the convergence behaviour of AMP-FL. For the sake of analysis, we define $\nabla F_k(\boldsymbol{w}_{k,t,l}) = \frac{1}{D_k}\sum_{(\boldsymbol{x},y)\in\mathcal{D}_k} \nabla f\left(\boldsymbol{x}, y; \boldsymbol{w}_{k,t,l}\right)$ as the full gradient of device k in the l-th iteration of round t, and $\tilde{\eta} = \eta\lambda$ as an auxiliary variable. Denote $F(\boldsymbol{w}^*)$ by the loss function of the optimal global model $\boldsymbol{w}^*$. Note that we use the latest received gradients of the pruned model regions of scheduled devices and unscheduled devices to update the global model. The staleness of these historical gradients may significantly affect the learning performance of the proposed AMP-FL. To characterize the impact of the staleness of devices' gradients on the learning performance, we define an AoI metric to identify the staleness of devices' gradients. Specifically, the AoI of the gradient in the

i-th region of device k is denoted by $\tau_{k,t}^{(i)}$, which evolves as

$$\tau_{k,t}^{(i)} = \begin{cases} \tau_{k,t-1}^{(i)} + 1, & \alpha_{k,t} m_{k,t}^{(i)} = 0, \\ 0, & \alpha_{k,t} m_{k,t}^{(i)} = 1, \end{cases} \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{I}. \tag{6.9}$$

To facilitate the analysis, we make the following standard assumptions which are widely used in the existing FL literature, e.g., [23, 27, 33, 35].

Assumption 10. *All the local loss functions, $F_k(\boldsymbol{w})$ ($\forall k \in \mathcal{K}$), are L-smooth. That is, for all $\boldsymbol{v}$ and $\boldsymbol{w}$, $\|\nabla F_k(\boldsymbol{w}) - \nabla F_k(\boldsymbol{v})\| \leq L \|\boldsymbol{w} - \boldsymbol{v}\|$.*

Assumption 11. *All the local loss functions, $F_k(\boldsymbol{w})$ ($\forall k \in \mathcal{K}$), are μ-strongly convex. That is, for all $\boldsymbol{v}$ and $\boldsymbol{w}$, $F_k(\boldsymbol{v}) \geq F_k(\boldsymbol{w}) + \langle F_k(\boldsymbol{w}), \boldsymbol{v} - \boldsymbol{w}\rangle + \frac{\mu}{2} \|\boldsymbol{v} - \boldsymbol{w}\|^2$.*

Assumption 12. *For the mini-batch data samples $\mathcal{B}_{k,t}$ that uniformly sampled from $\mathcal{D}_k$ on device k ($k \in \mathcal{K}$), the resulting stochastic gradient $\tilde{\nabla} F_k(\boldsymbol{w}_t)$ is an unbiased estimation of the full gradient $\nabla F_k(\boldsymbol{w}_t)$, i.e., $\mathbb{E}[\tilde{\nabla} F_k(\boldsymbol{w}_t)] = \nabla F_k(\boldsymbol{w}_t)$, and its variance is bounded by σ^2, i.e., $\mathbb{E}\|\tilde{\nabla} F_k(\boldsymbol{w}_t) - \nabla F_k(\boldsymbol{w}_t)\|^2 \leq \sigma^2$.*

Assumption 13. *The expected squared norm of devices' gradients is uniformly bounded by G^2, i.e., $\|\nabla F_k(\boldsymbol{w}_t)\|^2 \leq G^2$, for all $k = 1, 2, \cdots, K$ and $t = 0, 1, \cdots, T-1$.*

Before illustrating the convergence results of the proposed AMP-FL, we introduce two lemmas in the following to assist our convergence analysis.

Lemma 11. *Let Assumption 10, 12, and 13 hold, and the learning rate satisfies $\eta \leq \frac{1}{2\lambda L}$, the averaged drift of the local models from the global model after l iterations is bounded as*

$$\frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \sum_{i=1}^{I} \mathbb{E} \left\| \boldsymbol{w}_{k,t-\tau_{k,t}^{(i)}}^{(i)} - \boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)} \right\|^2 \leq 4(\lambda - 1) I (2\eta^2 \lambda G^2 + \eta^2 \sigma^2). \tag{6.10}$$

Proof. Please see Appendix D.1. □

Lemma 12. *Let Assumption 10, 12, and 13 hold, the averaged difference between the global model parameters in two different rounds is bounded as*

$$\sum_{k=1}^{K} \sum_{i=1}^{I} \mathbb{E} \left\| \boldsymbol{w}_t^{(i)} - \boldsymbol{w}_{t-\tau_{k,t}^{(i)}}^{(i)} \right\|^2$$

$$\leq 3\eta^2\bigg((\lambda^2+(\lambda-1)I)\sigma^2+(\lambda^2+2\lambda(\lambda-1)I)G^2\bigg)\sum_{k=1}^{K}\sum_{i=1}^{I}(\tau_{k,t}^{(i)})^2. \tag{6.11}$$

Proof. Please see Appendix D.2. □

Based on the above two lemmas, the one-round convergence bound of AMP-FL is derived as:

Theorem 5. *Let Assumption 10, 12, and 13 hold, and the learning rate satisfies $\eta \leq \frac{1}{2\lambda L}$, the one-round convergence bound is given by*

$$\mathbb{E}[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)] \leq (-\frac{1}{2}\eta+L\eta^2\lambda)\lambda\|\nabla F(\boldsymbol{w}_t)\|^2+c_1$$
$$+\frac{15\eta^2Lc_2}{4K}\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2, \tag{6.12}$$

where $c_1=4\eta(\lambda-1)IG^2+\left(4\eta^2L(\lambda-1)I+\frac{3}{4}\eta\lambda\right)\sigma^2$, $c_2=\lambda^2(\sigma^2+G^2)+(\lambda-1)I(\sigma^2+2\lambda G^2)$.

Proof. Please see Appendix D.3. □

According to Theorem 5, the summation of the square of each region's AoI in the local gradients, i.e., $\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2$, is a crucial factor that negatively affects the one-round convergence bound of AMP-FL. Minimizing $\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2$ through carefully designing the device scheduling and model pruning strategies is capable of narrowing the convergence bound for improving the learning performance. We have the following remark for the device scheduling and model pruning design.

Remark 6. *In practical wireless networks, only a small proportion of devices can be scheduled in each round due to the limited bandwidth resources. For device scheduling, one should schedule the devices that have a large summation of AoI over their model regions, i.e.,* $\sum_{i=1}^{I}(\tau_{k,t-1}^{(i)}+1)^2$*, since these devices are the main contributors for the term of* $\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2$*. In addition, for a scheduled device, one should preserve the model regions with large AoI while pruning the regions with small AoI.*

Based on Theorem 5, we further analyze the convergence bound of AMP-FL after T-rounds as follows:

Corollary 3. *Let Assumption 10-13 hold, the T-rounds convergence bound of AMP-FL is*

$$\begin{aligned}
&\mathbb{E}[F(\boldsymbol{w}_T) - F(\boldsymbol{w}^*)]\\
&\leq (1-\eta\lambda\mu+2L\eta^2\lambda^2\mu)^T\mathbb{E}[F(\boldsymbol{w}_0) - F(\boldsymbol{w}^*)] + \frac{1-(1-\eta\lambda\mu+2L\eta^2\lambda^2\mu)^T}{\eta\lambda\mu-2L\eta^2\lambda^2\mu}c_1\\
&+\frac{15}{4}\eta^2 Lc_2\sum_{t=0}^{T-1}(1-\eta\lambda\mu+2L\eta^2\lambda^2\mu)^{T-1-t}\frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2. \quad (6.13)
\end{aligned}$$

Proof. Please see Appendix D.4. □

From Corollary 3, the expected gap between $F(\boldsymbol{w}_T)$ and the optimal loss $F(\boldsymbol{w}^*)$ is bounded by three terms: 1) The initial gap between the global loss and the optimal loss. 2) A constant term related to the system hyperparameters caused by multiple local iterations ($\lambda > 1$) and stochastic gradient error. 3) The cumulative AoI of local gradients over T training rounds. The last term is highly related to model pruning, device scheduling, and wireless resource allocation policies. To minimize the global loss function, one can minimize the last term on the RHS of (6.13) through jointly designing the model pruning, device scheduling, and wireless resource allocation strategies. However, directly minimizing this term is impractical because it requires obtaining devices' channel state information during the entire learning course at the start of FL. To minimize the global loss, we have:

Remark 7. *Similar to many existing works, e.g., [27, 33, 35], the available wireless resource and devices are independent across different rounds in problem (6.8). Based on Theorem 5 and Corollary 3, we provide a reasonable objective function by decoupling the long-term problem into the training round level, i.e., $\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2$, which directly minimizes the upper bound on $\mathbb{E}[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)]$ and achieves global loss minimization.*

6.3.2 Problem Transformation

According to the convergence analysis results in Remark 6 and Remark 7, we transform problem (6.8) into minimize $\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2$ in each round (which is equivalent to maximize $\sum_{k=1}^{K}\sum_{i=1}^{I}\alpha_{k,t}m_{k,t}^{(i)}(\tau_{k,t-1}^{(i)}+1)^2$) for device scheduling, model pruning, and RB allocation policies design. Since $\alpha_{k,t}=\sum_{r=1}^{R}z_{k,t}^{(r)}\in\{0,1\}$, we have $\sum_{k=1}^{K}\sum_{i=1}^{I}\alpha_{k,t}m_{k,t}^{(i)}(\tau_{k,t-1}^{(i)}+1)^2=\sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{r=1}^{R}z_{k,t}^{(r)}m_{k,t}^{(i)}(\tau_{k,t-1}^{(i)}+1)^2$. In other words, when the RB allocation policy is determined, the device scheduling policy can be directly computed by $\alpha_{k,t}=\sum_{r=1}^{R}z_{k,t}^{(r)}$. Therefore, we transform problem (6.8) into the following problem:

$$\max_{\boldsymbol{Z}_t,\boldsymbol{m}_t}\ \sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{r=1}^{R}z_{k,t}^{(r)}m_{k,t}^{(i)}(\tau_{k,t-1}^{(i)}+1)^2 \tag{6.14}$$

$$\text{s. t.}\ \ (6.8a),(6.8b),(6.8c),(6.8d),(6.8e),(6.8f),(6.8g).$$

Problem (6.14) is a typical integer programming that is challenging to solve. In the following section, we develop an effective algorithm with polynomial time complexity to address its optimal solution. Note that problem (6.14) is to maximize the overall AoI across different devices and model regions. According to the evolution of AoI in Eq. (6.9), the model regions or devices are less frequently updated in the previous rounds have large AoI and thus tend to be selected to be updated in the current round. Thus, problem (6.14) helps regulate the updating frequency of diverse regions across devices, making each model region evenly trained on different devices and improving the learning performance.

6.4 Efficient Online Model Pruning and Resource Allocation

In this section, we develop an effective model pruning and RB allocation algorithm that solves problem (6.14). To this end, we first derive the optimal model pruning policy for devices under any given RB allocation policy. Based on the optimal pruning policy, we transform problem (6.14) into an equivalent linear programming problem which can be effectively solved. After that, to improve the implementation feasibility of AMP-FL in practical wireless networks, we propose a memory-friendly AMP-FL that is equivalent to the proposed AMP-FL in Section 6.2.1 but with a low memory requirement of the edge server.

6.4.1 Optimal Model Pruning Policy

For any given RB allocation policy $\boldsymbol{Z}_t$, the model pruning policies of devices do not affect each other and independently contribute to the objective function. That is, the model pruning policy of each device can be solely optimized. Motivated by this, we decompose the model pruning optimization problem for each scheduled device k ($k \in \boldsymbol{S}_t$) from problem (6.14) as follows:

$$\max_{\boldsymbol{m}_{k,t}} \quad \sum_{r=1}^{R} z_{k,t}^{(r)} \sum_{i=1}^{I} m_{k,t}^{(i)} (\tau_{k,t-1}^{(i)} + 1)^2 \tag{6.15}$$

$$\text{s. t.} \quad (6.8\text{e}), (6.8\text{f}), (6.8\text{g}),$$

$$\frac{\lambda L_b (1-\beta_{k,t})^2 \mathcal{C}}{f_k n_k} + \frac{(1-\beta_{k,t})^2 \mathcal{Q} q}{r_{k,t}(\boldsymbol{z}_{k,t})} \leq \mathcal{T}_{\max}, \tag{6.15a}$$

where constraint (6.15a) is obtained by rewrite constraint (6.8a). Problem (6.15) is a typical unweighted knapsack problem. Based on constraint (6.8e) and (6.15a), the pruning policy of device k should satisfy $\frac{1}{I}\sum_{i=1}^{I} m_{k,t}^{(i)} \leq \sqrt{\mathcal{T}_{\max}/(\frac{\lambda L_b \mathcal{C}}{f_k n_k} + \frac{\mathcal{Q} q}{r_{k,t}(\boldsymbol{z}_{k,t})})}$. Moreover, based on constraint (6.8f) and (6.8g), the number of preserved model regions, i.e., $\sum_{i=1}^{I} m_{k,t}^{(i)}$, should be an integer and not exceed total number regions of the global model, i.e., I. According to (6.15), one should preserve model regions as much as possible to increase the objective function value. Thus, the optimal pruning policy of device k satisfy

$$\frac{1}{I}\sum_{i=1}^{I} m_{k,t}^{(i)} = \min\left(\left\lfloor \sqrt{\frac{\mathcal{T}_{\max}}{\frac{\lambda L_b \mathcal{C}}{f_k n_k} + \frac{\mathcal{Q} q}{r_{k,t}(\boldsymbol{z}_{k,t})}}} \right\rfloor, 1 \right), \tag{6.16}$$

Algorithm 9 Adaptive Model Pruning Algorithm

```
1: Inputs: The AoI of device k's gradients in all model regions, i.e., $\{\tau_{k,t}^{(i)}, \forall i \in \mathcal{I}\}$. The RB allocation policy of device $k$, $\boldsymbol{z}_{k,t}$, device $k$'s CPU frequency $f_k$, channel gain $h_{k,t}$, and transmit power $p_k$.
2: Solve the optimal number of preserved model regions $\bar{\beta}_{k,t} = \sum_{i=1}^{I} m_{k,t}^{(i)}$ based on (6.16).
3: for each layer in the global model do
4:     Sort the regions in this layer according to their AoI (i.e., $\tau_{k,t}^{(i)}, \forall i \in \mathcal{I}$) in an descending order and then preserve the first $\bar{\beta}_{k,t}$ model regions and prune other regions.
5: end for
```

where $\lfloor \cdot \rfloor$ is the floor function which outputs the largest integer that does not exceed its input. From (6.16), when the RB allocation policy is given, the number of preserved regions for device k's sub-model is fixed. For the optimal model pruning policy of device k, we have the following remark:

Remark 8. *The optimal pruning policy for device k ($k \in \boldsymbol{S}_t$) is to preserve the model regions with large AoI while pruning the model regions with small AoI for maximizing the objective function of problem* (6.15)*.*

Note that, similar to many existing works, e.g., [63–65], this work adopts the width scaling approach to prune the global model, which removes a certain number of filters or neurons in each convolution layer or FC layer to generate a sub-model. To avoid introducing layer-wise hyperparameters, we use the same pruning ratio for every convolution or fully-connected layer. For each convolution layer or FC layer, we sort the filters or neurons based on their AoI in descending way, then gradually select the corresponding ratio (computed as (6.16)) of regions and remove the remaining regions. Let $\mathcal{L}$ denote the set of layers in the global model. Here, for each layer. many sorting algorithms can be utilized, e.g., Quicksort and Introsort, with a meagre time complexity of $\mathcal{O}(I_l \log I_l)$, where I_l is the number of filters or neurons in l-th layer of the global model. Due to $\sum_{l\in\mathcal{L}} I_l \log I_l \leq \sum_{l\in\mathcal{L}} I_l \log(\max_{l\in\mathcal{L}} I_l) = I \log(\max_{l\in\mathcal{L}} I_l)$, the model pruning process has a meagre time complexity of $\mathcal{O}(I \log(\max_{l\in\mathcal{L}} I_l))$. Thus, the proposed model pruning approach is easy to implement in practical wireless networks. We summarize the detailed steps of model pruning in Algorithm 9.

6.4.2 Optimal Resource Block Allocation

According to the above analysis, the optimal pruning strategy for each device k ($k \in \mathcal{K}$) can be solved when it accesses any RB r ($r \in \mathcal{R}$) using Algorithm 9, denoted as $\boldsymbol{m}_{k,t,r}^{*} = \{m_{k,t,r}^{(i,*)} : \forall i \in \mathcal{I}\}$. Based on this, we compute the optimal model pruning policies for all devices when they access any RB (i.e., $\{\boldsymbol{m}_{k,t,r}^{*} : \forall r \in \mathcal{R}, \forall k \in \mathcal{K}\}$) and then substitute them into problem (6.14). Thus, problem (6.14) can be simplified as the following equivalent RB allocation problem:

$$\begin{aligned} \max_{\boldsymbol{Z}_t} \quad & \sum_{k=1}^{K}\sum_{r=1}^{R} z_{k,t}^{(r)} \sum_{i=1}^{I} m_{k,t,r}^{(i,*)} (\tau_{k,t-1}^{(i)} + 1)^2 \qquad (6.17) \\ \text{s. t.} \quad & (6.8b), (6.8c), (6.8d). \end{aligned}$$

Problem (6.17) is a typical integer programming which is difficult to solve. Below we reformulate it as a maximum weight bipartite matching problem and find its optimal solution. To this end, we construct a complete and balanced bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{K} \cup \bar{\mathcal{R}}$ is the vertex set, and $\mathcal{E}$ is the set of edges that connect the vertices in K and $\bar{\mathcal{R}}$. In graph $\mathcal{G}$, each vertex k ($k \in \mathcal{K}$) corresponds to a device k. $\bar{\mathcal{R}} = \mathcal{R} \cup \mathcal{R}_v$ is an extended set of $\mathcal{R}$, where each vertex r ($r \in \mathcal{R}$) corresponds to r-th RB, $\mathcal{R}_v$ is the virtual vertex set used to construct a balanced bipartite graph. The weight of edges is given by

$$\Omega_{k,r} = \begin{cases} \sum\limits_{i=1}^{I} m_{k,t,r}^{(i,*)} (\tau_{k,t-1}^{(i)} + 1)^2, & \text{if } \; k \in \mathcal{K}, r \in \mathcal{R}, \\ 0, & \text{else.} \end{cases} \qquad (6.18)$$

Based on the above defined bipartite graph $\mathcal{G}$, problem (6.17) can be transformed to find a maximum weight perfect matching of graph $\mathcal{G}$. Let $\theta_{k,r} \in \{0, 1\}$ be the edge connecting vertex k and vertex r, where $\theta_{k,r} = 1$ denotes RB r is assigned to device k, and $\theta_{k,r} = 0$ otherwise. Denote $\boldsymbol{\theta}_k = \{\theta_{k,1}, \theta_{k,2}, \cdots, \theta_{k,R}\}$ by the edge connection indicator of device k to all RBs. The bipartite matching problem is given by:

$$\max_{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_K} \quad \sum\nolimits_{k=1}^{K} \sum\nolimits_{r=1}^{|\bar{\mathcal{R}}|} \theta_{k,r} \Omega_{k,r} \qquad (6.19)$$

$$\text{s. t.} \quad \sum\nolimits_{r=1}^{|\bar{\mathcal{R}}|} \theta_{k,r} = 1, \qquad (6.19a)$$

$$\sum\nolimits_{k=1}^{K} \theta_{k,r} = 1, \tag{6.19b}$$

$$\theta_{k,r} \in \{0,1\}, \forall k \in \mathcal{K}, \forall r \in \bar{\mathcal{R}}. \tag{6.19c}$$

Note that any solution of problem (6.19) corresponds to a perfect matching of graph $\mathcal{G}$. The constraints (6.19a), (6.19b), and (6.19c) are corresponding to the constraints (6.8b), (6.8c), and (6.8d), respectively. To find the optimal solution of problem (6.19), an intuitive approach is to calculate the objective value of all perfect matching of graph $\mathcal{G}$, and let the matching with maximum objective value as the final RB allocation policy. However, this approach may be infeasible in practice since there is a total of $K!$ perfect matching of graph $\mathcal{G}$, which has an exponential time complexity since $K! > \sqrt{2\pi K}(\frac{K}{e})^K$. By relaxing the integrality constraint (6.19c), problem (6.19) can be relaxed as the following linear programming:

$$\max_{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\cdots,\boldsymbol{\theta}_K} \quad \sum\nolimits_{k=1}^{K} \sum\nolimits_{r=1}^{|\bar{\mathcal{R}}|} \theta_{k,r}\Omega_{k,r} \tag{6.20}$$

$$\text{s. t.} \quad (6.19\text{a}), (6.19\text{b}),$$

$$0 \leq \theta_{k,r} \leq 1, \forall k \in \mathcal{K}, \forall r \in \bar{\mathcal{R}}. \tag{6.20a}$$

It is worth mentioning that in problem (6.20), each row in the coefficient matrix corresponding to (6.19a) and (6.19b) only contains a '1'. This implements that each square submatrix of this coefficient has determinant equal to 0, 1, or -1. Thus, this coefficient matrix is a totally unimodular matrix. Based on [114], the optimal solution of problem (6.20) is an integer solution. That is, the optimal solution of problem (6.20) equals to the optimal solution of problem (6.19). Therefore, we directly solve problem (6.20) to obtain the optimal solution of (6.19). Since problem (6.20) is a linear programming, we use the current matrix multiplication time algorithm [119] to solve it with a time complexity of $\mathcal{O}((K^{2+1/6})^2)$.

6.4.3 Complexity Analysis and Implementation

In the above analysis, we first transform problem (6.14) into an equivalent maximum weight perfect bipartite matching problem, i.e., problem (6.19). Then, we further

Algorithm 10 Efficient Device Scheduling, Model Pruning, and RB Allocation Algorithm

```
Inputs: The AoI of devices' gradients in all model regions, $\{\tau_{k,t}^{(i)}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}\}$. Device $k$'s CPU frequency $f_k$, channel gain $h_{k,t}$, and transmit power $p_k$.
Solve the optimal model pruning policy for each device $k$ ($k \in \mathcal{K}$) at each RB $r$ ($r \in \mathcal{R}$) using Algorithm 9.
Construct the linear programming problem (6.20).
Solve (6.20) by the current matrix multiplication time algorithm [119] and obtain the optimal solution $\{\theta_{k,r}^{*}, \forall k \in \mathcal{K}, \forall r \in \bar{\mathcal{R}}\}$.
Compute the RB allocation policy for each device $k$ ($k \in \mathcal{K}$) as $\boldsymbol{z}_{k,t}^{*} = \{z_{k,t}^{(r,*)}, \forall r \in \mathcal{R}\}$ where $z_{k,t}^{(r,*)} = \theta_{k,r}^{*}$.
Compute the device scheduling policy as $\boldsymbol{S}_t^{*} = \{\alpha_{k,t}^{*} = 1, \forall k \in \mathcal{K}\}$ where $\alpha_{k,t}^{*} = \sum_{r=1}^{R} z_{k,t}^{(r,*)}$.
Find the optimal model pruning policies for each scheduled device $k \in \boldsymbol{S}_t^{*}$, denoted as $\{\boldsymbol{m}_{k,t}^{*}, \forall k \in \boldsymbol{S}_t^{*}\}$

return The device scheduling policy $\boldsymbol{S}_t^{*}$, model pruning policy $\boldsymbol{m}_{k,t}^{*}$, and RB allocation policy $\boldsymbol{z}_{k,t}^{*}$.
```

transform problem (6.19) into its equivalent linear programming (6.20). It is worth mentioning that these are two equivalent transformations and do not change the optimality of problem (6.14). Thus, the optimal solution of problem (6.14) can be addressed by first solving the optimal solution of problem (6.20). When the optimal solution of problem (6.20) is found, the optimal RB allocation is determined. Furthermore, the optimal device scheduling policy can be computed by $\alpha_{k,t}^{*} = \sum_{r=1}^{R} z_{k,t}^{(r),*}$ ($\forall k \in \mathcal{K}$), and the optimal model pruning policy of each device can be determined by Algorithm 9. For clarity, we summarize the detailed steps for solving problem (6.14) in Algorithm 10. In Algorithm 10, constructing the linear programming problem (6.20) requires running $K \times R$ times of Algorithm 9 to calculate the optimal model pruning policy for each device k ($k \in \mathcal{K}$) at each RB r ($r \in \mathcal{R}$). Thus the overall time complexity to solve the problem (6.14) is $\mathcal{O}\left(KRI \log I + (K^{2+1/6})^2\right)$.

In practical wireless networks, implementing the proposed AMP-FL in Section 6.2.1 requires the edge server to maintain the gradient information for all devices. Thus, the memory size requirement of the edge server scales with the model size and the number of devices. With the increase in device number, the memory space of the edge server may be exhausted and thus restrict the scale of the FL system and the global model. To tackle this issue, we distribute the memory requirement to devices for forming a memory-friendly AMP-FL which is equivalent to the proposed AMP-FL in Section 6.2.1. As a result, the edge server only need to maintain a single gradient array, $\bar{\boldsymbol{G}}_t$, to cache the aggregated local gradient information, and each device maintains a gradient array $\boldsymbol{G}_{k,t}$, to cache its previous latest gradient. Then we replace step 4) and step 5) in Section 6.2.1 with the following steps:

- Replace step 4) in Section 6.2.1 with: After finishing the local training process, each scheduled device k ($k \in \boldsymbol{S}_t$) uploads the difference between its current and previous gradient, i.e., $\bar{\boldsymbol{G}}_{k,t}^{(i)} = \tilde{\boldsymbol{g}}_{k,t}^{(i)} - \boldsymbol{G}_{k,t-1}^{(i)}$, to the edge server.
- Replace step 5) in Section 6.2.1 with: After receiving devices' gradient information, the edge server updates the maintained gradient according to $\bar{\boldsymbol{G}}_t^{(i)} = \bar{\boldsymbol{G}}_{t-1}^{(i)} + \frac{1}{K}\sum_{k=1}^{K} \bar{\boldsymbol{G}}_{k,t}^{(i)}$. Then, the edge server updates the global model as $\boldsymbol{w}_{t+1}^{(i)} = \boldsymbol{w}_t^{(i)} - \eta \bar{\boldsymbol{G}}_t^{(i)}$.

By replacing step 4) and step 5) in Section 6.2.1 with the above two steps, the edge server distributes the memory requirement to the devices. We summarise the steps of implementing this memory-friendly AMP-FL in Algorithm 11.

In the following theorem, we prove the equivalence of Algorithm 11 and the proposed AMP-FL in Section 6.2.1.

Theorem 6. *Algorithm 11 is equivalent to the proposed AMP-FL in Section 6.2.1.*

Proof. We prove Theorem 6 by mathematical induction approach. Firstly, the maintained gradient array $\bar{\boldsymbol{G}}_t$ at the edge server satisfies:

$$\begin{aligned}\bar{\boldsymbol{G}}_t^{(i)} &= \bar{\boldsymbol{G}}_{t-1}^{(i)} + \frac{1}{K}\sum_{k=1}^{K} \alpha_{k,t} m_{k,t}^{(i)} \bar{\boldsymbol{G}}_{k,t}^{(i)} \\ &= \bar{\boldsymbol{G}}_{t-1}^{(i)} + \frac{1}{K}\sum_{k=1}^{K} \alpha_{k,t} m_{k,t}^{(i)} \left(\tilde{\boldsymbol{g}}_{k,t}^{(i)} - \boldsymbol{G}_{k,t-1}^{(i)}\right) \\ &= \bar{\boldsymbol{G}}_{t-1}^{(i)} + \frac{1}{K}\sum_{k=1}^{K} \left(\boldsymbol{G}_{k,t}^{(i)} - \boldsymbol{G}_{k,t-1}^{(i)}\right). \end{aligned} \tag{6.21}$$

Note that at the beginning of the learning process, the devices' gradient array $\boldsymbol{G}_{k,-1}$ and

Algorithm 11 Memory-friendly AMP-FL

```
**Initialization:** The edge server initials its gradient array $\bar{\boldsymbol{G}}_{-1} = \mathbf{0}$ and the global model $\boldsymbol{w}_0$, each device $k$ ($k \in \mathcal{K}$) initial their gradient array as $\boldsymbol{G}_{k,-1} = \mathbf{0}$
**Server side:**
for $t = 0, 1, \cdots, T-1$ do
    Determine the scheduled devices and generate a sub-model for each scheduled device through model pruning.
    if Receive the gradient information from the selected devices then
        Update the gradient array $\bar{\boldsymbol{G}}_t$ as
        $\bar{\boldsymbol{G}}_t^{(i)} = \bar{\boldsymbol{G}}_{t-1}^{(i)} + \frac{1}{K}\sum_{k=1}^{K} \alpha_{k,t} m_{k,t}^{(i)} (\tilde{\boldsymbol{g}}_{k,t}^{(i)} - \boldsymbol{G}_{k,t-1}^{(i)})$
        Update the global model as $\boldsymbol{w}_{t+1}^{(i)} = \boldsymbol{w}_t^{(i)} - \eta \bar{\boldsymbol{G}}_t^{(i)}$.
    else
        $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t$
    end if
end for
**Device side:**
if Device $k$ is scheduled then
    Download its corresponding sub-model from the edge server;
    for $l = 0, 1, \cdots, \lambda - 1$ do
        Perform local training according to (6.1);
    end for
    Compute the cumulative stochastic gradient $\tilde{\boldsymbol{g}}_{k,t}^{(i)} = \frac{1}{\eta}(\boldsymbol{w}_t^{(i)} - \boldsymbol{w}_{k,t,\lambda}^{(i)})$
    Upload the $\tilde{\boldsymbol{g}}_{k,t} - \boldsymbol{G}_{k,t-1}$ to the edge server.
    Update the gradient array $\boldsymbol{G}_{k,t}$ according to (6.2).
end if
```

the server's gradient array $\boldsymbol{G}_{-1}$ are all initialized with $\mathbf{0}$. Thus, when $t = 0$, we have

$$\bar{\boldsymbol{G}}_0^{(i)} = \bar{\boldsymbol{G}}_{-1}^{(i)} + \frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{G}_{k,0}^{(i)} - \boldsymbol{G}_{k,-1}^{(i)}\right) = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{G}_{k,0}^{(i)}. \tag{6.22}$$

When $t = 1$,

$$\begin{aligned}\bar{\boldsymbol{G}}_1^{(i)} &= \bar{\boldsymbol{G}}_0^{(i)} + \frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{G}_{k,1}^{(i)} - \boldsymbol{G}_{k,0}^{(i)}\right) \\ &= \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{G}_{k,1}^{(i)} + \bar{\boldsymbol{G}}_0^{(i)} - \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{G}_{k,0}^{(i)} \\ &= \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{G}_{k,1}^{(i)}. \end{aligned} \tag{6.23}$$

Similarly, for $t > 1$, $\bar{\boldsymbol{G}}_t^{(i)} = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{G}_{k,t}^{(i)}$. Thus, the updated global model through Algorithm 11 is $\boldsymbol{w}_{t+1}^{(i)} = \boldsymbol{w}_t^{(i)} - \eta\bar{\boldsymbol{G}}_t^{(i)} = \boldsymbol{w}_t^{(i)} - \eta\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{G}_{k,t}^{(i)}$, which equals the updated global model by the proposed AMP-FL in Section 6.2.1. Thus, Algorithm 11 is equivalent to the AMP-FL algorithm in Section 6.2.1. □

Table 6-A: System Parameters

Parameter	Value	Parameter	Value
K	100	R	10
B	1 MHz	N_0	-174 dBm/Hz
n_k ($\forall k \in \mathcal{K}$)	4	h_0	-30 dBm
q	32 bits	η	0.05
τ	8	L_b	64
Q (CNN)	36,758	C_k (CNN)	782,816
$\mathcal{T}_{\max}$ (CNN)	0.1s	Q (VGG-11)	9,287,434
C_k (VGG-11)	362,285,568	$\mathcal{T}_{\max}$ (VGG-11)	20 s
p_k ($\forall k \in \mathcal{K}$)	30 dBm	s	2

6.5 Simulation Results

In this section, simulations are conducted to evaluate the performance of the proposed AMP-FL algorithm and device scheduling approach. If not specified, the default system settings are given as Table 6-A. We consider an edge server situated at the centre of a circular area with a radius of 500m serving K randomly distributed devices. The channel gain is modelled as $h_{k,t} = h_0 \rho_{k,t} d_k^{-2}$, where d_k is the distance from device k to the edge server, $\rho_{k,t} \sim \text{Exp}(1)$ is the Rayleigh fading channel gain [12, 98]. For each device, its CPU frequency is uniformly selected from $\{0.85, 1.12, 1.2, 1.3\}$GHz. Similar to [27, 35], we do not compute the exact value of the interference (I_m, $\forall r \in \mathcal{R}$) since we mainly focus on FL system instead of other service areas. The inter-cell interference at each RB r, i.e., I_r, is randomly selected from the range of $\left[10^2 BN_0, 10^5 BN_0\right]$. For each device, its CPU frequency is uniformly selected from $\{0.85, 1.12, 1.2, 1.3\}$ GHz, and each CPU cycle can process 4 FLOPs.

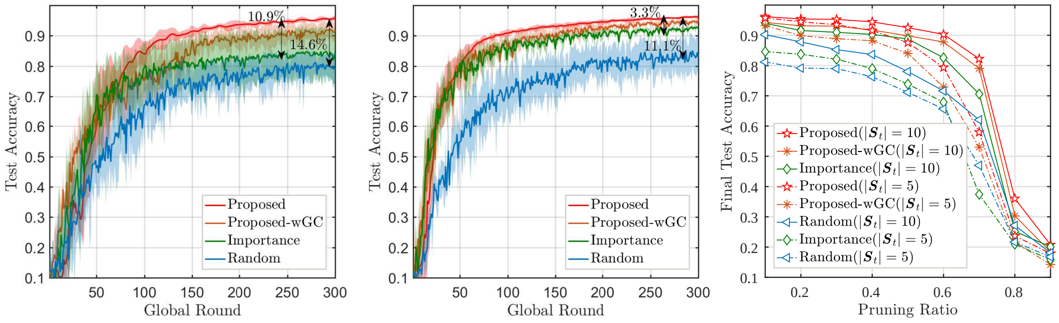
We evaluate the proposed approaches on two typical classification tasks using MNIST and CIFAR-10 datasets. For the MNIST dataset, we train a CNN with the following structure: two 5 × 5 convolution layers with 6 and 16 channels, respectively, and each of them is followed by a 2 × 2 max-pooling layer; a 128-neuron FC layer; and a 10-unit softmax output layer. For CIFAR-10, we train a VGG-11 model [69]. Note that the original VGG-11 has 1000 output units. To adapt VGG-11 to the CIFAR-10 dataset, we remove its last max-pooling layer, then replace its FC layers as the following structure:

two FC layers with 512 and 128 neurons; a 10-unit softmax output layer. We utilize a typical non-IID data partitioning method for both the above datasets as follows: we sort all data samples according to the label, then divide them into $sK/10$ shards and assign each device with s shard. By this means, each device obtains at most s types of data in the dataset. If not specified, $s = 2$. For all the above-illustrated two models, cross-entropy is used as the loss function.

6.5.1 Comparison of Model Pruning Strategies

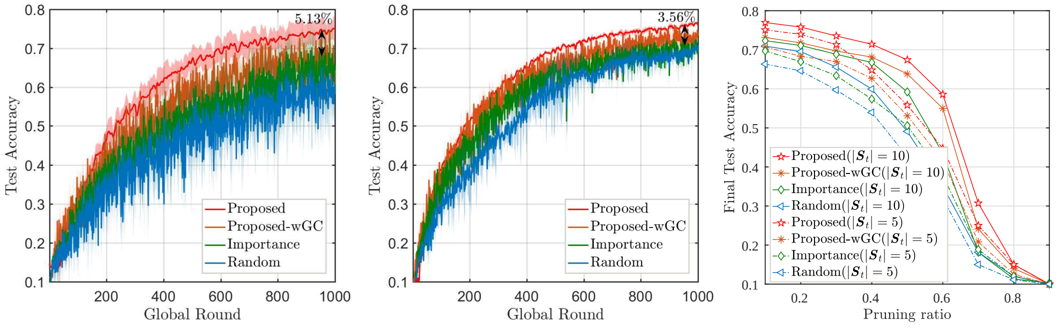
In this subsection, we evaluate the proposed model pruning approach by comparing it with the following approaches under different device schedule numbers and pruning ratios: 1) The proposed model pruning without gradient compensation (Proposed-wGC): The edge server utilizes the proposed model pruning approach (i.e., Algorithm 9) to generate sub-models. However, the server only uses the received sub-model gradients from devices for global model updating without compensating the pruned model regions' gradients. The gradients of the pruned model regions of devices are set to zero for aligning the model architecture. That is, all devices' gradients have the same structure as the global model. 2) Importance-aware model pruning: In each round, the edge server removes the less important filters and neurons in the global model to generate sub-models. The importance score of each filter in the convolution layers is computed as the kernel weights summation. The importance score of each neuron in the FC layer is calculated as its connected input weights summation [120]. 3) Random pruning [60]: In each round, the server randomly prunes the global model to generate sub-models for devices based on their pruning ratios.

Fig. 6.2 compares the learning performance of FL with different model pruning policies on the MNIST dataset. By setting the pruning ratio of all devices to 0.1, Fig. 6.2(a) and Fig. 6.2(b) test the performance of all the model pruning policies under $|\boldsymbol{S}_t|= 5$ and $|\boldsymbol{S}_t|= 10$, respectively. Compared to the benchmarks, the proposed approach improves over 10.9% and 3.3% accuracy when $|\boldsymbol{S}_t|= 5$ and $|\boldsymbol{S}_t|= 10$, respectively. In



(a) (b) (c)

Figure 6.2: Comparison of the learning performance of FL with different pruning strategies on the MNIST dataset: (a) Test accuracy, $|\boldsymbol{S}_t|= 5$, $\beta_{k,t} = 0.1$, (b) Test accuracy, $|\boldsymbol{S}_t|= 10$, $\beta_{k,t} = 0.1$, (c) Final test accuracy of FL after 300 rounds of training.



(a) (b) (c)

Figure 6.3: Comparison of the learning performance of FL with different pruning strategies on the CIFAR-10 dataset: (a) Test accuracy, $|\boldsymbol{S}_t|= 5$, $\beta_{k,t} = 0.1$, Test accuracy, $|\boldsymbol{S}_t|= 10$, $\beta_{k,t} = 0.1$, (c) Final test accuracy of FL after 1000 rounds of training..

addition, the proposed pruning approach with gradient compensation performs better than that without gradient compensation. This demonstrates the effectiveness of the proposed gradient compensation mechanism. Fig. 6.2(c) shows how the pruning ratio affects the final accuracy of the global model trained under different pruning policies. Note that all the accuracy results in Fig. 6.2(c) are obtained by training the global model under corresponding pruning policies with 300 rounds. We can see that the proposed pruning approach outperforms the benchmarks under different pruning ratios and participant numbers. Moreover, for all pruning policies, the final accuracy under

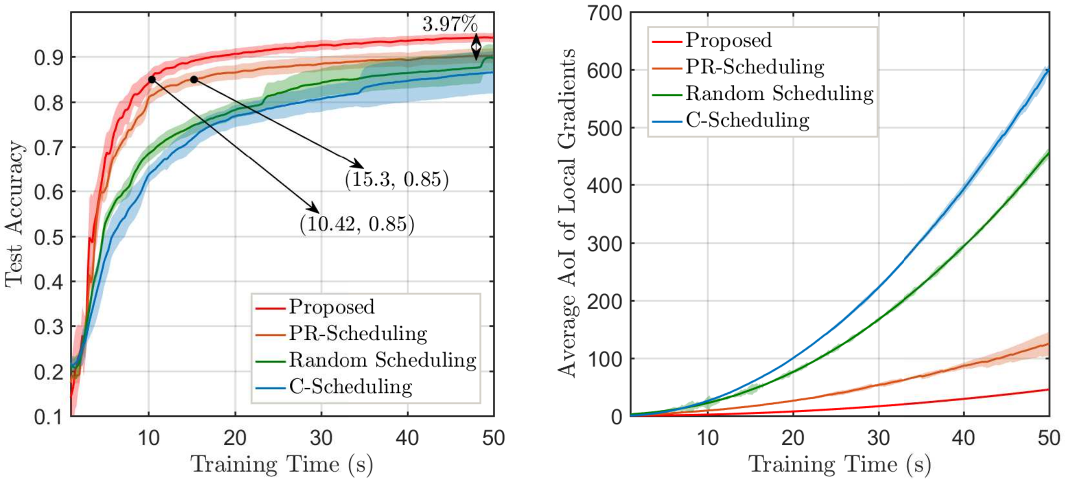
$|\boldsymbol{S}_t|= 10$ is higher than that under $|\boldsymbol{S}_t|= 5$. This indicates scheduling more devices in each round improves the learning performance of AMP-FL. In addition, the final model accuracy under all pruning policies decreases with the increase in the pruning ratio. This is because a larger pruning ratio induces that the sub-models have fewer parameters, and more filters and neurons have been trained fewer times.

A similar comparison is conducted on the CIFAR-10 dataset in Fig. 6.3. It is also observed that the proposed model pruning policy converges faster than the benchmarks. When the pruning ratios of all approaches are set to be 0.1, the proposed approach is capable of boosting 5.13% and 3.56% accuracy under $|\boldsymbol{S}_t|= 5$ and $|\boldsymbol{S}_t|= 10$, respectively. It is worth mentioning that the proposed approach with gradient compensation outperforms that without gradient compensation. In addition, the proposed-wGC approach remains performs better than the other two benchmarks. This demonstrated the effectiveness of the proposed gradient compensation mechanism and model pruning approach.

6.5.2 Comparison of Device Scheduling Policies

In this section, we evaluate the effectiveness of the proposed device scheduling and resource allocation approach by comparing it with: 1) Pruning ratio minimization-aware device scheduling (PR-scheduling) [59, 61]: In each round, the edge server selects a subset of devices that satisfies the latency constraint and has the minimal sum of the pruning ratio. 2) Channel gain-aware device scheduling (C-scheduling) [35]: The edge server schedules the devices with maximal channel gain and satisfies the latency constraint to perform training in each round. 3) Random scheduling. In each round, the edge server randomly selects a subset of devices and their corresponding RBs that satisfy the latency constraint.

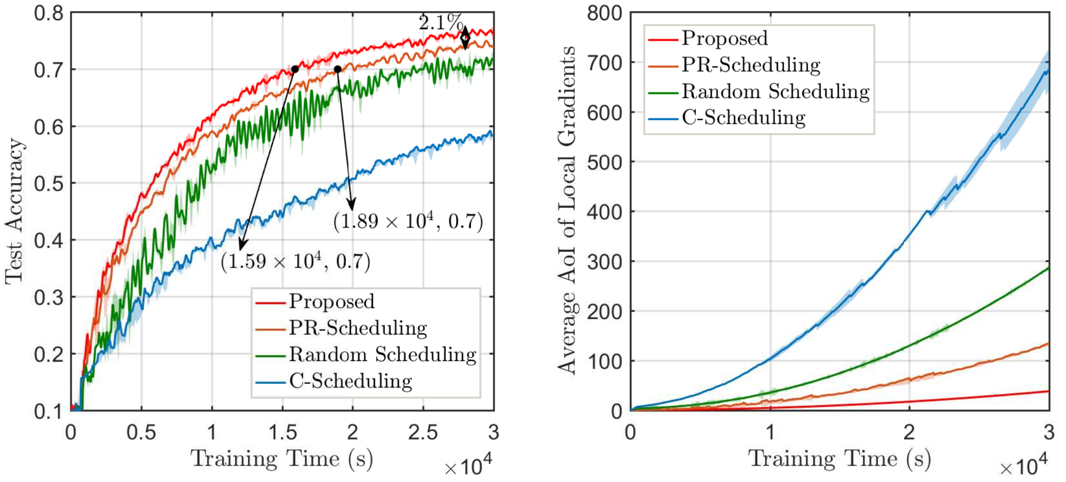
Fig. 6.4 compares the proposed scheduling approach with the above three approaches on MNIST dataset. From Fig. 6.4(a), compared to the benchmarks, the proposed device scheduling achieves higher accuracy and faster convergence speed. Specifically, the pro-



(a)

(b)

Figure 6.4: Comparison of learning performance for different device scheduling approaches on MNIST dataset.



(a)

(b)

Figure 6.5: Comparison of learning performance for different device scheduling approaches on CIFAR-10 dataset.

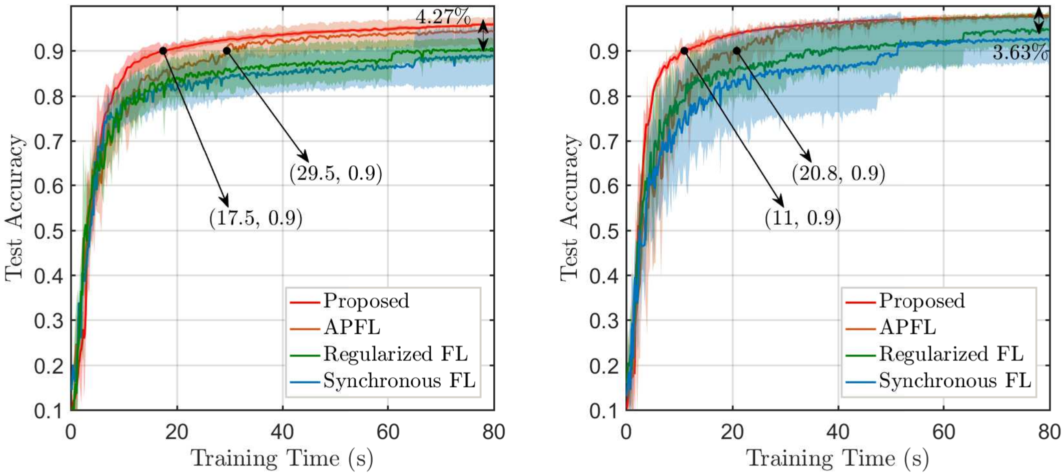
posed device scheduling approach boosts at least 3.97% accuracy than the benchmarks. Given the target accuracy is 85%, the proposed device scheduling approach only takes 10.42 seconds to achieve the target, while the best benchmark, i.e., the PR-scheduling scheme, requires 15.3 seconds. Compared to the benchmarks, the proposed approach is able to save 31.9% training time to obtain 85% test accuracy. The latent reason why the proposed device scheduling approach outperforms the benchmarks is illustrated in Fig. 6.4(b), which plots the average AoI of devices' local gradients. We find that the proposed method possesses the lowest average AoI of local gradients. In addition, for all the device scheduling algorithms, the one with lower AoI obtains higher learning accuracy. This phenomenon demonstrated the convergence results in Remark 6, which suggests minimizing the average AoI of local gradients to enhance the learning performance.

Fig. 6.5 evaluates the learning performance of all the device scheduling approaches on the CIFAR-10 dataset and shows the same conclusion as MNIST. From Fig. 6.5(a), the proposed approach achieves 2.1% accuracy improvement after 3×10^4 seconds of training. Given the target accuracy is 70%, the proposed device scheduling approach saves at least 15.87% training time compared to the benchmarks. In addition, Fig. 6.5(b) shows that the proposed device scheduling approach has the lowest average AoI of gradients compared to the benchmarks. This further demonstrated the correctness of the convergence results in Remark 6.

6.5.3 Overall Effectiveness

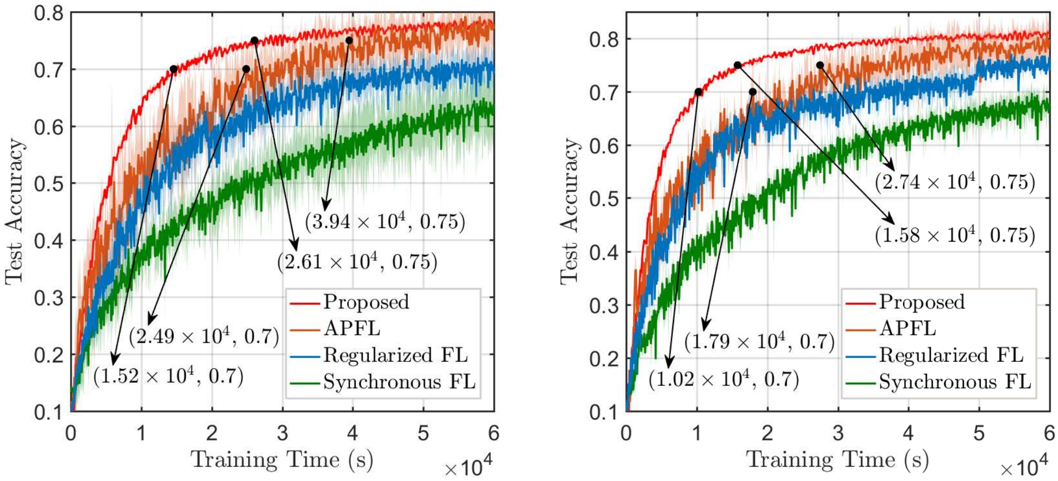
This subsection evaluates AMP-FL by comparing it to three FL algorithms as follows: 1) Synchronous FL [27, 33, 35]: The scheduled devices train the entire global model and upload the trained model to the edge server for aggregation. 2) Regularized FL [121]: Regularized FL utilizes a weight-based proximal term to limit the impact of local updates to tackle the data heterogeneity among devices. 3) Adaptive personalized FL (APFL) [108]: The selected devices train their local models and the received global model. After that, APFL integrates devices' local models and global model to create a personalized model for each device.

Fig. 6.6 shows the learning performance of AMP-FL and three benchmarks on MNIST dataset. Fig. 6.6(a) sets the data heterogeneity-related parameter to $s = 2$, i.e., each device in the system has at most two types of data samples of the MNIST dataset.



(a) (b)

Figure 6.6: Learning performance of different FL algorithms on the MNIST dataset: (a) $s = 2$, (b) $s = 3$.



(a) (b)

Figure 6.7: Learning performance for different FL algorithms on the CIFAR-10 dataset: (a) $s = 2$, (b) $s = 3$.

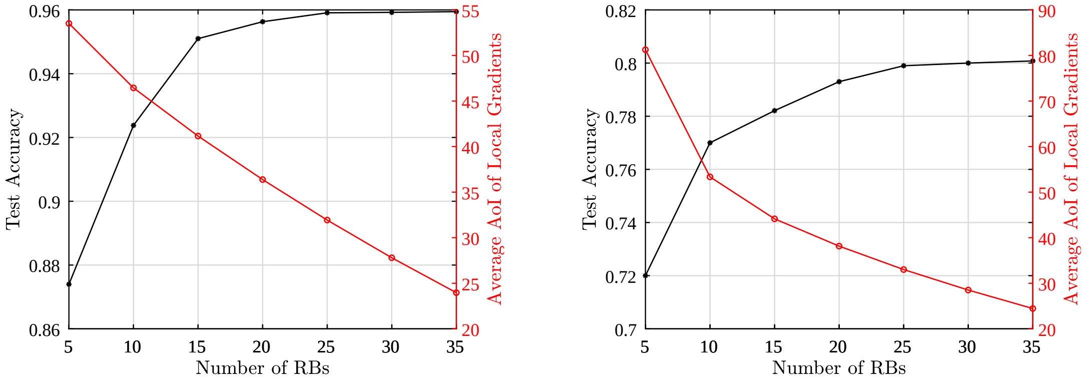
We can see that AMP-FL significantly outperforms Synchronous FL and Regularized FL, i.e., it improves around 4.3% test accuracy compared to these two benchmarks. Although AMP-FL only obtains a slight accuracy improvement to the APFL approach, it converges fast than APFL. AMP-FL only takes 17.5s to achieve 90% accuracy, while

APFL takes 29.5s. That is, AMP-FL provides a 1.7x speed up compared to APFL. Fig. 6.6(b) compares the learning performance of all the FL algorithms under $s = 3$, drawing a similar conclusion to the setting of $s = 2$. Specifically, AMP-FL boosts 3.63% accuracy compared to Synchronous FL and Regularized FL and achieves a 1.9x speed up when the target accuracy is 90% compared to APFL. In addition, for all the FL algorithms, their learning performance under $s = 3$ is better than that under $s = 2$. This is because the high data heterogeneity would introduce higher variance in the global model update and degrade the learning performance.

Fig. 6.7 conducts a similar comparison on the CIFAR-10 dataset. From Fig. 6.7(a) with setting $s = 2$, when the target accuracy is 70% and 75%, the proposed AMP-FL is capable of providing a 1.6x and 1.5x speed up compared to the benchmarks, respectively. In Fig. 6.7(b), we set the data heterogeneity-related parameter to $s = 3$. It is observed that AMP-FL achieves a 1.75x and 1.7x speed up when the target accuracy is 70% and 75%, respectively. Moreover, we can see that the learning process of the proposed AMP-FL is more stable than that of the benchmarks since the shadow band of AMP-FL is slim than the benchmarks. The benefits come from the proposed gradient compensation mechanism and model pruning approach, which prevents the global model from being biased toward devices with high communication and computation capabilities. From the results in Fig. 6.6 and Fig. 6.7, dynamically adjusting the local models to adapt devices' computation and communication capabilities is an efficient approach to mitigate the straggler effects in practical wireless FL systems.

6.5.4 Impact of Wireless Resource on Learning Performance

In this subsection, we evaluate the impacts of the number of RBs on the learning performance of AMP-FL, including test accuracy and Average AoI of devices' gradients. Note that in this section, the results on MNIST and CIFAR-10 are achieved after 50 and 3×10^4 seconds of training, respectively.



(a)

(b)

Figure 6.8: Learning performance of the proposed AMP-FL under different number of RBs: (a) on MNIST dataset, (b) on CIFAR-10 dataset.

In Fig. 6.8, we evaluate the effects of the number of RBs on the test accuracy and average AoI on the MNIST and CIFAR-10 datasets. From the results on MNIST dataset in Fig. 6.8(a), it is observed that the test accuracy of AMP-FL keeps increasing along with the increase in the number of RBs. This is because the increasing number of RBs allows more devices to participate in the learning process in each round. In addition, the average AoI of devices' gradients decreases with the increase in the number of RBs. According to the definition of AoI in (9), the AoI of a model region increases when the corresponding device is not selected or the model region is pruned. Increasing the number of resource blocks would increase the number of selected devices in each round, and thus more model regions are selected in each round. Consequently, the average AoI across devices would be reduced. The results on the CIFAR-10 dataset in Fig. 6.8(b) show a similar conclusion to the results in Fig. 6.8(a), indicating that increasing the number of RBs helps improve test accuracy of reducing the average AoI of devices' gradients. These simulation results further verifies our theoretical analysis results in Remark 6, which suggests minimizing the average AoI of local gradients to enhance the learning performance.

6.6 Summary

In this chapter, we developed a novel AMP-FL framework which dynamically prunes the global model to generate sub-models adapted to devices' communication and computation capabilities. This framework is capable of simultaneously reducing communication and computation overhead for devices to enable efficient FL among heterogeneous devices. To prevent the diverse structures of pruned local models from affecting the training convergence, we proposed a gradient compensation mechanism to compensate for the gradients of pruned model regions by devices' historical gradients. We introduced an AoI metric to characterize the staleness of local gradients and analyzed the convergence bound of AMP-FL. The convergence bound suggests scheduling devices with large AoI and pruning the model regions with small AoI for devices in the per-round learning process. Based on this, we develop an effective device scheduling, model pruning, and RB allocation approach to enhance the learning performance of AMP-FL in wireless networks. Experimental results show that compared to the benchmark FL algorithms, the proposed AMP-FL is capable of achieving 1.9x and 1.6x speed up on MNIST and CIFAR-10 datasets, respectively.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis concentrates on the joint design of wireless networks and learning mechanisms to address different challenges that emerged in wireless FL, aiming to improve the learning performance and enhance the robustness of FL in practical wireless networks. Specifically, six fundamental challenges under different scenarios are investigated, i.e., data heterogeneity, scarce wireless resources, high communication overhead, model heterogeneity, device heterogeneity, and unreliable communications. For these challenges, this thesis proposes four schemes to address them and verifies the effectiveness of the designed schemes in learning performance.

Chapter 3 started by investigating the origin of learning performance degradation caused by partial device participation through convergence analysis, which revealed that the learning convergence declined due to the error between the scheduled devices aggregated gradient and full participation aggregated gradient. In addition, considering the limited wireless bandwidth, we propose to schedule a subset of representative devices and the corresponding pre-device stepsizes to approximate the full participation aggregated gradient while capturing the trade-off between learning performance and latency for FL. Simulation results demonstrated the efficiency of the proposed scheme in terms of convergence speed and learning accuracy.

Chapter 4 developed a novel KFL framework to address the data heterogeneity, high communication overhead, and model heterogeneity in wireless FL. Unlike the conventional FL algorithms that aggregate local models, KFL aggregates light high-level data features, namely knowledge, in the per-round learning process. Compared to the conventional model aggregation-based FL algorithms, KFL possesses three advantages, i.e., allowing devices to be equipped with heterogeneous local models, greatly reducing the communication overhead, and mitigating the data heterogeneity problem. In addition, when deploying KFL in wireless networks with energy-limited devices, we theoretically and experimentally demonstrated that scheduling more devices in the early rounds helps improve learning performance.

Chapter 5 proposed a gradient recycling-based FL framework, i.e., FL-GR, to address the unreliable communication and scarce resources in wireless FL. FL-GR reuse the historical gradients of unscheduled and transmission-failure devices to calibrate the global update and improve the learning performance of FL. On this basis, we theoretically revealed that minimizing the average square of local gradients staleness in the learning process helps improve learning performance. Following that, we developed a joint device scheduling, resource allocation and power control approach to enhance the performance of FL. Extensive simulations verified the effectiveness of the developed approaches in unreliable wireless networks.

Chapter 6 proposed an adaptive model pruning-based FL approach to tackle the device heterogeneity, in which the edge server dynamically generates sub-models by pruning the global model for devices local training to adapt their heterogeneous computation capabilities and time-varying channel conditions. Since the model pruning produces diverse structures of devices submodels and negatively affects the learning performance, we propose compensating for the gradients of pruned model regions by devices historical gradients. Following that, we introduced an AoI metric to characterize the staleness of local gradients and theoretically analyze the convergence behaviour of AMP-FL. Based on the convergence bound, we further jointly optimized the device scheduling, model pruning, and resource block (RB) allocation policies to enhance the learning efficiency of FL.

7.2 Future Directions

In this section, some possible extensions of the current works in this thesis and future research directions are summarized as follows:

7.2.1 Federated Split Learning

In FL, the devices undertake the overall training computations, and the server is only responsible for aggregating the local updates uploaded from devices. However, training and transmitting model updates are prohibitively expensive for resource-constrained devices, especially when training a large ML model. To relieve the computational burden for devices, federated split learning (FSL) [122, 123] was developed to split the ML model into two parts so that most computations are offloaded to the server, and devices only train a small part of the learning model. With the development of machine learning techniques, one generated insight is that scaling up model size can effectively improve model accuracy. Under this scope, FSL is a promising technique to enable collaborative training of large ML models at the wireless networks.

Although FSL is device-friendly, it still suffers from high communication costs due to the back-and-forth transmission of smash data and gradients of every data sample in each epoch. In addition, the existing FSL frameworks mainly focused on reducing the communication cost of homogeneous client-based systems. Most of the practical system limitations, e.g., device heterogeneity and data heterogeneity, have not been well investigated. Thus, it is essential to develop innovative FSL frameworks to address practical challenges and improve learning efficiency.

7.2.2 Federated Multi-Modal Learning

With the advancement of data collection techniques, end users are interested in how different types of data can collaborate to improve our life experiences. However, most existing FL algorithms mainly focused on the single-modal data scenarios, e.g., computer vision, audio, and natural language processing.

In practical systems, devices may have heterogeneous setups of sensors and their local data consists of different combinations of modalities. The FL algorithm utilizing multi-modal data sources remains in its infancy [124, 125]. With the modality incongruity, devices may solve different tasks on different parameter spaces, which escalates the difficulties in federated training. In addition, it would be hard to perform accurate model aggregation across different types of clients. Considering these practical challenges and applications, developing effective Federated multi-modal learning schemes is still an open problem.

7.2.3 Federated Continue Learning

In practical systems, the local datasets at devices may dynamically change over time. For example, a mobile phone user may delete some photos or take some new photos. In addition, devices may join and depart the training without prior notice. These practical situations induced the time-varying local datasets in the federated training process. However, the existing FL research mainly relies on fixed data distribution among devices throughout the entire learning process, which may confront severe learning performance degradation under the time-varying local datasets.

To tackle the time-varying data in FL, integrating continual learning [126] into FL is a promising framework. In fact, continual learning approaches have been widely investigated in the centralized learning setting to avoid the problem of catastrophic forgetting in time-varying data. The existing continual learning methods are mainly from the perspective of regularization [127], experience memory [128], and dynamic architectures [129]. Directly integrating these continual learning techniques into FL would increase the memory and computation burdens for the resource-constrained devices. Thus, it is essential to design novel federated continual learning approaches to enable resource-efficient learning under time-varying data scenarios.

Appendix A

Proof in Chapter 2

A.1 Proof of Theorem 1

According to the L-Lipschitz continuous of loss gradients $\nabla F_k(\boldsymbol{w})$ in Assumption 1, we have

$$F_k(\boldsymbol{w}) \leq F_k(\boldsymbol{v}) - \langle \nabla F_k(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle + \frac{L}{2} \left\| \boldsymbol{w} - \boldsymbol{v} \right\|^2 \tag{A.1}$$

For ease of proof, we define $\bar{\boldsymbol{g}}_t = \frac{1}{K} \sum_{k=1}^{K} \sum_{l=0}^{\tau-1} \nabla F_k(\boldsymbol{w}_{k,t,l})$. Thus,

$$\begin{aligned}
&\mathbb{E}\left[F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}_t)\right] \\
&\leq \mathbb{E}\left[\langle \nabla F(\boldsymbol{w}_t), \boldsymbol{w}_{t+1} - \boldsymbol{w}_t \rangle\right] + \frac{L}{2} \mathbb{E} \left\| \boldsymbol{w}_{t+1} - \boldsymbol{w}_t \right\|^2 \\
&= -\eta \mathbb{E}\left[\langle \nabla F(\boldsymbol{w}_t), \widetilde{\boldsymbol{g}}_t \rangle\right] + \frac{L}{2} \eta^2 \mathbb{E} \|\widetilde{\boldsymbol{g}}_t\|^2 \\
&= -\eta \mathbb{E}\left[\langle \nabla F(\boldsymbol{w}_t), \widetilde{\boldsymbol{g}}_t \rangle\right] + \frac{L}{2} \eta^2 \mathbb{E} \left\| \nabla F(\boldsymbol{w}_t) - \nabla F(\boldsymbol{w}_t) + \widetilde{\boldsymbol{g}}_t \right\|^2 \\
&= -\frac{L}{2} \eta^2 \mathbb{E} \|\nabla F(\boldsymbol{w}_t)\|^2 + (L\eta^2 - \eta) \mathbb{E}\left[\langle \nabla F(\boldsymbol{w}_t), \widetilde{\boldsymbol{g}}_t \rangle\right] + \frac{L}{2} \eta^2 \mathbb{E} \|-\nabla F(\boldsymbol{w}_t) + \widetilde{\boldsymbol{g}}_t\|^2 \\
&\overset{(a)}{\leq} -\frac{L}{2} \eta^2 \mathbb{E} \|\nabla F(\boldsymbol{w}_t)\|^2 + \frac{L}{2} \eta^2 \mathbb{E} \|-\nabla F(\boldsymbol{w}_t) + \boldsymbol{g}_t - \boldsymbol{g}_t + \widetilde{\boldsymbol{g}}_t\|^2 \\
&\overset{(b)}{\leq} -\frac{L}{2} \eta^2 \mathbb{E} \|\nabla F(\boldsymbol{w}_t)\|^2 + L\eta^2 \mathbb{E} \|-\boldsymbol{g}_t + \widetilde{\boldsymbol{g}}_t\|^2 + L\eta^2 \mathbb{E} \|-\nabla F(\boldsymbol{w}_t) + \boldsymbol{g}_t\|^2 \\
&= -\frac{L}{2} \eta^2 \mathbb{E} \|\nabla F(\boldsymbol{w}_t)\|^2 + L\eta^2 \mathbb{E} \|-\boldsymbol{g}_t + \widetilde{\boldsymbol{g}}_t\|^2 + L\eta^2 \mathbb{E} \|-\nabla F(\boldsymbol{w}_t) + \bar{\boldsymbol{g}}_t - \bar{\boldsymbol{g}}_t + \boldsymbol{g}_t\|^2 \\
&\overset{(c)}{=} -\frac{L}{2} \eta^2 \mathbb{E} \|\nabla F(\boldsymbol{w}_t)\|^2 + L\eta^2 \mathbb{E} \|-\boldsymbol{g}_t + \widetilde{\boldsymbol{g}}_t\|^2
\end{aligned}$$

$$+ L\eta^2\mathbb{E}\|-\nabla F(\boldsymbol{w}_t) + \bar{\boldsymbol{g}}_t\|^2 + L\eta^2\mathbb{E}\|-\bar{\boldsymbol{g}}_t + \boldsymbol{g}_t\|^2, \tag{A.2}$$

where (a) derived by Cauchy-Schwarz inequality and $\eta \leq \frac{1}{L}$, (b) follows the triangle-inequality, (c) is due to the unbiased stochastic gradient in Assumption 2.

Below we first bound $\mathbb{E}\,\|-\nabla F(\boldsymbol{w}_t) + \bar{\boldsymbol{g}}_t\|^2$.

$$\begin{aligned}
&\mathbb{E}\,\|-\nabla F(\boldsymbol{w}_t) + \bar{\boldsymbol{g}}_t\|^2 \\
&= \mathbb{E}\left\|\frac{1}{K}\sum_{k=1}^{K}\sum_{l=1}^{\tau-1}\nabla F(\boldsymbol{w}_{k,t,l})\right\|^2 \\
&\overset{(a)}{\leq} \frac{1}{K}\sum_{k=1}^{K}(\tau-1)\sum_{l=1}^{\tau-1}\mathbb{E}\|\nabla F(\boldsymbol{w}_{k,t,l})\|^2 \\
&= \frac{1}{K}\sum_{k=1}^{K}(\tau-1)\sum_{l=1}^{\tau-1}\mathbb{E}\left\|\nabla F(\boldsymbol{w}_{k,t,l}) - \nabla F(\boldsymbol{w}_{k,t,l},\mathcal{B}_{k,t,l}) + \nabla F(\boldsymbol{w}_{k,t,l},\mathcal{B}_{k,t,l})\right\|^2 \\
&\overset{(b)}{=} \frac{1}{K}\sum_{k=1}^{K}(\tau-1)\sum_{l=1}^{\tau-1}\left\{\mathbb{E}\|\nabla F(\boldsymbol{w}_{k,t,l}) - \nabla F(\boldsymbol{w}_{k,t,l},\mathcal{B}_{k,t,l})\|^2 + \mathbb{E}\|\nabla F(\boldsymbol{w}_{k,t,l},\mathcal{B}_{k,t,l})\|^2\right\} \\
&\overset{(c)}{\leq} (\tau-1)^2(G^2+\chi^2)
\end{aligned} \tag{A.3}$$

where (a) follows Jensen's inequality, (b) is due to the unbiased gradient in Assumption 2, (c) follows the Assumption 2 and Assumption 3. In the following, we bound $\mathbb{E}\|-\bar{\boldsymbol{g}}_t + \boldsymbol{g}_t\|^2$ as follows:

$$\begin{aligned}
\mathbb{E}\|-\bar{\boldsymbol{g}}_t + \boldsymbol{g}_t\|^2 &= \mathbb{E}\left\|\frac{1}{K}\sum_{k=1}^{K}\sum_{l=0}^{\tau-1}(\nabla F(\boldsymbol{w}_{k,t,l},\mathcal{B}_{k,t,l}) - \nabla F(\boldsymbol{w}_{t,l}))\right\|^2 \\
&\overset{(a)}{\leq} \frac{1}{K}\sum_{k=1}^{K}\tau\sum_{l=0}^{\tau-1}\mathbb{E}\|(\nabla F\left(\boldsymbol{w}_{k,t,l},\mathcal{B}_{k,t,l}\right) - \nabla F(\boldsymbol{w}_{t,l}))\|^2 \\
&\overset{(b)}{\leq} \tau^2 G^2
\end{aligned} \tag{A.4}$$

where (a) follows Jensen's inequality, (b) is due to Assumption 2. Substituting (A.3) and (A.4) into (A.2), the proof is completed.

A.1.1 Proof of Corollary 1

By using the L-smooth of loss functions, we have

$$\left\|\nabla F(\boldsymbol{w}_t)\right\|^2 \leq 2L\left(F(\boldsymbol{w}_t) - F(\boldsymbol{w}^*)\right). \tag{A.5}$$

By substracting $F(\boldsymbol{w}^*)$ for both $F(\boldsymbol{w}_{t+1})$ and $F(\boldsymbol{w}_t)$ in the one-round convergence bound in (3.12),

$$\begin{aligned}\mathbb{E}(F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}^*)) \leq \mathbb{E}(F(\boldsymbol{w}_t) - F(\boldsymbol{w}^*)) - \frac{L}{2}\eta^2\mathbb{E}\left\|\nabla F(\boldsymbol{w}_t)\right\|^2 \\ + L\eta^2\left(2\tau^2 - 2\tau + 1\right)G^2 + L\eta^2(\tau-1)^2\chi^2 + L\eta^2\left\|-\boldsymbol{g}_t + \widetilde{\boldsymbol{g}}_t\right\|^2.\end{aligned} \tag{A.6}$$

Substituting (A.5) into (A.6), we have

$$\begin{aligned}F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}^*) \leq \left(1 - L^2\eta^2\right)\left(F(\boldsymbol{w}_t) - F(\boldsymbol{w}^*)\right) \\ + L\eta^2\left(2\tau^2 - 2\tau + 1\right)G^2 + L\eta^2(\tau-1)^2\varepsilon^2 + L\eta^2\left\|-\boldsymbol{g}_t + \widetilde{\boldsymbol{g}}_t\right\|^2.\end{aligned} \tag{A.7}$$

Telescoping the above equation, the convergence bound after T rounds can be derived as Corollary 1. The proof is completed.

A.1.2 Proof of Lemma 3

From the definition of $T_{k,t}^{\mathrm{C}}$ in (3.7), it is straightforward to see that $T_{k,t}^{\mathrm{C}}$ is a monotonically decreasing function with respect to $\theta_{k,t}$. For any device that finished the local gradient computing process earlier than other devices, we can reallocate some of its bandwidth to other slower devices. As a result, the one-round latency determined by the slowest device can be reduced. The bandwidth reallocation process will be performed until all devices simultaneously finish the local gradient computing and uploading. Consequently, the optimal solution of (3.19) is achieved when the entire bandwidth is allocated to all scheduled devices to have the same finishing time. Thus, the optimal bandwidth allocation policy satisfies

$$\begin{cases} T_{k,t}^{\mathrm{L}} + T_{k,t}^{\mathrm{C}} = \mathcal{T}_t^*(\boldsymbol{S}_t), \forall k \in \boldsymbol{S}_t \\ \sum_{k\in\boldsymbol{S}_t}\theta_k = 1, \end{cases} \tag{A.8}$$

where $\mathcal{T}_t^*(\boldsymbol{S}_t)$ is the optimal latency in round t. By solving (A.8), the proof is completed.

A.1.3 Proof of Lemma 4

For ease of presentation, we first denote the minimal gradient uploading latency for device k $(k \in \mathcal{K})$ as $T_{k,t}^{\mathrm{C,min}} = \frac{Qq}{B \log\left(1+\frac{p_k h_{k,t}}{\sigma^2}\right)}$, which is derived when device k occupies the entire bandwidth to upload its gradient. Below we first prove that the optimal latency function $\mathcal{T}_t^*(\boldsymbol{S})$ is a monotonically increasing function with the device set $\boldsymbol{S}$. Based on Lemma 3, for device set $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2 \subseteq \mathcal{K}$, we have

$$\sum_{k \in \boldsymbol{S}_1} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_1) - T_{k,t}^{\mathrm{L}}} = \sum_{k \in \boldsymbol{S}_2} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_2) - T_{k,t}^{\mathrm{L}}} = 1, \tag{A.9}$$

which equivalent to

$$\sum_{k \in \boldsymbol{S}_1} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_1) - T_{k,t}^{\mathrm{L}}} = \sum_{k \in \boldsymbol{S}_1} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_2) - T_{k,t}^{\mathrm{L}}} + \sum_{k \in \boldsymbol{S}_2 \setminus \boldsymbol{S}_1} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_2) - T_{k,t}^{\mathrm{L}}}. \tag{A.10}$$

By rearranging the above equation, we have

$$\sum_{k \in \boldsymbol{S}_1} \left(\frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_1) - T_{k,t}^{\mathrm{L}}} - \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_2) - T_{k,t}^{\mathrm{L}}} \right) = \sum_{k \in \boldsymbol{S}_2 \setminus \boldsymbol{S}_1} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_2) - T_{k,t}^{\mathrm{L}}} > 0 \tag{A.11}$$

Thus, we have $\mathcal{T}_t^*(\boldsymbol{S}_1) \leq \mathcal{T}_t^*(\boldsymbol{S}_2)$. That is, $\mathcal{T}_t^*(\boldsymbol{S})$ is a monotonically increasing function with the device set $\boldsymbol{S}$. Similarly, for device $h \in \mathcal{K} \setminus \boldsymbol{S}_2$, we have

$$\sum_{k \in \boldsymbol{S}_1} \left(\frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*\left(\boldsymbol{S}_1 + h\right) - T_{k,t}^{\mathrm{L}}} - \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_1) - T_{k,t}^{\mathrm{L}}} \right) + \frac{T_{h,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*\left(\boldsymbol{S}_1 + h\right) - T_{h,t}^{\mathrm{L}}} = 0 \tag{A.12}$$

By rearranging the above equation, we have

$$\mathcal{T}_t^*\left(\boldsymbol{S}_1 + h\right) - \mathcal{T}_t^*(\boldsymbol{S}_1) = \frac{T_{h,t}^{\mathrm{C,min}}}{1 + \sum_{k \in \boldsymbol{S}_1} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_1) - T_{k,t}^{\mathrm{L}}} \frac{T_{k,t}^{\mathrm{L}} - T_{h,t}^{\mathrm{L}}}{\mathcal{T}_t^*(\boldsymbol{S}_1 + h) - T_{k,t}^{\mathrm{L}}}} \tag{A.13}$$

Since $\mathcal{T}_t^*\left(\boldsymbol{S}_1 + h\right) < \mathcal{T}_t^*\left(\boldsymbol{S}_2 + h\right)$, based on (A.9), we have $\mathcal{T}_t^*\left(\boldsymbol{S}_1 + h\right) - \mathcal{T}_t^*(\boldsymbol{S}_1) \leq \mathcal{T}_t^*\left(\boldsymbol{S}_2 + h\right) - \mathcal{T}_t^*(\boldsymbol{S}_2)$, the proof is completed.

A.1.4 Proof of Lemma 5

For ease of presentation, we define two device sets, $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, such that $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2 \subseteq \mathcal{K}$, and a device $h \in \mathcal{K} \backslash \boldsymbol{S}_2$. Based on the definition of $H(\boldsymbol{S}_t)$, we have $\mathcal{H}(\boldsymbol{S}_1) \geq \mathcal{H}(\boldsymbol{S}_2)$ and $\mathcal{H}(\boldsymbol{S}_1 \cup \{h\}) \geq \mathcal{H}(\boldsymbol{S}_2 \cup \{h\})$. Moreover, we have

$$\begin{aligned}
&\mathcal{H}(\boldsymbol{S}_1 \cup \{h\}) - \mathcal{H}(\boldsymbol{S}_1) \\
&\quad = \sum_{k=1}^{K} \min_{e \in \boldsymbol{S}_1 \cup \{h\}} \|\nabla F_k(\boldsymbol{w}_t) - \nabla F_e(\boldsymbol{w}_t)\| - \sum_{k=1}^{K} \min_{e \in \boldsymbol{S}_1} \|\nabla F_k(\boldsymbol{w}_t) - \nabla F_e(\boldsymbol{w}_t)\| \\
&\quad = \sum_{k=1}^{K} \min \left(0, \|\nabla F_k(\boldsymbol{w}_t) - \nabla F_n(\boldsymbol{w}_t)\| - \min_{h \in \boldsymbol{S}_1} \|\nabla F_k(\boldsymbol{w}_t) - \nabla F_h(\boldsymbol{w}_t)\| \right). \qquad \text{(A.14)}
\end{aligned}$$

Since $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2$, we have $\min\limits_{h \in \boldsymbol{S}_1} \|\nabla F_k(\boldsymbol{w}_t) - \nabla F_h(\boldsymbol{w}_t)\| \geq \min\limits_{h \in \boldsymbol{S}_2} \|\nabla F_k(\boldsymbol{w}_t) - \nabla F_h(\boldsymbol{w}_t)\|$. Thus, $\mathcal{H}(\boldsymbol{S}_1 \cup \{h\}) - \mathcal{H}(\boldsymbol{S}_1) \leq \mathcal{H}(\boldsymbol{S}_2 \cup \{h\}) - \mathcal{H}(\boldsymbol{S}_2)$, the proof is completed.

Appendix B

Proof in Chapter 3

B.1 Proof of Lemma 6

Using L_u smooth of $F_k(\cdot, \boldsymbol{v}_k)$ and L_v-smooth of $F(\boldsymbol{u}_k, \cdot)$, we have

$$F_k(\boldsymbol{u}_k', \boldsymbol{v}_k') - F_k\left(\boldsymbol{u}_k, \boldsymbol{v}_k'\right) \leq \left\langle \nabla_{\boldsymbol{u}} F_k\left(\boldsymbol{u}_k, \boldsymbol{v}_k'\right), \boldsymbol{u}_k' - \boldsymbol{u}_k \right\rangle + \frac{L_u}{2} \left\| \boldsymbol{u}_k' - \boldsymbol{u}_k \right\|^2, \tag{B.1}$$

and

$$F_k\left(\boldsymbol{u}_k, \boldsymbol{v}_k'\right) - F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) \leq \left\langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{v}_k' - \boldsymbol{v}_k \right\rangle + \frac{L_v}{2} \left\| \boldsymbol{v}_k' - \boldsymbol{v}_k \right\|^2. \tag{B.2}$$

Summarizing (B.1) and (B.2), we have

$$\begin{aligned} F_k(\boldsymbol{u}_k', \boldsymbol{v}_k') - F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) \leq &\left\langle \nabla_{\boldsymbol{u}} F_k\left(\boldsymbol{u}_k, \boldsymbol{v}_k'\right), \boldsymbol{u}_k' - \boldsymbol{u}_k \right\rangle + \frac{L_u}{2} \left\| \boldsymbol{u}_k' - \boldsymbol{u}_k \right\|^2 \\ &+ \left\langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{v}_k' - \boldsymbol{v}_k \right\rangle + \frac{L_v}{2} \left\| \boldsymbol{v}_k' - \boldsymbol{v}_k \right\|^2. \end{aligned} \tag{B.3}$$

We now focus on bounding $\langle \nabla_{\boldsymbol{u}} F_k\left(\boldsymbol{u}_k, \boldsymbol{v}_k'\right), \boldsymbol{u}_k' - \boldsymbol{u}_k \rangle$ as follows:

$$\begin{aligned} &\left\langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k'), \boldsymbol{u}_k' - \boldsymbol{u}_k \right\rangle \\ &\overset{(a)}{=} \left\langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{u}_k' - \boldsymbol{u}_k \right\rangle + \left\langle \nabla_{\boldsymbol{u}} F_k\left(\boldsymbol{u}_k, \boldsymbol{v}_k'\right) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{u}_k' - \boldsymbol{u}_k \right\rangle \\ &\overset{(b)}{\leq} \left\langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{u}_k' - \boldsymbol{u}_k \right\rangle + \left\| \nabla_{\boldsymbol{u}} F_k\left(\boldsymbol{u}_k, \boldsymbol{v}_k'\right) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) \right\| \left\| \boldsymbol{u}_k' - \boldsymbol{u}_k \right\| \\ &\overset{(c)}{\leq} \left\langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{u}_k' - \boldsymbol{u}_k \right\rangle + L_{uv} \left\| \boldsymbol{v}_k' - \boldsymbol{v}_k \right\| \left\| \boldsymbol{u}_k' - \boldsymbol{u}_k \right\| \end{aligned}$$

$$\overset{(c)}{\leq} \big\langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{u}_k' - \boldsymbol{u}_k \big\rangle + \frac{1}{2}\chi L_v \big\|\boldsymbol{v}_k' - \boldsymbol{v}_k\big\|^2 + \frac{1}{2}\chi L_u \big\|\boldsymbol{u}_k' - \boldsymbol{u}_k\big\|^2, \tag{B.4}$$

where (a) is derived by adding and substracting $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ into $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k')$, (b) follows the Cauchy-Schwarz inequality, (c) comes from Assumption 4, (d) is due to the definition of χ. Substituting (B.4) into (B.3), the proof completes.

B.2 Proof of Lemma 7

According to Lemma 6, we have

$$\begin{aligned} & F_k(\boldsymbol{u}_{k,t+1}, \boldsymbol{v}_{k,t+1}) - F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) \\ & \qquad \leq \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \boldsymbol{u}_{k,t+1} - \boldsymbol{u}_{k,t} \rangle + \frac{1+\chi}{2} L_u \|\boldsymbol{u}_{k,t+1} - \boldsymbol{u}_{k,t}\|^2 \\ & \qquad\qquad + \langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \boldsymbol{v}_{k,t+1} - \boldsymbol{v}_{k,t} \rangle + \frac{1+\chi}{2} L_v \|\boldsymbol{v}_{k,t+1} - \boldsymbol{v}_{k,t}\|^2. \end{aligned} \tag{B.5}$$

Below we focus on bounding the four terms on the RHS of (B.5). Firstly, we bound $\langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \boldsymbol{u}_{k,t+1} - \boldsymbol{u}_{k,t} \rangle$ as follows:

$$\begin{aligned} & \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \boldsymbol{u}_{k,t+1} - \boldsymbol{u}_{k,t} \rangle \\ & = -\eta_u \sum_{l=0}^{\tau-1} \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,l}) \rangle \\ & \overset{(a)}{\leq} -\frac{\eta_u \tau}{2} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \\ & + \frac{\eta_u}{2} \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,l})\|^2 \\ & \overset{(b)}{\leq} -\frac{\eta_u \tau}{2} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 + \eta_u \lambda^2 \sum_{l=0}^{\tau-1} \|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2 \\ & + \eta_u \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2, \end{aligned} \tag{B.6}$$

where (a) is derived by adding and substracting $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})$ into $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l})$ and using the triangle inequality, (b) follows the triangle inequality. For the second term on the RHS of (B.5), we bound $\|\boldsymbol{u}_{k,t+1} - \boldsymbol{u}_{k,t}\|^2$ as

$$\|\boldsymbol{u}_{k,t+1} - \boldsymbol{u}_{k,t}\|^2 = \eta_u^2 \left\| \sum_{l=0}^{\tau-1} (\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,l})) \right\|^2$$

$$
\begin{aligned}
&\overset{(a)}{\leq} \eta_u^2 \tau \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,l})\|^2 \\
&\overset{(b)}{\leq} 2\eta_u^2 \tau \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l})\|^2 + 2\eta_u^2 \tau \sum_{l=0}^{\tau-1} \|\lambda \nabla L_k(\boldsymbol{u}_{k,t,l})\|^2 \\
&\overset{(c)}{\leq} 4\eta_u^2 \tau^2 \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 + 2\eta_u^2 \tau \lambda^2 \sum_{l=0}^{\tau-1} \|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2 \\
&+ 4\eta_u^2 \tau \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2, \qquad \text{(B.7)}
\end{aligned}
$$

where (a) is due to Jensen's inequality, (b) follows the triangle inequality, (c) is derived by adding and substracting $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})$ into $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l})$. We now focus on bounding $\sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2$ which appears in both (B.6) and (B.7) as

$$
\begin{aligned}
&\sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \\
&\overset{(a)}{\leq} 2\sum_{l=0}^{\tau-1} \Big(\|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t,l})\|^2 \\
&+ \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \Big) \\
&\overset{(b)}{\leq} 2\sum_{l=0}^{\tau-1} L_u^2 \|\boldsymbol{u}_{k,t,l} - \boldsymbol{u}_{k,t}\|^2 + 2\sum_{l=0}^{\tau-1} \chi^2 L_u L_v \|\boldsymbol{v}_{k,t,l} - \boldsymbol{v}_{k,t}\|^2. \qquad \text{(B.8)}
\end{aligned}
$$

where (a) derived by adding and substracting $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t,l})$ and using the triangle inequality, (b) follows Assumption 4 and the definition of χ.

For the last two terms on the RHS of (B.5), we have

$$
\begin{aligned}
&\langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \boldsymbol{v}_{k,t+1} - \boldsymbol{v}_{k,t} \rangle + \frac{1+\chi}{2} L_v \|\boldsymbol{v}_{k,t+1} - \boldsymbol{v}_{k,t}\|^2 \\
&= -\eta_v \sum_{l=0}^{\tau-1} \langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) \rangle + \frac{1+\chi}{2} L_v \eta_v^2 \left\| \sum_{l=0}^{\tau-1} \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) \right\|^2 \\
&\overset{(a)}{\leq} -\eta_v \sum_{l=0}^{\tau-1} \langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) \rangle + \frac{1+\chi}{2} L_v \eta_v^2 \tau \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l})\|^2 \\
&\overset{(b)}{\leq} ((1+\chi) L_v \eta_v^2 \tau^2 - \frac{1}{2} \eta_v \tau) \|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \\
&+ ((1+\chi) L_v \eta_v^2 \tau + \frac{1}{2} \eta_v) \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2, \qquad \text{(B.9)}
\end{aligned}
$$

where (a) is due to Jensen's inequality, (b) is derived by adding and substracting $\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})$ into $\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l})$ and using the triangle inequality. In (B.9), we bound $\sum_{l=0}^{\tau-1}\|-\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) + \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l})\|^2$ as

$$\begin{aligned}
&\sum_{l=0}^{\tau-1}\|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \\
&\leq 2\sum_{l=0}^{\tau-1}\|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t,l})\|^2 \\
&+ 2\sum_{l=0}^{\tau-1}\|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \\
&\leq 2\sum_{l=0}^{\tau-1}\chi^2 L_u L_v\|\boldsymbol{u}_{k,t,l} - \boldsymbol{u}_{k,t}\|^2 + 2\sum\nolimits_{l=0}^{\tau-1} L_v^2\|\boldsymbol{v}_{k,t,l} - \boldsymbol{v}_{k,t}\|^2. \qquad \text{(B.10)}
\end{aligned}$$

Substituting (B.6), (B.7), (B.8), (B.9), and (B.10) into (B.5), and the learning rates satisfy $\eta_u \leq \frac{1}{4\tau(1+\chi)L_u}$ and $\eta_v \leq \frac{1}{2\tau(1+\chi)L_v}$, we have

$$\begin{aligned}
F_k(\boldsymbol{u}_{k,t+1}, \boldsymbol{v}_{k,t+1}) - F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) &\leq \left(2(1+\chi)L_u\eta_u^2\tau^2 - \frac{1}{2}\eta_u\tau\right)\|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \\
+\left((1+\chi)L_v\eta_v^2\tau^2 - \frac{1}{2}\eta_v\tau\right)\|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 &+ (3\eta_u L_u^2 + 2\eta_v\chi^2 L_u L_v)\sum_{l=0}^{\tau-1}\|\boldsymbol{u}_{k,t,l} - \boldsymbol{u}_{k,t}\|^2 \\
+ (3\eta_u\chi^2 L_u L_v + 2\eta_v L_v^2)\sum_{l=0}^{\tau-1}\|\boldsymbol{v}_{k,t,l} - \boldsymbol{v}_{k,t}\|^2 &+ \frac{5}{4}\eta_u\lambda^2\sum_{l=0}^{\tau-1}\|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2. \qquad \text{(B.11)}
\end{aligned}$$

Below we focus on bounding two terms in (B.11), i.e., $\sum\limits_{l=0}^{\tau-1}\|\boldsymbol{v}_{k,t,l} - \boldsymbol{v}_{k,t}\|^2$ and $\sum\limits_{l=0}^{\tau-1}\|\boldsymbol{u}_{k,t,l} - \boldsymbol{u}_{k,t}\|^2$. Firstly, for $\sum_{l=0}^{\tau-1}\|\boldsymbol{v}_{k,t,l} - \boldsymbol{v}_{k,t}\|^2$, we have

$$\begin{aligned}
\sum_{l=0}^{\tau-1}\|\boldsymbol{v}_{k,t,l} - \boldsymbol{v}_{k,t}\|^2 &= \sum_{l=0}^{\tau-1}\eta_v^2\left\|\sum_{n=0}^{l-1}\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,n}, \boldsymbol{v}_{k,t,n})\right\|^2 \\
&\overset{(a)}{\leq} \sum_{l=0}^{\tau-1}\eta_v^2 l\sum_{n=0}^{l-1}\|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,n}, \boldsymbol{v}_{k,t,n})\|^2 \\
&\overset{(b)}{\leq} \eta_v^2 G_2^2\frac{\tau(\tau+1)(2\tau+1)}{6}, \qquad \text{(B.12)}
\end{aligned}$$

where (a) comes from the Jensen's inequality, (b) follows the bounded gradient assumption in Assumption 5. For $\sum_{l=0}^{\tau-1}\|\boldsymbol{u}_{k,t,l} - \boldsymbol{u}_{k,t}\|^2$, we have

$$\sum_{l=0}^{\tau-1}\|\boldsymbol{u}_{k,t,l} - \boldsymbol{u}_{k,t}\|^2$$

$$
\begin{aligned}
&= \sum_{l=0}^{\tau-1} \eta_u^2 \left\| \sum_{n=0}^{l-1} \left(\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,n}, \boldsymbol{v}_{k,t,n}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,n})\right) \right\|^2 \\
&\overset{(a)}{\leq} \sum_{l=0}^{\tau-1} \eta_u^2 l \sum_{n=0}^{l-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,n}, \boldsymbol{v}_{k,t,n}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,n})\|^2 \\
&\overset{(b)}{\leq} \sum_{l=0}^{\tau-1} \eta_u^2 l \sum_{n=0}^{l-1} 2\|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,n}, \boldsymbol{v}_{k,t,n})\|^2 + \sum_{l=0}^{\tau-1} \eta_u^2 l \sum\nolimits_{n=0}^{l-1} 2 \left\|\lambda \nabla L_k(\boldsymbol{u}_{k,t,n})\right\|^2 \\
&\overset{(c)}{\leq} \frac{\tau(\tau+1)(2\tau+1)}{3} \eta_u^2 G_1^2 + 2\eta_u^2 \lambda^2 \sum_{l=0}^{\tau-1} l \sum_{n=0}^{l-1} \|\nabla L_k(\boldsymbol{u}_{k,t,n})\|^2, \qquad \text{(B.13)}
\end{aligned}
$$

where (a) is due to the Jensen's inequality, (b) follows the triangle inequlity, (c) is due to Assumption 5. Substituting (B.12) and (B.13) into (B.11), the proof is completed.

B.3 Proof of Theorem 2

By substituting (4.18) into (4.2), we have the one-round convergence bounded of the global loss as follows:

$$
\begin{aligned}
&F(\boldsymbol{W}_{t+1}) - F(\boldsymbol{W}_t) \leq \sum_{k=1}^{K} \frac{D_k}{D} (2(1+\chi) L_u \eta_u^2 \tau^2 - \frac{1}{2}\eta_u \tau) \left\|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\right\|^2 \\
&+ \sum_{k=1}^{K} \frac{D_k}{D} ((1+\chi) L_v \eta_v^2 \tau^2 - \frac{1}{2}\eta_v \tau) \left\|\nabla_v F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\right\|^2 + \frac{5}{4}\eta_u \lambda^2 \sum_{k=1}^{K} \frac{D_k}{D} \sum_{l=0}^{\tau-1} \|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2 \\
&+ A_1 + 2\eta_u^2 \lambda^2 (3\eta_u L_u^2 + 2\eta_v \chi^2 L_u L_v) \sum_{k=1}^{K} \frac{D_k}{D} \sum_{l=0}^{\tau-1} (\tau - l) \left\|\nabla L_k(\boldsymbol{u}_{k,t,l})\right\|^2, \qquad \text{(B.14)}
\end{aligned}
$$

Below we bound $\|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2$. For ease of proof, we introduce an auxiliary variable $\bar{\boldsymbol{\Omega}}_{c,t} = \frac{\sum_{k\in\mathcal{K}} D_{k,c} \boldsymbol{\Omega}_{k,c,t}}{\sum_{k\in\mathcal{K}} D_{k,c}}$, which aggregates all devices's knowledge about class c ($\forall c \in \mathcal{C}$).

$$
\begin{aligned}
\|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2 &= \left\| \frac{1}{D_k} \sum_{c=1}^{C} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \|h_k(\boldsymbol{u}_{k,t,l}; \boldsymbol{x}) - \boldsymbol{\Omega}_{c,t}\| \, \nabla h_k(\boldsymbol{u}_{k,t,l}; \boldsymbol{x}) \right\|^2 \\
&\overset{(a)}{\leq} \frac{1}{D_k^2} C \sum_{c=1}^{C} D_{k,c} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \|h_k(\boldsymbol{u}_{k,t,l}; x) - \boldsymbol{\Omega}_{c,t}\|^2 \|\nabla h_k(\boldsymbol{u}_{k,t,l}; \boldsymbol{x})\|^2 \\
&\overset{(b)}{\leq} \frac{1}{D_k^2} C \sum_{c=1}^{C} D_{k,c} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \|h_k(\boldsymbol{u}_{k,t,l}; \boldsymbol{x}) - \boldsymbol{\Omega}_{c,t}\|^2 \, \vartheta^2
\end{aligned}
$$

$$\overset{(c)}{\leq} 2\frac{1}{D_k^2}\vartheta^2 C \sum_{c=1}^{C} D_{k,c} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \left\| h_k(\boldsymbol{u}_{k,t,l};\boldsymbol{x}) - \bar{\boldsymbol{\Omega}}_{c,t} \right\|^2 + 2\frac{1}{D_k^2}\vartheta^2 C \sum_{c=1}^{C} D_{k,c}^2 \left\| \bar{\boldsymbol{\Omega}}_{c,t} - \boldsymbol{\Omega}_{c,t} \right\|^2, \tag{B.15}$$

where (a) follows Jensen's inequality, (b) is due to Assumption 6, (c) derived by adding and substracting $\bar{\boldsymbol{\Omega}}_{c,t}$ into $\boldsymbol{\Omega}_{c,t}$ and using the triangle inequality.

Below we focus on bounding the two terms on the RHS of (B.15), where the first term is bounded as

$$\begin{aligned} & 2\frac{1}{D_k^2}\vartheta^2 C \sum_{c=1}^{C} D_{k,c} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \left\| h_k(\boldsymbol{u}_{k,t,l};\boldsymbol{x}) - \bar{\boldsymbol{\Omega}}_{c,t} \right\|^2 \\ & = 2\frac{1}{D_k^2}\vartheta^2 C \sum_{c=1}^{C} D_{k,c} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \left\| \frac{1}{D_c} \sum_{h=1}^{K} \sum_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{n,c}} \left(h_k(\boldsymbol{u}_{k,t,l};\boldsymbol{x}) - h_n(\boldsymbol{u}_{h,t};\boldsymbol{x}_1) \right) \right\|^2 \\ & \leq 8\vartheta^2\varsigma^2, \end{aligned} \tag{B.16}$$

where the inequality is due to Jensen's inequality and Assumption 6. For the second term on the RHS of (B.15), we have

$$\begin{aligned} \left\| \bar{\boldsymbol{\Omega}}_{c,t} - \boldsymbol{\Omega}_{c,t} \right\|^2 &= \left\| \frac{\sum\limits_{k=1}^{K} \sum\limits_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{k,c}} h_k(\boldsymbol{u}_{k,t};\boldsymbol{x}_1)}{D_c} - \frac{\sum\limits_{k=1}^{K} \alpha_{k,t-1} \sum\limits_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{k,c}} h_k(\boldsymbol{u}_{k,t};\boldsymbol{x}_1)}{\sum\limits_{k=1}^{K} \alpha_{k,t-1} D_{k,c}} \right\|^2 \\ &= \left\| \frac{\sum\limits_{k=1}^{K}(1-\alpha_{k,t-1}) \sum\limits_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{k,c}} h_k(\boldsymbol{u}_{k,t};\boldsymbol{x}_1)}{D_c} - \frac{(D_c - \sum\limits_{k=1}^{K} \alpha_{k,t-1} D_{k,c}) \sum\limits_{k=1}^{K} \alpha_{k,t-1} \sum\limits_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{k,c}} h_k(\boldsymbol{u}_{k,t};\boldsymbol{x}_1)}{D_c \sum\limits_{k=1}^{K} \alpha_{k,t-1} D_{k,c}} \right\|^2 \\ &\leq \left(\frac{\sum\limits_{k=1}^{K}(1-\alpha_{k,t-1}) \sum\limits_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{k,c}} \| h_k(\boldsymbol{u}_{k,t};\boldsymbol{x}_1) \|}{D_c} + \frac{(D_c - \sum\limits_{k=1}^{K} \alpha_{k,t-1} D_{k,c}) \sum\limits_{k=1}^{K} \alpha_{k,t-1} \sum\limits_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{k,c}} \| h_k(\boldsymbol{u}_{k,t};\boldsymbol{x}_1) \|}{D_c \sum_{k=1}^{K} \alpha_{k,t-1} \mathcal{D}_{k,c}} \right)^2 \\ &\overset{(a)}{\leq} 4\varsigma^2 \left(\frac{D_c - \sum_{k=1}^{K} \alpha_{k,t-1} D_{k,c}}{D_c} \right)^2, \end{aligned} \tag{B.17}$$

where (a) is due to Assumption 6. Substituting (B.15), (B.16), and (B.17) into (B.14), then substracting $F(\boldsymbol{W}^*)$ into both $F(\boldsymbol{W}_{t+1})$ and $F(\boldsymbol{W}_t)$, we have

$$F(\boldsymbol{W}_{t+1}) - F(\boldsymbol{W}^*)$$

$$
\begin{aligned}
&\le F(\boldsymbol{W}_t) - F(\boldsymbol{W}^*) + \left(2(1+\chi)L_u\eta_u^2\tau^2 - \frac{1}{2}\eta_u\tau\right)\sum_{k=1}^{K}\frac{D_k}{D}\|\nabla_{\boldsymbol{u}}F_k(\boldsymbol{u}_{k,t},\boldsymbol{v}_{k,t})\|^2 \\
&+ \left((1+\chi)L_v\eta_v^2\tau^2 - \frac{1}{2}\eta_v\tau\right)\sum_{k=1}^{K}\frac{D_k}{D}\|\nabla_{\boldsymbol{v}}F_k(\boldsymbol{u}_{k,t},\boldsymbol{v}_{k,t})\|^2 \\
&+ A_1 + A_2 + A_2\sum_{k=1}^{K}\frac{1}{D}\frac{1}{D_k}C\sum_{c=1}^{C}D_{k,c}^2\left(\frac{D_c - \sum\limits_{k=1}^{K}\alpha_{k,t-1}D_{k,c}}{D_c}\right)^2,
\end{aligned}
\tag{B.18}
$$

where $A_2 = 10\eta_u\lambda^2\tau\vartheta^2\varsigma^2 + 8\eta_u^2\lambda^2\vartheta^2\varsigma^2\left(3\eta_u L_u^2 + 2\eta_v\chi^2 L_u L_v\right)\tau(\tau+1)$.

By using the L-smooth of loss functions, we have

$$
\|\nabla_{\boldsymbol{u}}F_k(\boldsymbol{u}_{k,t},\boldsymbol{v}_{k,t})\|^2 \le 2L_u\left(F_k(\boldsymbol{u}_{k,t},\boldsymbol{v}_{k,t}) - F_k(\boldsymbol{u}_k^*,\boldsymbol{v}_k^*)\right), \tag{B.19}
$$

and

$$
\|\nabla_{\boldsymbol{v}}F_k(\boldsymbol{u}_{k,t},\boldsymbol{v}_{k,t})\|^2 \le 2L_v\left(F_k(\boldsymbol{u}_{k,t},\boldsymbol{v}_{k,t}) - F_k(\boldsymbol{u}_k^*,\boldsymbol{v}_k^*)\right). \tag{B.20}
$$

Substituting (B.19) and (B.20) into (B.18), we have

$$
\begin{aligned}
F(\boldsymbol{W}_{t+1}) - F(\boldsymbol{W}^*) \le{}& A_3(F(\boldsymbol{W}_t) - F(\boldsymbol{W}^*)) + A_1 + A_2 \\
&+ A_2\sum_{k=1}^{K}\frac{1}{D}\frac{1}{D_k}C\sum_{c=1}^{C}D_{k,c}^2\left(\frac{D_c - \sum_{k=1}^{K}\alpha_{k,t-1}D_{k,c}}{D_c}\right)^2,
\end{aligned}
\tag{B.21}
$$

where $A_3 = 1 + (4L_u^2\eta_u^2 + 2L_v^2\eta_v^2)(1+\chi)\tau^2 - (\eta_u L_u + \eta_v L_v)\tau$. By telescoping the above inequality, we have

$$
\begin{aligned}
F(\boldsymbol{W}_T) - F(\boldsymbol{W}^*) \le{}& A_3^T(F(\boldsymbol{W}_0) - F(\boldsymbol{W}^*)) + \frac{1-A_3^T}{1-A_3}(A_1 + A_2) \\
&+ A_2\sum_{t=1}^{T-1}A_3^{T-1-t}\sum_{k=1}^{K}\frac{1}{D}\frac{1}{D_k}C\sum_{c=1}^{C}\frac{D_{k,c}^2}{D_c^2}\left(D_c - \sum_{k=1}^{K}\alpha_{k,t-1}D_{k,c}\right)^2.
\end{aligned}
\tag{B.22}
$$

Below we bounding the last term on the RHS of (B.22) as

$$
\begin{aligned}
&A_2\sum_{t=1}^{T-1}A_3^{T-1-t}\sum_{k=1}^{K}\frac{1}{D}\frac{1}{D_k}C\sum_{c=1}^{C}\frac{D_{k,c}^2}{D_c^2}\left(D_c - \sum\nolimits_{k=1}^{K}\alpha_{k,t-1}D_{k,c}\right)^2 \\
&= A_2\sum_{t=0}^{T-2}A_3^{T-2-t}\sum_{k=1}^{K}\frac{1}{D}\frac{1}{D_k}C\sum_{c=1}^{C}\frac{D_{k,c}^2}{D_c^2}\left(D_c - \sum\nolimits_{k=1}^{K}\alpha_{k,t}D_{k,c}\right)^2 \\
&\overset{(a)}{\le} \frac{A_2KC}{D}\sum_{t=0}^{T-2}A_3^{T-2-t}\sum_{k=1}^{K}\sum_{c=1}^{C}\frac{D_{k,c}^2}{D_kD_c^2}\sum_{k=1}^{K}(1-\alpha_{k,t})D_{k,c}^2
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{A_2 C K}{D} \sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} \sum_{c=1}^{C} \frac{D_{k,c}^2}{D_k D_c^2} \sum_{k=1}^{K} D_{k,c}^2 \\
&- \frac{A_2 K}{D} \sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} C \sum_{c=1}^{C} \frac{D_{k,c}^2}{D_k D_c^2} \sum_{k=1}^{K} \alpha_{k,t} D_{k,c}^2, \qquad \text{(B.23)}
\end{aligned}
$$

where (a) is due to Jensen's inequality and $(1-\alpha_{k,t})^2 = 1-\alpha_{k,t}$. For the last term on the RHS of (B.23), we have

$$
\begin{aligned}
&\frac{A_2 K}{D} \sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} C \sum_{c=1}^{C} \frac{D_{k,c}^2}{D_k D_c^2} \sum_{k=1}^{K} \alpha_{k,t} D_{k,c}^2 \\
&\overset{(a)}{\geq} A_2 \frac{1}{DK(T-1)} \left(\sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{c=1}^{C} \sum_{k=1}^{K} \frac{D_{k,c}}{\sqrt{D_k} D_c} \sum\nolimits_{k=1}^{K} \alpha_{k,t} D_{k,c} \right)^2 \\
&\geq A_2 \frac{1}{DK(T-1)} \frac{1}{\max_{1 \leq k \leq K} D_k} \left(\sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} \alpha_{k,t} \sum_{c=1}^{C} D_{k,c} \right)^2 \\
&= A_2 \frac{1}{DK(T-1)} \frac{1}{\max_{1 \leq k \leq K} D_k} \left(\sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} \alpha_{k,t} D_k \right)^2, \qquad \text{(B.24)}
\end{aligned}
$$

where (a) follows Jensen's inequality. Substituting (B.23) and (B.24) into (B.22), the proof is completed.

B.4 Proof of Proposition 1

The first-order and second-order derivatives of the objective function (4.23) with respect to $\mathcal{T}_{k,t}^{\mathrm{U}}$ are

$$
\frac{\partial \left(\sum_{k \in \boldsymbol{S}_t} q_k(t) E_{k,t} \right)}{\partial \mathcal{T}_{k,t}^{\mathrm{U}}} = q_k(t) \frac{\theta_{k,t} B N_0 \mathcal{T}_{k,t}^{\mathrm{U}} - N_0 Q q \ln 2}{h_{k,t} \mathcal{T}_{k,t}^{\mathrm{U}}} 2^{\frac{Qq}{\theta_{k,t} B \mathcal{T}_{k,t}^{\mathrm{U}}}} - \frac{q_k(t) \theta_{k,t} B N_0}{h_{k,t}}, \qquad \text{(B.25)}
$$

and

$$
\frac{\partial^2 \left(\sum_{k \in \boldsymbol{S}_t} q_k(t) E_{k,t} \right)}{\partial (\mathcal{T}_{k,t}^{\mathrm{U}})^2} = \frac{q_k(t) Q^2 q^2 N_0 (\ln 2)^2}{\theta_{k,t} B h_{k,t} \left(\mathcal{T}_{k,t}^{\mathrm{U}} \right)^3} 2^{\frac{Qq}{\theta_{k,t} B \mathcal{T}_{k,t}^{\mathrm{U}}}} \geq 0. \qquad \text{(B.26)}
$$

Thus, $\frac{\partial \left(\sum_{k \in \boldsymbol{S}_t} q_k(t) E_{k,t} \right)}{\partial \mathcal{T}_{k,t}^{\mathrm{U}}}$ is an increasing function with respect to $\mathcal{T}_{k,t}^{\mathrm{U}}$. Since $\lim\limits_{\mathcal{T}_{k,t}^{\mathrm{U}} \to \infty} \frac{dE_{k,t}^{\mathrm{U}}}{d\mathcal{T}_{k,t}^{\mathrm{U}}} = 0$, we have $\frac{\partial \left(\sum_{k \in \boldsymbol{S}_t} q_k(t) E_{k,t} \right)}{\partial \mathcal{T}_{k,t}^{\mathrm{U}}} \leq 0$. That is, the objective function (4.23) is an non-increasing function with respect to the communication time $\mathcal{T}_{k,t}^{\mathrm{U}}$. The optimal completion time of device k is $\mathcal{T}_{k,t}^{\mathrm{U}} = \mathcal{T}_{\max} - \mathcal{T}_{k}^{\mathrm{L}}$. Thus, the optimal transmit power of device k satisfy

(4.24).

B.5 Proof of Lemma 8

Problem (4.25) is a typical convex optimization problem, its proof is similar to the proof of Proposition 1, and thus omitted for brevity. By using KKT conditions, the Lagrange function of problem (4.25) is

$$\mathcal{L}(\theta_t, \mu) = \sum_{k \in \boldsymbol{S}_t} q_k(t) \frac{\theta_{k,t} B N_0 (\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}})}{h_{k,t}} \mathcal{I}(\theta_{k,t}) + \mu \left(\sum_{k=1}^{K} \theta_{k,t} - 1 \right), \tag{B.27}$$

where μ is the Lagrange multiplier associated with constraint (4.14c). The first-order derivative of $\mathcal{L}(\theta_t, \mu)$ is

$$\frac{\partial \mathcal{L}(\theta_t, \mu)}{\partial \theta_{k,t}} = \frac{B N_0 q_k(t) (\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}})}{h_{k,t}} \big(\mathcal{I}(\theta_{k,t}) + \theta_{k,t} \mathcal{I}'(\theta_{k,t}) \big) + \mu. \tag{B.28}$$

Let $\frac{\partial \mathcal{L}(\theta_t, \mu)}{\partial \theta_{k,t}} = 0$, we have

$$\mathcal{I}(\theta_{k,t}) + \theta_{k,t} \mathcal{I}'(\theta_{k,t}) = \frac{-\mu h_{k,t}}{B N_0 q_k(t) (\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}})}. \tag{B.29}$$

Its inverse function is shown to be (4.28). Given constraint (4.25a), the optimal bandwidth allocation policy is given as (4.27). In addition, similar to the proof of Proposition 1, one can prove that the objective function (4.25) is a decreasing function of $\theta_{k,t}$. Thus, $\sum_{k=1}^{K} \theta_{k,t}^* = 1$ always holds for the optimal solution.

B.6 Proof of Proposition 2

For the ease of presentation, we define the Lyapunov function as $\mathcal{V}(t) = \sum_{k=1}^{K} \frac{1}{2} q_{k,t}^2$, the Lyapunov drift of round t as $\Delta_1(t) = \mathcal{V}(t+1) - \mathcal{V}(t)$. According to the evolution of the virtual queue defined in (4.21), we have $q_{k,t+1}^2 \leq \left(q_{k,t} + \alpha_{k,t} E_{k,t} - \frac{E_k}{T} \right)^2$. For $\Delta_1(t)$, we have

$$\Delta_1(t) = \frac{1}{2} \sum_{k=1}^{K} (q_{k,t+1}^2 - q_{k,t}^2) \leq \sum_{k=1}^{K} \left(\frac{1}{2} (q_{k,t} + \alpha_{k,t} E_{k,t} - \frac{E_k}{T})^2 - \frac{1}{2} q_{k,t}^2 \right)$$

$$\leq \zeta_0 + \sum_{k=1}^{K} q_{k,t} \left(\alpha_{k,t} E_{k,t} - \frac{E_k}{T} \right), \tag{B.30}$$

where $\zeta_0 = \frac{1}{2}\sum_{k=1}^{K} \zeta_k^2$, $\zeta_k = \max_t \left\{ \left| \alpha_{k,t} E_{k,t} - \frac{E_k}{T} \right| \right\}$. By adding $-V\gamma_t \sum_{k=1}^{K} \alpha_{k,t} D_k$ on both sides of (B.30), an upper bound of the one-round drift-plus-penalty function is given by

$$\Delta_1(t) - V\gamma_t \sum_{k=1}^{K} \alpha_{k,t} D_k \leq \zeta_0 + \sum_{k=1}^{K} q_{k,t} (\alpha_{k,t} E_{k,t} - \frac{E_k}{T}) - V\gamma_t \sum_{k=1}^{K} \alpha_{k,t} D_k. \tag{B.31}$$

The drift-plus-penalty algorithm of Lyapunov optimization aims to minimize the upper bound of $\Delta_1(t) - \gamma_t V \sum_{k=1}^{K} \alpha_{k,t} D_k$. Define the T-round drift as $\Delta_T = \mathcal{V}(T-1) - \mathcal{V}(0) = \sum_{k=1}^{K} \frac{1}{2} q_{k,T-1}^2$. Then, the T-round drift-plus-penalty function can be bounded by

$$\Delta_T - V \sum_{t=0}^{T-1} \gamma_t \sum_{k=1}^{K} \alpha_{k,t} D_k \leq T\zeta_0 + \sum_{t=0}^{T-1} \left(\sum_{k=1}^{K} q_{k,t} (\alpha_{k,t} E_{k,t} - \frac{E_k}{T}) - V\gamma_t \sum_{k=1}^{K} \alpha_{k,t} D_k \right). \tag{B.32}$$

Based on the above analysis, we first prove the feasibility of the proposed algorithm. We use superscript $*$ to denote the optimal offline solution of problem (4.22), superscript $\dagger$ to represent the solution of the proposed drift-plus-penalty algorithm. For a feasible solution with $\alpha_{k,t} = 0$ and $E_{k,t} = 0$, we have

$$\Delta_T = \sum\nolimits_{k=1}^{K} \frac{1}{2} q_{k,T-1}^2 \leq T\zeta_0 + V \sum\nolimits_{t=0}^{T-1} \gamma_t D. \tag{B.33}$$

Thus, we have

$$\left(\sum_{k=1}^{K} q_{k,T-1} \right)^2 \leq K \sum_{k=1}^{K} q_{k,T-1}^2 \leq 2K \left(T\zeta_0 + V \sum_{t=0}^{T-1} \gamma_t D \right), \tag{B.34}$$

where the first inequation comes from Jensen's inequality. According to the evolution of the virtual queue defined in (4.21), we have $\alpha_{k,t} E_{k,t} - \frac{E_k}{T} \leq q_{k,t+1} - q_{k,t}$, summing this inequation over T rounds, we have

$$\sum_{t=0}^{T-1} \sum_{k=1}^{K} \left(\alpha_{k,t} E_{k,t} - \frac{E_k}{T} \right) \leq \sum_{t=0}^{T-1} \sum_{k=1}^{K} (q_{k,t+1} - q_{k,t}) \leq \sqrt{2K \left(T\zeta_0 + V \sum_{t=0}^{T-1} \gamma_t D \right)}. \tag{B.35}$$

By rearranging the above inequation, the energy consumption bound in (4.31) is derived. Below we analyze the optimality of the proposed drift-plus-penalty algorithm, which minimize the RHS in (B.32). Since Δ_T is positive, based on (B.32), we have

$$-V\sum_{t=0}^{T-1}\gamma_t\sum_{k=1}^{K}\alpha_{k,t}^{\dagger}D_{k,c} \le T\zeta_0 + \sum_{t=0}^{T-1}\sum_{k=1}^{K}q_{k,t}\left(\alpha_{k,t}^{*}E_{k,t} - \frac{E_k^*}{T}\right) - V\sum_{t=0}^{T-1}\gamma_t\sum_{k=1}^{K}\alpha_{k,t}^{*}D_{k,c}, \tag{B.36}$$

Next, we bound the second term in the RHS of (B.36) as

$$\begin{aligned}\sum_{t=0}^{T-1}\sum_{k=1}^{K}q_{k,t}\left(\alpha_{k,t}^{*}E_{k,t} - \frac{E_k}{T}\right) &= \sum_{t=0}^{T-1}\sum_{k=1}^{K}(q_{k,t} - q_{k,0})\left(\alpha_{k,t}^{*}E_{k,t} - \frac{E_k}{T}\right) \\ &\le \frac{T(T-1)}{2}\sum_{k=1}^{K}\zeta_k^2. \end{aligned} \tag{B.37}$$

Substituting (B.37) into (B.36), the inequation (4.30) is derived, and the proof is completed.

Appendix C

Proof in Chapter 4

C.1 Proof of Lemma 9

For $\lambda = 1$, the bound trivially holds since $\boldsymbol{w}_{k,t}^{(0)} = \boldsymbol{w}_t$. For $\lambda \geq 2$, we have

$$
\begin{aligned}
&\mathbb{E}\|\boldsymbol{w}_{k,t}^{(l)} - \boldsymbol{w}_t\|^2 = \mathbb{E}\|\boldsymbol{w}_{k,t}^{(l-1)} - \boldsymbol{w}_t - \eta\tilde{\nabla}F_k(\boldsymbol{w}_{k,t}^{(l-1)})\|^2 \\
&\overset{(a)}{=} \mathbb{E}\left\|\boldsymbol{w}_{k,t}^{(l-1)} - \boldsymbol{w}_t - \eta\nabla F_k(\boldsymbol{w}_{k,t}^{(l-1)})\right\|^2 + \eta^2\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{k,t}^{(l-1)}) - \tilde{\nabla}F_k(\boldsymbol{w}_{k,t}^{(l-1)})\right\|^2 \\
&\overset{(b)}{\leq} \mathbb{E}\left\|\boldsymbol{w}_{k,t}^{(l-1)} - \boldsymbol{w}_t - \eta\nabla F_k(\boldsymbol{w}_{k,t}^{(l-1)})\right\|^2 + \eta^2\sigma^2 \\
&= \mathbb{E}\left\|\boldsymbol{w}_{k,t}^{(l-1)} - \boldsymbol{w}_t\right\|^2 + \eta^2\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{k,t}^{(l-1)})\right\|^2 \\
&- 2\mathbb{E}\left\langle \frac{1}{\sqrt{\lambda-1}}(\boldsymbol{w}_{k,t}^{(l-1)} - \boldsymbol{w}_t), \eta\sqrt{\lambda-1}\nabla F_k(\boldsymbol{w}_{k,t}^{(l-1)})\right\rangle + \eta^2\sigma^2 \\
&\overset{(c)}{\leq} (1 + \frac{1}{\lambda-1})\mathbb{E}\left\|\boldsymbol{w}_{k,t}^{(l-1)} - \boldsymbol{w}_t\right\|^2 + \lambda\eta^2\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{k,t}^{(l-1)})\right\|^2 + \eta^2\sigma^2 \\
&\overset{(d)}{\leq} (1 + \frac{1}{\lambda-1})\mathbb{E}\left\|\boldsymbol{w}_{k,t}^{(l-1)} - \boldsymbol{w}_t\right\|^2 + 2\lambda\eta^2\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{k,t}^{(l-1)}) - \nabla F_k(\boldsymbol{w}_t)\right\|^2 \\
&+ 2\lambda\eta^2\|\nabla F_k(\boldsymbol{w}_t)\|^2 + \eta^2\sigma^2 \\
&\overset{(e)}{\leq} \left(1 + \frac{1}{\lambda-1} + 2\lambda\eta^2L^2\right)\mathbb{E}\left\|\boldsymbol{w}_{k,t}^{(l-1)} - \boldsymbol{w}_t\right\|^2 + 2\lambda\eta^2\|\nabla F_k(\boldsymbol{w}_t)\|^2 + \eta^2\sigma^2,
\end{aligned}
\tag{C.1}
$$

where (a) is derived by adding and subtracting $\nabla F_k(\boldsymbol{w}_{k,t}^{(l-1)})$ into $\tilde{\nabla}F_k(\boldsymbol{w}_{k,t}^{(l-1)})$ and using the unbiased stochastic gradient in Assumption 8, (b) is due to the bounded variance of stochastic gradient in Assumption 8, (c) comes from the triangle inequality, (d) is derived by adding and subtracting $\nabla F_k(\boldsymbol{w}_t)$ into $\nabla F_k(\boldsymbol{w}_{k,t}^{(l-1)})$ and using the triangle inequality, (e) is due to the L-smooth of local loss functions in Assumption 7. Let $\eta \leq \frac{1}{2\lambda L}$, we have $2\lambda\eta^2 L^2 \leq \frac{1}{2\lambda} \leq \frac{1}{2(\lambda-1)}$. Thus, we have

$$\underbrace{\mathbb{E}\left\|\boldsymbol{w}_{k,t}^{(l)} - \boldsymbol{w}_t\right\|^2}_{y_l} \leq \underbrace{\left(1 + \frac{3}{2(\lambda-1)}\right)}_{c_1} \underbrace{\mathbb{E}\left\|\boldsymbol{w}_{k,t}^{(l-1)} - \boldsymbol{w}_t\right\|^2}_{y_{l-1}} + \underbrace{2\lambda\eta^2 \left\|\nabla F_k(\boldsymbol{w}_t)\right\|^2 + \eta^2\sigma^2}_{c_2}. \tag{C.2}$$

By telescoping the above inequation, we have $y_l = c_2\frac{1-c_1^l}{1-c_1} \leq c_2\frac{c_1^{\lambda-1}-1}{c_1-1}$. That is,

$$\mathbb{E}\left\|\boldsymbol{w}_{k,t}^{(l)} - \boldsymbol{w}_t\right\|^2 \leq \left(2\lambda\eta^2 \left\|\nabla F_k(\boldsymbol{w}_t)\right\|^2 + \eta^2\sigma^2\right) \frac{\left(1+\frac{3}{2(\lambda-1)}\right)^{\lambda-1} - 1}{\frac{3}{2(\lambda-1)}}. \tag{C.3}$$

In (C.3), we have $\left(1+\frac{3}{2(\lambda-1)}\right)^{\lambda-1} = \left(1+\frac{3}{2(\lambda-1)}\right)^{\frac{2(\lambda-1)}{3}\frac{3}{2}} \leq e^{\frac{3}{2}} \leq 5$ and $\frac{2(\lambda-1)}{3} \leq (\lambda-1)$. Thus,

$$\begin{aligned}\mathbb{E}\left\|\boldsymbol{w}_{k,t}^{(l)} - \boldsymbol{w}_t\right\|^2 &\leq 4(\lambda-1)\left(2\lambda\eta^2 \left\|\nabla F_k(\boldsymbol{w}_t)\right\|^2 + \eta^2\sigma^2\right) \\ &\overset{(a)}{\leq} 4(\lambda-1)(2\lambda\eta^2 G^2 + \eta^2\sigma^2), \end{aligned}\tag{C.4}$$

where (a) follows Assumption 9. Let $\eta = \frac{\tilde{\eta}}{\lambda}$, the proof is completed.

C.2 Proof of Lemma 10

For any two round t and t' that satisfies $t \geq t'$, we have

$$\begin{aligned}&\mathbb{E}\|\boldsymbol{w}_t - \boldsymbol{w}_{t'}\|^2 = \mathbb{E}\left\|\sum_{j=t'}^{t-1}(\boldsymbol{w}_{j+1} - \boldsymbol{w}_j)\right\|^2 \\ &= \tilde{\eta}^2\mathbb{E}\left\|\sum_{j=t'}^{t-1}\frac{1}{\lambda K}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\tilde{\nabla}F_k\left(\boldsymbol{w}_{k,j-\tau_{k,j}}^{(l)}\right)\right\|^2 \\ &\overset{(a)}{\leq} \tilde{\eta}^2(t-t')\sum_{j=t'}^{t-1}\mathbb{E}\left\|\frac{1}{\lambda K}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\tilde{\nabla}F_k\left(\boldsymbol{w}_{k,j-\tau_{k,j}}^{(l)}\right)\right\|^2 \\ &\overset{(b)}{\leq} 3\tilde{\eta}^2(t-t')\sum_{j=t'}^{t-1}\mathbb{E}\left\|\frac{1}{\lambda K}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\tilde{\nabla}F_k\left(\boldsymbol{w}_{k,j-\tau_{k,j}}^{(l)}\right) - \nabla F_k\left(\boldsymbol{w}_{k,j-\tau_{k,j}}^{(l)}\right)\right)\right\|^2 \\ &+ 3\tilde{\eta}^2(t-t')\sum_{j=t'}^{t-1}\mathbb{E}\left\|\frac{1}{\lambda K}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\nabla F_k\left(\boldsymbol{w}_{k,j-\tau_{k,j}}^{(l)}\right) - \nabla F_k\left(\boldsymbol{w}_{j-\tau_{k,j}}\right)\right)\right\|^2\end{aligned}$$

$$
\begin{aligned}
&+ 3\tilde{\eta}^2 (t-t') \sum_{j=t'}^{t-1} \mathbb{E} \left\| \frac{1}{K} \sum_{k=1}^{K} \nabla F_k \left(\boldsymbol{w}_{j-\tau_{k,j}} \right) \right\|^2 \\
&\overset{(c)}{\leq} 3\tilde{\eta}^2 (t-t') \sum_{j=t'}^{t-1} \frac{1}{\lambda K} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \mathbb{E} \left\| \tilde{\nabla} F_k \left(\boldsymbol{w}_{k,j-\tau_{k,j}}^{(l)} \right) - \nabla F_k \left(\boldsymbol{w}_{k,j-\tau_{k,j}}^{(l)} \right) \right\|^2 \\
&+ 3\tilde{\eta}^2 (t-t') \sum_{j=t'}^{t-1} \frac{1}{\lambda K} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \mathbb{E} \left\| \nabla F_k \left(\boldsymbol{w}_{k,j-\tau_{k,j}}^{(l)} \right) - \nabla F_k \left(\boldsymbol{w}_{j-\tau_{k,j}} \right) \right\|^2 \\
&+ 3\tilde{\eta}^2 (t-t') \sum_{j=t'}^{t-1} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E} \left\| \nabla F_k \left(\boldsymbol{w}_{j-\tau_{k,j}} \right) \right\|^2 \\
&\overset{(d)}{\leq} 3\tilde{\eta}^2 (t-t') \sum_{j=t'}^{t-1} \frac{1}{\lambda K} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} L^2 \mathbb{E} \left\| \boldsymbol{w}_{k,j-\tau_{k,j}}^{(l)} - \boldsymbol{w}_{j-\tau_{k,j}} \right\|^2 + 3\tilde{\eta}^2 (t-t')^2 (\sigma^2 + G^2), \quad \text{(C.5)}
\end{aligned}
$$

where (a) is due to Jensen's inequality, (b) is derived by adding and subtracting both $\nabla F_k(\boldsymbol{w}_{k,j-\tau_{k,j}}^l)$ and $\nabla F_k(\boldsymbol{w}_{j-\tau_{k,j}})$ into $\tilde{\nabla} F_k(\boldsymbol{w}_{k,j-\tau_{k,j}}^l)$, then using the triangle inequality, (c) follows the Jensen's inequality, (d) comes from the Assumption 7, 8, and 9. According to Lemma 9, we have

$$
\mathbb{E} \left\| \boldsymbol{w}_{k,j-\tau_{k,j}}^l - \boldsymbol{w}_{j-\tau_{k,j}} \right\|^2 \leq \frac{4(\lambda-1)\tilde{\eta}^2}{\lambda} (2G^2 + \frac{\sigma^2}{\lambda}). \tag{C.6}
$$

Substituting (C.6) into (C.5), the proof is completed.

C.3 Proof of Theorem 4

By using the L-smooth of the loss functions, we have

$$
F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}_t) \leq \langle \nabla F(\boldsymbol{w}_t), \boldsymbol{w}_{t+1} - \boldsymbol{w}_t \rangle + \frac{L}{2} \left\| \boldsymbol{w}_{t+1} - \boldsymbol{w}_t \right\|^2. \tag{C.7}
$$

Thus, we have

$$
\begin{aligned}
&\mathbb{E}\left[F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}_t) \right] \\
&\leq \mathbb{E}\left[-\tilde{\eta} \left\langle \nabla F(\boldsymbol{w}_t), \frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \tilde{\nabla} F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)}) \right\rangle \right] + \frac{L\tilde{\eta}}{2} \mathbb{E} \left\| \frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \tilde{\nabla} F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)}) \right\|^2 \\
&= \underbrace{\mathbb{E}\left[-\tilde{\eta} \left\langle \nabla F(\boldsymbol{w}_t), \frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \left(\tilde{\nabla} F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)}) - \nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)}) \right) \right\rangle \right]}_{A_1}
\end{aligned}
$$

$$
\begin{aligned}
&+ \underbrace{\mathbb{E}\left[-\tilde{\eta}\left\langle \nabla F(\boldsymbol{w}_t), \frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}})\right\rangle\right]}_{A_2} \\
&+ \underbrace{\frac{L\tilde{\eta}}{2}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\tilde{\nabla} F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}})\right\|^2}_{A_3}.
\end{aligned}
\tag{C.8}
$$

Below we focus on bounding the three terms in (C.8). Due to reuse of noisy gradient, the stochastic gradient noise $\tilde{\nabla} F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}}) - \nabla F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}})$ is correlated with $\boldsymbol{w}_t$. Thus, $A_1 \neq 0$. For A_1, we have

$$
\begin{aligned}
A_1 &= \mathbb{E}\left[-\tilde{\eta}\left\langle \nabla F(\boldsymbol{w}_t), \frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\tilde{\nabla} F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}}) - \nabla F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}})\right)\right\rangle\right] \\
&= \underbrace{\mathbb{E}\left[-\tilde{\eta}\left\langle \nabla F(\boldsymbol{w}_t) - \nabla F(\boldsymbol{w}_{t-\tau_{t,k}}), \frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\tilde{\nabla} F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}}) - \nabla F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}})\right)\right\rangle\right]}_{B_1} \\
&\quad - \underbrace{\mathbb{E}\left[\tilde{\eta}\left\langle \nabla F(\boldsymbol{w}_{t-\tau_{t,k}}), \frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\tilde{\nabla} F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}}) - \nabla F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}})\right)\right\rangle\right]}_{B_2}.
\end{aligned}
\tag{C.9}
$$

Since $\boldsymbol{w}_{t-\tau_{t,k}}$ is independent with $\tilde{\nabla} F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}}) - \nabla F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}})$, we have $B_2 = 0$. For B_1, we have:

$$
\begin{aligned}
B_1 &= -\tilde{\eta}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\left[\left\langle \nabla F(\boldsymbol{w}_t) - \nabla F(\boldsymbol{w}_{t-\tau_{t,k}}), \tilde{\nabla} F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}}) - \nabla F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}})\right\rangle\right] \\
&\overset{(a)}{\leq} \frac{1}{2}\tilde{\eta}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\nabla F(\boldsymbol{w}_t) - \nabla F(\boldsymbol{w}_{t-\tau_{t,k}})\right\|^2 \\
&+ \frac{1}{2}\tilde{\eta}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\left\|\tilde{\nabla} F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}}) - \nabla F_k(\boldsymbol{w}^{(l)}_{k,t-\tau_{t,k}})\right\|^2 \\
&\overset{(b)}{\leq} \frac{1}{2}\tilde{\eta}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\nabla F(\boldsymbol{w}_t) - \nabla F(\boldsymbol{w}_{t-\tau_{t,k}})\right\|^2 + \frac{1}{2}\tilde{\eta}\sigma^2 \\
&\overset{(c)}{\leq} \frac{1}{2}\tilde{\eta}L^2\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\boldsymbol{w}_t - \boldsymbol{w}_{t-\tau_{t,k}}\right\|^2 + \frac{1}{2}\tilde{\eta}\sigma^2,
\end{aligned}
\tag{C.10}
$$

where (a) follows the triangle inequality, (b) is due to Assumption 8, (c) comes from the L-smooth of the loss function. According to the above analysis, we have

$$
A_1 \leq \frac{1}{2}\tilde{\eta}L^2\frac{1}{K}\sum\nolimits_{k=1}^{K}\mathbb{E}\left\|\boldsymbol{w}_t - \boldsymbol{w}_{t-\tau_{t,k}}\right\|^2 + \frac{1}{2}\tilde{\eta}\sigma^2. \tag{C.11}
$$

For A_2, we have

$$\begin{aligned}
A_2 &= -\tilde{\eta}\mathbb{E}\left[\left\langle \nabla F(\boldsymbol{w}_t), \frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})\right\rangle\right] \\
&= -\tilde{\eta}\|\nabla F(\boldsymbol{w}_t)\|^2 + \tilde{\eta}\mathbb{E}\left[\left\langle \nabla F(\boldsymbol{w}_t), \nabla F(\boldsymbol{w}_t) - \frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})\right\rangle\right] \\
&= -\tilde{\eta}\|\nabla F(\boldsymbol{w}_t)\|^2 + \tilde{\eta}\mathbb{E}\left[\left\langle \nabla F(\boldsymbol{w}_t), \frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\nabla F_k(\boldsymbol{w}_t) - \nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})\right)\right\rangle\right] \\
&\leq -\frac{1}{2}\tilde{\eta}\|\nabla F(\boldsymbol{w}_t)\|^2 + \frac{1}{2}\tilde{\eta}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\nabla F_k(\boldsymbol{w}_t) - \nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})\right)\right\|^2 \\
&\leq -\frac{1}{2}\tilde{\eta}\|\nabla F(\boldsymbol{w}_t)\|^2 + \tilde{\eta}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\nabla F_k(\boldsymbol{w}_t) - \nabla F_k(\boldsymbol{w}_{t-\tau_{t,k}})\right)\right\|^2 \\
&+ \tilde{\eta}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\nabla F_k(\boldsymbol{w}_{t-\tau_{t,k}}) - \nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})\right)\right\|^2 \\
&\leq -\frac{1}{2}\tilde{\eta}\left\|\nabla F(\boldsymbol{w}_t)\right\|^2 + \tilde{\eta}\frac{1}{K}\sum_{k=1}^{K}L^2\mathbb{E}\left\|\boldsymbol{w}_t - \boldsymbol{w}_{t-\tau_{t,k}}\right\|^2 \\
&+ \tilde{\eta}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}L^2\mathbb{E}\left\|\boldsymbol{w}_{t-\tau_{t,k}} - \boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)}\right\|^2. \qquad \text{(C.12)}
\end{aligned}$$

For A_3, we have

$$\begin{aligned}
A_3 &= \frac{L\tilde{\eta}}{2}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\tilde{\nabla}F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})\right\|^2 \\
&\overset{(a)}{\leq} L\tilde{\eta}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\tilde{\nabla}F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)}) - \nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})\right)\right\|^2 \\
&+ L\tilde{\eta}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})\right\|^2 \\
&\overset{(b)}{\leq} L\tilde{\eta}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\left\|\tilde{\nabla}F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)}) - \nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})\right\|^2 \\
&+ L\tilde{\eta}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})\right\|^2 \\
&\overset{(c)}{\leq} L\tilde{\eta}\sigma^2 + L\tilde{\eta}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})\right\|^2, \qquad \text{(C.13)}
\end{aligned}$$

where (a) is derived by adding and subtracting $\nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})$ into $\tilde{\nabla}F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)})$ and using the triangle inequality, (b) follows Jensen's inequality, (c) is due to the bounded variance of stochastic gradient in Assumption 8. Below we bound $\left\| \frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)}) \right\|^2$ in (C.13) as:

$$
\begin{aligned}
&\mathbb{E}\left\| \frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)}) \right\|^2 \\
&\overset{(a)}{\leq} 3\mathbb{E}\left\| \frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \Big(\nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)}) - \nabla F_k(\boldsymbol{w}_{t-\tau_{t,k}}) \Big) \right\|^2 \\
&+3\mathbb{E}\left\| \frac{1}{K} \sum_{k=1}^{K} \left(\nabla F_k(\boldsymbol{w}_{t-\tau_{t,k}}) - \nabla F_k(\boldsymbol{w}_t) \right) \right\|^2 + 3\left\|\nabla F(\boldsymbol{w}_t)\right\|^2 \\
&\overset{(b)}{\leq} \frac{3}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \mathbb{E}\left\| \nabla F_k(\boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)}) - \nabla F_k(\boldsymbol{w}_{t-\tau_{t,k}}) \right\|^2 \\
&+\frac{3}{K} \sum_{k=1}^{K} \mathbb{E}\left\| \nabla F_k(\boldsymbol{w}_{t-\tau_{t,k}}) - \nabla F_k(\boldsymbol{w}_t) \right\|^2 + 3\left\|\nabla F(\boldsymbol{w}_t)\right\|^2 \\
&\overset{(c)}{\leq} 3\frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} L^2 \mathbb{E}\left\| \boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)} - \boldsymbol{w}_{t-\tau_{t,k}} \right\|^2 \\
&+3\frac{1}{K} \sum_{k=1}^{K} L^2 \mathbb{E}\left\| \boldsymbol{w}_{t-\tau_{t,k}} - \boldsymbol{w}_t \right\|^2 + 3\|\nabla F(\boldsymbol{w}_t)\|^2.
\end{aligned}
\tag{C.14}
$$

Substituting (C.14) into (C.13), we have

$$
\begin{aligned}
A_3 \leq L\tilde{\eta}\sigma^2 &+ 3L\tilde{\eta}\frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} L^2 \mathbb{E}\left\| \boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)} - \boldsymbol{w}_{t-\tau_{t,k}} \right\|^2 \\
&+3L\tilde{\eta}\frac{1}{K} \sum_{k=1}^{K} L^2 \mathbb{E}\left\| \boldsymbol{w}_{t-\tau_{t,k}} - \boldsymbol{w}_t \right\|^2 + 3L\tilde{\eta}\left\|\nabla F(\boldsymbol{w}_t)\right\|^2.
\end{aligned}
\tag{C.15}
$$

Substituting (C.11), (C.12), and (C.15) into (C.8), we have:

$$
\begin{aligned}
\mathbb{E}\left[F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}_t)\right] \leq &\left(-\frac{1}{2}\tilde{\eta} + 3L\tilde{\eta} \right) \|\nabla F(\boldsymbol{w}_t)\|^2 + (\frac{1}{2} + L)\tilde{\eta}\sigma^2 \\
&+ (\frac{3}{2} + 3L)\tilde{\eta}L^2 \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left\| \boldsymbol{w}_t - \boldsymbol{w}_{t-\tau_{t,k}} \right\|^2 \\
&+ (1+3L)\tilde{\eta}L^2 \frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \mathbb{E}\left\| \boldsymbol{w}_{t-\tau_{t,k}} - \boldsymbol{w}_{k,t-\tau_{t,k}}^{(l)} \right\|^2.
\end{aligned}
\tag{C.16}
$$

According to Lemma 9 and Lemma 10, as well as $\tilde{\eta} \leq \frac{1}{2L}$ we have

$$\begin{aligned}
&\mathbb{E}\left[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)\right]\\
&\quad\leq\left(-\frac{1}{2}\tilde{\eta}+3L\tilde{\eta}\right)\|\nabla F(\boldsymbol{w}_t)\|^2+\frac{(\tilde{\eta}+3\tilde{\eta}L)\,(\lambda-1)}{\lambda}\left(2G^2+\frac{\sigma^2}{\lambda}\right)+\frac{(\tilde{\eta}+1)\,\sigma^2}{2}\\
&\quad+\frac{9}{8}(\tilde{\eta}+1)\left(\left(1+\frac{2(\lambda-1)}{\lambda}\right)G^2+\left(1+\frac{(\lambda-1)}{\lambda^2}\right)\sigma^2\right)\mathbb{E}\left[\frac{1}{K}\sum\nolimits_{k=1}^{K}\tau_{k,t}^2\right]. \qquad (C.17)
\end{aligned}$$

According to the evolution of devices' staleness in (5.19), we have $\tau_{k,t}=(1-\alpha_{k,t}s_{k,t})(\tau_{k,t-1}+1)$. Note that $\alpha_{k,t}s_{k,t}\in\{0,1\}$, which induces $(1-\alpha_{k,t}s_{k,t})^2=(1-\alpha_{k,t}s_{k,t})$. Thus, we have

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\tau_{k,t}^2\right]&=\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}(1-\alpha_{k,t}s_{k,t})^2(\tau_{k,t-1}+1)^2\right]\\
&=\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}(1-\alpha_{k,t}s_{k,t})\,(\tau_{k,t-1}+1)^2\right]\\
&\overset{(a)}{=}\frac{1}{K}\sum_{k=1}^{K}(\tau_{k,t-1}+1)^2\left(1-\alpha_{k,t}\sum_{r=1}^{R}z_{k,t}^{(r)}\Pr(\mathrm{SINR}_{k,t}^{(r)}\geq\gamma_{\mathrm{th}})\right), \qquad (C.18)
\end{aligned}$$

where the last inequality is due to $\mathbb{E}\left[s_{k,t}\right]=\sum_{r=1}^{R}z_{k,t}^{(r)}\Pr(\mathrm{SINR}_{k,t}^{(r)}\geq\gamma_{\mathrm{th}})$. By substituting (C.18) into (C.17), the proof is completed.

C.4 Proof of Corollary 2

To prove Corollary 2, we first prove a key property of smooth functions. Let $F(\boldsymbol{w}^*)$ denote the optimal global loss, i.e., $F(\boldsymbol{w}^*)\leq F(\boldsymbol{w}),\forall\boldsymbol{w}$. Based on the L-smooth of $F(\boldsymbol{w})$, we have

$$\begin{aligned}
F(\boldsymbol{w}^*)&\leq F\left(\boldsymbol{w}-\frac{1}{L}\nabla F(\boldsymbol{w})\right)\\
&\leq F(\boldsymbol{w})-\left\langle\nabla F(\boldsymbol{w}),\frac{1}{L}\nabla F(\boldsymbol{w})\right\rangle+\frac{1}{2L}\|\nabla F(\boldsymbol{w})\|^2\\
&=F(\boldsymbol{w})-\frac{1}{2L}\|\nabla F(\boldsymbol{w})\|^2. \qquad (C.19)
\end{aligned}$$

By rearranging the above inequality, the global loss function $F(\boldsymbol{w})$ with L-smooth satisfies

$$\|\nabla F(\boldsymbol{w})\|^2\leq 2L\left(F(\boldsymbol{w})-F(\boldsymbol{w}^*)\right). \qquad (C.20)$$

Let $F(\boldsymbol{w}_{t+1})$ and $F(\boldsymbol{w}_t)$ in (C.17) subtract $F(\boldsymbol{w}^*)$, then utilizing the property of L-smooth in (C.20), we have

$$\begin{aligned}
&\mathbb{E}\left[F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}^*)\right] \\
&\leq (1 - \tilde{\eta}L + 6\tilde{\eta}L^2)\mathbb{E}\left[(F(\boldsymbol{w}_t) - F(\boldsymbol{w}^*))\right] + \frac{(\tilde{\eta} + 3\tilde{\eta}L)\,(\lambda - 1)}{\lambda}(2G^2 + \frac{\sigma^2}{\lambda}) \\
&+ \frac{(\tilde{\eta} + 1)\,\sigma^2}{2} + c\frac{1}{K}\sum_{k=1}^{K}(\tau_{k,t-1} + 1)^2\left(1 - \alpha_{k,t}\sum_{r=1}^{R} z_{k,t}^{(r)}\,\mathrm{Pr}(\mathrm{SINR}_{k,t}^{(r)} \geq \gamma_{\mathrm{th}})\right), \qquad \text{(C.21)}
\end{aligned}$$

By telescoping the above inequality, the proof is completed.

Appendix D

Proof in Chapter 5

D.1 Proof of Lemma 11

If $\lambda = 1$, the inequality is trivially satisfied since $\boldsymbol{w}_{k,t,0}^{(i)} - \boldsymbol{w}_{k,t}^{(i)}$. For $\lambda > 1$, we have

$$\begin{aligned}\mathbb{E}\|\boldsymbol{w}_{k,t,l}^{(i)} - \boldsymbol{w}_t^{(i)}\|^2 &= \mathbb{E}\left\|\boldsymbol{w}_{k,t,l-1}^{(i)} - \eta\nabla\tilde{F}_k(\boldsymbol{w}_{k,t,l-1}^{(i)}) - \boldsymbol{w}_t^{(i)}\right\|^2 \\ &\leq \mathbb{E}\left\|\boldsymbol{w}_{k,t,l-1}^{(i)} - \boldsymbol{w}_t^{(i)} - \eta\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})\right\|^2 + \eta^2\sigma^2, \qquad \text{(D.1)}\end{aligned}$$

where the inequality is by subtracting $\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})$ into $\nabla\tilde{F}_k(\boldsymbol{w}_{k,t,l-1}^{(i)}) - \boldsymbol{w}_t^{(i)}$, then using the triangle inequality and Assumption 12. Below we bound the first term in the above inequaltion as

$$\begin{aligned}&\mathbb{E}\left\|\boldsymbol{w}_{k,t,l-1}^{(i)} - \boldsymbol{w}_t^{(i)} - \eta\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})\right\|^2 \\ &= \mathbb{E}\|\boldsymbol{w}_{k,t,l-1}^{(i)} - \boldsymbol{w}_t^{(i)}\|^2 + \eta^2\mathbb{E}\|\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})\|^2 \\ &- 2\mathbb{E}\left\langle \frac{1}{\sqrt{\lambda-1}}(\boldsymbol{w}_{k,t,l-1}^{(i)} - \boldsymbol{w}_t^{(i)}), \eta\sqrt{\lambda-1}\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})\right\rangle \\ &\overset{(a)}{\leq} (1 + \frac{1}{\lambda-1})\mathbb{E}\left\|\boldsymbol{w}_{k,t,l-1}^{(i)} - \boldsymbol{w}_t^{(i)}\right\|^2 + \eta^2\lambda\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})\right\|^2 \\ &\leq (1 + \frac{1}{\lambda-1})\mathbb{E}\left\|\boldsymbol{w}_{k,t,l-1}^{(i)} - \boldsymbol{w}_t^{(i)}\right\|^2 + 2\eta^2\lambda\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{k,t}^{(i)})\right\|^2 \\ &+ 2\eta^2\lambda\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)}) - \nabla F_k(\boldsymbol{w}_{k,t}^{(i)})\right\|^2\end{aligned}$$

$$\overset{(b)}{\leq} (1 + \frac{1}{\lambda - 1} + 2\eta^2 \lambda L^2)\mathbb{E}\left\| \boldsymbol{w}^{(i)}_{k,t,l-1} - \boldsymbol{w}^{(i)}_{t} \right\|^2 + 2\eta^2 \lambda \mathbb{E}\left\| \nabla F_k(\boldsymbol{w}^{(i)}_{k,t}) \right\|^2$$

$$\overset{(c)}{\leq} (1 + \frac{3}{2(\lambda - 1)})\mathbb{E}\left\| \boldsymbol{w}^{(i)}_{k,t,l-1} - \boldsymbol{w}^{(i)}_{t} \right\|^2 + 2\eta^2 \lambda \mathbb{E}\left\| \nabla F_k(\boldsymbol{w}^{(i)}_{k,t}) \right\|^2, \tag{D.2}$$

where (a) comes from the triangle inequality, (b) comes from the L-smooth of loss functions, (c) is due to $\eta < \frac{1}{2\lambda L}$. Thus, we have

$$\mathbb{E}\left\| \boldsymbol{w}^{(i)}_{k,t,l} - \boldsymbol{w}^{(i)}_{t} \right\|^2 \leq (1 + \frac{3}{2(\lambda - 1)})\mathbb{E}\left\| \boldsymbol{w}^{(i)}_{k,t,l-1} - \boldsymbol{w}^{(i)}_{t} \right\|^2$$
$$+ 2\eta^2 \lambda \mathbb{E}\| \nabla F_k(\boldsymbol{w}^{(i)}_{k,t}) \|^2 + \eta^2\sigma^2. \tag{D.3}$$

By telescoping the above inequality, we have

$$\mathbb{E}\left\| \boldsymbol{w}^{(i)}_{k,t,l} - \boldsymbol{w}^{(i)}_{t} \right\|^2 \leq \left(2\eta^2 \lambda \mathbb{E}\| \nabla F_k(\boldsymbol{w}^{(i)}_{t}) \|^2 + \eta^2\sigma^2\right) \frac{(1 + \frac{3}{2(\lambda-1)})^{\lambda-1} - 1}{\frac{3}{2(\lambda-1)}}. \tag{D.4}$$

Since $(1 + \frac{3}{2(\lambda-1)})^{\lambda-1} = (1 + \frac{3}{2(\lambda-1)})^{\frac{2(\lambda-1)}{3}\frac{3}{2}} \leq e^{\frac{3}{2}} < 5$ and $\frac{2(\lambda-1)}{3} < \lambda - 1$, we have

$$\mathbb{E}\| \boldsymbol{w}^{(i)}_{k,t,l} - \boldsymbol{w}^{(i)}_{t} \|^2 \leq 4(\lambda - 1)(2\eta^2 \lambda G^2 + \eta^2\sigma^2). \tag{D.5}$$

By substituting the above inequality into the left-hand-side of (6.10), the proof is completed.

D.2 Proof of Lemma 12

For $t_1 > t_2$, we have

$$\mathbb{E}\left\| \boldsymbol{w}^{(i)}_{t_1} - \boldsymbol{w}^{(i)}_{t_2} \right\|^2 = \mathbb{E}\left\| \sum\nolimits_{t=t_2}^{t_1-1} (\boldsymbol{w}^{(i)}_{t+1} - \boldsymbol{w}^{(i)}_{t}) \right\|^2$$
$$= \tilde{\eta}^2 \mathbb{E}\left\| \sum\nolimits_{t=t_2}^{t_1-1} \frac{1}{K\lambda} \sum\nolimits_{k=1}^{K} \sum\nolimits_{l=0}^{\lambda-1} \nabla \tilde{F}_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}) \right\|^2$$
$$\overset{(a)}{\leq} 3\tilde{\eta}^2(t_1 - t_2) \sum_{t=t_2}^{t_1-1} \frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \mathbb{E}\left\| \nabla \tilde{F}_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}) - \nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}) \right\|^2$$
$$+ 3\tilde{\eta}^2(t_1 - t_2) \sum_{t=t_2}^{t_1-1} \frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \mathbb{E}\left\| \nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}) - \nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}) \right\|^2$$
$$+ 3\tilde{\eta}^2(t_1 - t_2) \sum\nolimits_{t=t_2}^{t_1-1} \mathbb{E}\left\| \nabla F(\boldsymbol{w}^{(i)}_{t-\tau^{(i)}_{k,t}}) \right\|^2$$

$$\overset{(b)}{\leq} 3\tilde{\eta}^2(t_1-t_2)\sum_{t=t_2}^{t_1-1}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}L^2\mathbb{E}\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\|^2$$
$$+3\tilde{\eta}^2(t_1-t_2)^2(\sigma^2+G^2), \tag{D.6}$$

where (a) is adding and subtracting $\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})$ and $\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})$ into $\nabla\tilde{F}_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})$, (b) is due to Assumption 12 and 13. Based on Lemma 11, by substituting (6.10) into the above inequality, we have

$$\mathbb{E}\left\|\boldsymbol{w}^{(i)}_t-\boldsymbol{w}^{(i)}_{t-\tau^{(i)}_{k,t}}\right\|^2\leq 3\eta^2(\tau^{(i)}_{k,t})^2\Big((\lambda^2+(\lambda-1)I)\sigma^2+(\lambda^2+2\lambda(\lambda-1)I)G^2\Big). \tag{D.7}$$

Substituting the above inequation into the left-hand-side of (6.11), the proof is completed.

D.3 Proof of Theorem 5

Using the L-smooth property of local loss function, we have $\mathbb{E}[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)]\leq\mathbb{E}\langle\nabla F(\boldsymbol{w}_t),\,\boldsymbol{w}_{t+1}-\boldsymbol{w}_t\rangle+\frac{L}{2}\mathbb{E}\left\|\boldsymbol{w}_{t+1}-\boldsymbol{w}_t\right\|^2$. It is worth mentioning that both the inner product and norm can be broken down and reformulated as the sum of inner products and norms over all parameter regions, respectively. Thus, we have

$$\mathbb{E}\left[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)\right]\leq\sum_{i=1}^{I}\mathbb{E}\left\langle\nabla F(\boldsymbol{w}^{(i)}_t),\boldsymbol{w}^{(i)}_{t+1}-\boldsymbol{w}^{(i)}_t\right\rangle+\frac{L}{2}\sum_{i=1}^{I}\mathbb{E}\|\boldsymbol{w}^{(i)}_{t+1}-\boldsymbol{w}^{(i)}_t\|^2$$
$$=-\tilde{\eta}\sum_{i=1}^{I}\mathbb{E}\Big\langle\nabla F(\boldsymbol{w}^{(i)}_t),\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla\tilde{F}_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big\rangle$$
$$+\frac{L}{2}\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla\tilde{F}_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\right\|^2$$
$$=\underbrace{-\tilde{\eta}\sum_{i=1}^{I}\mathbb{E}\Big\langle\nabla F(\boldsymbol{w}^{(i)}_t),\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big\rangle}_{A_1}$$
$$\underbrace{+\frac{L}{2}\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla\tilde{F}_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\right\|^2}_{A_2}$$

$$\underbrace{-\tilde{\eta}\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\Big\langle \nabla F(\boldsymbol{w}_t^{(i)}), \nabla\tilde{F}_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}) - \nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big\rangle}_{A_3}, \tag{D.8}$$

where the last step is derived by adding and subtracting $\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})$ into $\nabla\tilde{F}_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})$. Below we bound the three terms in (D.8). For A_1,

$$\begin{aligned}
A_1 &= -\tilde{\eta}\sum_{i=1}^{I}\mathbb{E}\Big\langle \nabla F(\boldsymbol{w}_t^{(i)}), \nabla F(\boldsymbol{w}_t^{(i)}) - \nabla F(\boldsymbol{w}_t^{(i)}) + \frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big\rangle \\
&\overset{(a)}{=} -\tilde{\eta}\mathbb{E}\left\|\nabla F(\boldsymbol{w}_t)\right\|^2 + \tilde{\eta}\sum_{i=1}^{I}\mathbb{E}\Big\langle \nabla F(\boldsymbol{w}_t^{(i)}), \nabla F(\boldsymbol{w}_t^{(i)}) - \frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big\rangle \\
&\overset{(b)}{\leq} -\frac{1}{2}\tilde{\eta}\mathbb{E}\left\|\nabla F(\boldsymbol{w}_t)\right\|^2 + \frac{1}{2}\tilde{\eta}\sum_{i=1}^{I}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\Big(\nabla F_k(\boldsymbol{w}_t^{(i)}) - \nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big)\right\|^2,
\end{aligned} \tag{D.9}$$

where (a) is due to $\|\nabla F(\boldsymbol{w}_t)\|^2 = \sum_{i=1}^{I}\|\nabla F(\boldsymbol{w}_t^{(i)})\|^2$, (b) follows the triangle inequality. For the last term on the RHS of (D.9), we have

$$\begin{aligned}
&\frac{1}{2}\tilde{\eta}\sum_{i=1}^{I}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\Big(\nabla F_k(\boldsymbol{w}_t^{(i)}) - \nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big)\right\|^2 \\
&\overset{(a)}{\leq} \frac{1}{2}\tilde{\eta}\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_t^{(i)}) - \nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\right\|^2 \\
&\overset{(b)}{\leq} \tilde{\eta}\sum\nolimits_{i=1}^{I}\frac{1}{K}\sum\nolimits_{k=1}^{K}\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_t^{(i)}) - \nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})\right\|^2 \\
&+ \tilde{\eta}\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}) - \nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\right\|^2 \\
&\overset{(c)}{\leq} \tilde{\eta}\sum_{i=1}^{I}\frac{1}{K}\sum_{k=1}^{K}L^2\mathbb{E}\left\|\boldsymbol{w}_t^{(i)} - \boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\right\|^2 \\
&+ \tilde{\eta}\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}L^2\mathbb{E}\left\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l} - \boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\right\|^2,
\end{aligned} \tag{D.10}$$

where (a) follows Jensen's inequality, (b) comes from adding and subtracting $\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})$ into $\nabla F_k(\boldsymbol{w}_t^{(i)})$, (c) is due to the L-smooth of loss functions in Assumption 10. Substituting (D.10) into (D.9), we have

$$A_1 \leq -\frac{1}{2}\tilde{\eta}\mathbb{E}\left\|\nabla F(\boldsymbol{w}_t)\right\|^2 + \tilde{\eta}L^2\sum_{i=1}^{I}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\boldsymbol{w}_t^{(i)} - \boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\right\|^2$$

$$+\,\tilde{\eta}L^2\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\left\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\right\|^2. \tag{D.11}$$

Now we focus on bounding A_2 as follows:

$$\begin{aligned}
A_2 &= \frac{L}{2}\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\nabla\tilde{F}_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})+\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\right)\right\|^2\\
&\overset{(a)}{\leq}\frac{L}{2}\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\right\|^2+\frac{L}{2}\tilde{\eta}^2\sigma^2\\
&\overset{(b)}{\leq} L\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\|\nabla F(\boldsymbol{w}^{(i)}_t)\|^2+\frac{L}{2}\tilde{\eta}^2\sigma^2\\
&\underbrace{+\,L\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t})\right)\right\|^2}_{A_{2,2}},
\end{aligned} \tag{D.12}$$

where (a) follows the triangle inequality and the bounded noise of SGD in Assumption 12, (b) is derived by adding and subtracting $\nabla F(\boldsymbol{w}^{(i)}_t)$ into $\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})$, then using the triangle inequality. Now we bound the second term on the RHS of (D.12) as

$$\begin{aligned}
A_{2,2} &\overset{(a)}{\leq} 2L\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\left\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\left(\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})\right)\right\|^2\\
&+2L\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\left\|\frac{1}{K}\sum_{k=1}^{K}\left(\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t})\right)\right\|^2\\
&\overset{(b)}{\leq} 2\tilde{\eta}^2L^3\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\left\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\right\|^2\\
&+2\tilde{\eta}^2L^3\sum_{i=1}^{I}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}-\boldsymbol{w}^{(i)}_t\|^2,
\end{aligned} \tag{D.13}$$

where (a) is derived by adding and subtracting $\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})$ into $\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})$, (b) follows Assumption 10. Thus, we have

$$\begin{aligned}
A_2 &\leq L\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\|\nabla F(\boldsymbol{w}^{(i)}_t)\|^2+2\tilde{\eta}^2L^3\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\left\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\right\|^2\\
&+2\tilde{\eta}^2L^3\sum_{i=1}^{I}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}-\boldsymbol{w}^{(i)}_t\|^2+\frac{L}{2}\tilde{\eta}^2\sigma^2.
\end{aligned} \tag{D.14}$$

For A_3, we have

$$A_3\overset{(a)}{=}-\tilde{\eta}\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\Big\langle\nabla F(\boldsymbol{w}^{(i)}_t)-\nabla F(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}),$$

$$
\begin{aligned}
&\nabla \tilde{F}_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}) - \nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big\rangle \\
&\overset{(b)}{\le} \frac{1}{2}\tilde{\eta} \sum\nolimits_{i=1}^{I} \frac{1}{K} \sum\nolimits_{k=1}^{K} \mathbb{E}\left\| \nabla F(\boldsymbol{w}^{(i)}_t) - \nabla F(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}) \right\|^2 + \frac{1}{2}\tilde{\eta}\sigma^2 \\
&\overset{(c)}{\le} \frac{1}{2}\tilde{\eta} L^2 \sum\nolimits_{i=1}^{I} \frac{1}{K} \sum\nolimits_{k=1}^{K} \mathbb{E}\left\| \boldsymbol{w}^{(i)}_t - \boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}} \right\|^2 + \frac{1}{2}\tilde{\eta}\sigma^2,
\end{aligned} \tag{D.15}
$$

where (a) is derived by adding and subtracting $\nabla F(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})$ into $\nabla F(\boldsymbol{w}^{(i)}_t)$, then using Assumption 12, (b) follow the triangle inequality and the bounded noise of SGD, (c) is due to the L-smooth of loss functions. Substituting (D.11), (D.14), (D.15) into (D.8), and let $\tilde{\eta} < \frac{1}{2L}$, we have

$$
\begin{aligned}
\mathbb{E}\left[F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}_t)\right] \le & \left(-\frac{1}{2}\tilde{\eta} + L\tilde{\eta}^2 \right) \mathbb{E}\left\|\nabla F(\boldsymbol{w}_t)\right\|^2 \\
& + \frac{5}{2}\tilde{\eta} L^2 \sum_{i=1}^{I} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left\| \boldsymbol{w}^{(i)}_t - \boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}} \right\|^2 \\
& + \frac{3}{4}\tilde{\eta}\sigma^2 + 2\tilde{\eta}L^2 \sum_{i=1}^{I} \frac{1}{K\lambda} \sum_{k=1}^{K} \sum_{l=0}^{\lambda-1} \mathbb{E}\left\| \boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l} - \boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}} \right\|^2 .
\end{aligned} \tag{D.16}
$$

Based on Lemma 11 and Lemma 12, by substituting (6.10) and (6.11) into the above inequality and assuming $\tilde{\eta} < \frac{1}{2L}$,

$$
\begin{aligned}
\mathbb{E}[F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}_t)] \le & \left(-\frac{1}{2}\eta + L\eta^2\lambda\right)\lambda \left\|\nabla F(\boldsymbol{w}_t)\right\|^2 + 4\eta(\lambda-1)IG^2 + \left(4\eta^2 L(\lambda-1)I + \frac{3}{4}\eta\lambda\right)\sigma^2 \\
& + \frac{15\eta^2 L}{4} \left(\lambda^2(\sigma^2 + G^2) + (\lambda-1)I(\sigma^2 + 2\lambda G^2) \right) \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{I} (\tau^{(i)}_{k,t})^2 .
\end{aligned} \tag{D.17}
$$

According to the evolution of the AoI of local gradients, we have

$$
(\tau^{(i)}_{k,t})^2 = (1 - \alpha_{k,t} m^{(i)}_{k,t})^2 (\tau^{(i)}_{k,t-1} + 1)^2 = (1 - \alpha_{k,t} m^{(i)}_{k,t})(\tau^{(i)}_{k,t-1} + 1)^2 . \tag{D.18}
$$

Substituting the (D.18) into (D.17), the proof is completed.

D.4 Proof of Corollary 3

By the μ-strongly convex of loss functions, we have $\|\nabla F(\boldsymbol{w}_t)\|^2 \ge 2\mu(F(\boldsymbol{w}_t) - F(\boldsymbol{w}^*))$. Substituting this inequality into (6.12), then adding and subtracting $F(\boldsymbol{w}^*)$ into (6.12), we have

$$\begin{aligned}\mathbb{E}[F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}^*)] \leq (1 - \eta\lambda\mu + 2L\eta^2\lambda^2\mu)\mathbb{E}[F(\boldsymbol{w}_t) - F(\boldsymbol{w}^*)] + c_1 \\ + \frac{15}{4}\eta^2 L c_2 \frac{1}{K}\sum\nolimits_{k=1}^{K}\sum\nolimits_{i=1}^{I}(\tau_{k,t}^{(i)})^2,\end{aligned} \tag{D.19}$$

where $c_1 = 4\eta(\lambda-1)IG^2 + \left(4\eta^2 L(\lambda-1)I + \frac{3}{4}\eta\lambda\right)\sigma^2$ and $c_2 = \lambda^2(\sigma^2+G^2)+(\lambda-1)I(\sigma^2+2\lambda G^2)$. By telescoping the above inequality, the proof is completed.

References

[1] H. Hellström, J. M. B. da Silva Jr, M. M. Amiri, M. Chen, V. Fodor, H. V. Poor, C. Fischione *et al.*, "Wireless for machine learning: A survey," *Foundations and Trends® in Signal Processing*, vol. 15, no. 4, pp. 290–399, 2022.

[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] X. Huang, K. Zhang, F. Wu, and S. Leng, "Collaborative machine learning for energy-efficient edge networks in 6G," *IEEE Netw.*, vol. 35, no. 6, pp. 12–19, 2021.

[4] 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.

[5] M. Goddard, "The EU general data protection regulation (GDPR): European regulation that has a global impact," *International Journal of Market Research*, vol. 59, no. 6, pp. 703–705, 2017.

[6] D. Ghosh, "What you need to know about californias new data privacy law," *Harvard Business Review*, vol. 11, 2018.

[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. Artificial Intelligence and Statistics (AISTATS)*, 20–22, Apr. 2017.

[8] F. Hanzely, B. Zhao, and M. Kolar, "Personalized federated learning: A unified

framework and universal optimization techniques," *arXiv preprint arXiv:2102.09743*, 2021.

[9] K. Pfeiffer, M. Rapp, R. Khalili, and J. Henkel, "Federated learning for computationally-constrained heterogeneous devices: A survey," *ACM Computing Surveys*, 2023.

[10] C. Yang, Q. Wang, M. Xu, Z. Chen, K. Bian, Y. Liu, and X. Liu, "Characterizing impacts of heterogeneity in federated learning upon large-scale smartphone data," in *Proc. the Web Conference 2021*, 2021, pp. 935–946.

[11] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[12] Z. Chen, W. Yi, Y. Liu, and A. Nallanathan, "Knowledge-aided federated learning for energy-limited wireless networks," *IEEE Trans. Commun.*, vol. 71, no. 6, pp. 3368–3386, 2023.

[13] P. S. Bouzinis, P. D. Diamantoulakis, and G. K. Karagiannidis, "Wireless quantized federated learning: A joint computation and communication design," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2756–2770, 2023.

[14] E. Ozfatura, K. Ozfatura, and D. Gündüz, "Time-correlated sparsification for communication-efficient federated learning," in *Proc. IEEE Int. Symposium on Info. Theory (ISIT)*. IEEE, 2021, pp. 461–466.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. and Pattern Recogn. (CVPR)*.

[16] Y. Li, W. Zhou, H. Wang, H. Mi, and T. M. Hospedales, "Fedh2l: Federated learning with model and statistical heterogeneity," *arXiv preprint arXiv:2101.11296*, 2021.

[17] W. Tong and G. Y. Li, "Nine challenges in artificial intelligence and wireless communications for 6g," *IEEE Wireless Commun.*, vol. 29, no. 4, pp. 140–145, 2022.

[18] S. Dutta, J. Wang, and G. Joshi, "Slow and stale gradients can win the race," *IEEE J. Sel. Areas in Inf. Theory*, vol. 2, no. 3, pp. 1012–1024, 2021.

[19] J. Liu, H. Xu, L. Wang, Y. Xu, C. Qian, J. Huang, and H. Huang, "Adaptive

asynchronous federated learning in resource-constrained edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 2, pp. 674–690, 2023.

[20] A. Li, J. Sun, X. Zeng, M. Zhang, H. Li, and Y. Chen, "Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking," in *Proc. ACM Conf. Embedded Networked Sensor Syst.*, 2021, pp. 42–55.

[21] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, 2021.

[22] Y. Xi, A. Burr, J. Wei, and D. Grace, "A general upper bound to evaluate packet error rate over quasi-static fading channels," *IEEE Trans. Wireless Commun.*, vol. 10, no. 5, pp. 1373–1377, 2011.

[23] Z. Chen, W. Yi, A. Nallanathan, and G. Y. Li, "Efficient wireless federated learning with partial model aggregation," *arXiv preprint arXiv:2204.09746*, 2022.

[24] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, 2021.

[25] M. Zhang, G. Zhu, S. Wang, J. Jiang, Q. Liao, C. Zhong, and S. Cui, "Communication-efficient federated edge learning via optimal probabilistic device scheduling," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8536–8551, 2022.

[26] E. Rizk, S. Vlaski, and A. H. Sayed, "Optimal importance sampling for federated learning," in *Proc. IEEE ICASSP*, 2021.

[27] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, 2021.

[28] Y. Jee Cho, J. Wang, and G. Joshi, "Towards understanding biased client selection in federated learning," in *Proc Artificial Intelligence and Statistics (AISTATS)*, 28–30 Mar 2022.

[29] I. Mohammed, S. Tabatabai, A. Al-Fuqaha, F. E. Bouanani, J. Qadir, B. Qolomany, and M. Guizani, "Budgeted online selection of candidate iot clients to par-

ticipate in federated learning," *IEEE Internet of Things J.*, vol. 8, no. 7, pp. 5938–5952, 2021.

[30] A. Tak, Z. Mlika, and S. Cherkaoui, "Data-aware device scheduling for federated edge learning," *IEEE Trans. Cognitive Commun. and Netw.*, vol. 8, no. 1, pp. 408–421, 2022.

[31] 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 Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3643–3658, 2021.

[32] J. Leng, Z. Lin, M. Ding, P. Wang, D. Smith, and B. Vucetic, "Client scheduling in wireless federated learning based on channel and learning qualities," *IEEE Wireless Commun. Letters*, vol. 11, no. 4, pp. 732–735, 2022.

[33] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, 2021.

[34] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7947–7962, 2021.

[35] 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 Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, 2021.

[36] S. Yue, J. Ren, J. Xin, D. Zhang, Y. Zhang, and W. Zhuang, "Efficient federated meta-learning over multi-access wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1556–1570, 2022.

[37] K. Guo, Z. Chen, H. H. Yang, and T. Q. S. Quek, "Dynamic scheduling for heterogeneous federated learning in private 5g edge networks," *IEEE J. Sel. Topics in Signal Process.*, vol. 16, no. 1, pp. 26–40, 2022.

[38] W. Shi, Y. Sun, S. Zhou, and Z. Niu, "Device scheduling and resource allocation for federated learning under delay and energy constraints," in *Proc.IEEE Int. Workshop on Signal Process. Advances in Wireless Commun. (SPAWC)*, 2021, pp. 596–600.

[39] C. Bucilu, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, 2006, pp. 535–541.

[40] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[41] 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.

[42] J. Zhang, S. Guo, X. Ma, H. Wang, W. Xu, and F. Wu, "Parameterized knowledge transfer for personalized federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 092–10 104, 2021.

[43] D. Yao, W. Pan, Y. Dai, Y. Wan, X. Ding, H. Jin, Z. Xu, and L. Sun, "Local-global knowledge distillation in heterogeneous federated learning with non-iid data," *arXiv preprint arXiv:2107.00051*, 2021.

[44] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Communication-efficient federated distillation with active data sampling," in *Proc. IEEE Int. Conf. Commun.(ICC)*, 2022.

[45] 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 Commun. Letters*, vol. 24, no. 10, pp. 2211–2215, 2020.

[46] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc Int. Conf. Machine Learning (ICML)*, 18–24 Jul, 2021.

[47] J.-H. Ahn, O. Simeone, and J. Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," in *Proc. IEEE Annual Int. Symp. on Personal, Indoor and Mobile Radio Commun. (PIMRC)*, 2019, pp. 1–6.

[48] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. Neural Information Processing Systems (NeurIPS)*, 2020.

[49] 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.

[50] T. Yu, E. Bagdasaryan, and V. Shmatikov, "Salvaging federated learning by local adaptation," *arXiv preprint arXiv:2002.04758*, 2020.

[51] Y. Wang, Y. Xu, Q. Shi, and T.-H. Chang, "Quantized federated learning under transmission delay and outage constraints," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 323–341, 2022.

[52] H. Chen, S. Huang, D. Zhang, M. Xiao, M. Skoglund, and H. V. Poor, "Federated learning over wireless iot networks with optimized communication and resources," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16 592–16 605, 2022.

[53] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, 2020.

[54] M. Salehi and E. Hossain, "Federated learning in unreliable and resource-constrained cellular wireless networks," *IEEE Trans. Commun.*, vol. 69, no. 8, pp. 5136–5151, 2021.

[55] H. Hellström, V. Fodor, and C. Fischione, "Over-the-air federated learning with retransmissions," in *Proc Int. Workshop on Signal Process. Advances in Wireless Commun. (SPAWC)*, 2021, pp. 291–295.

[56] X. Su, Y. Zhou, L. Cui, and J. Liu, "On model transmission strategies in federated learning with lossy communications," *IEEE Trans. Parallel and Distributed Systems*, vol. 34, no. 4, pp. 1173–1185, 2023.

[57] H. Ye, L. Liang, and G. Y. Li, "Decentralized federated learning with unreliable communications," *IEEE J. Sel. Topics in Signal Process.*, pp. 1–1, 2022.

[58] M. Shirvanimoghaddam, A. Salari, Y. Gao, and A. Guha, "Federated learning with erroneous communication links," *IEEE Commun. Letters*, vol. 26, no. 6, pp. 1293–1297, 2022.

[59] 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 Trans. Commun.*, vol. 70, no. 1, pp. 231–244, 2022.

[60] D. Wen, K.-J. Jeon, and K. Huang, "Federated dropout-a simple approach for

enabling federated learning on resource constrained devices," *IEEE Wireless Commun. Letters*, vol. 11, no. 5, pp. 923–927, 2022.

[61] S. Liu, G. Yu, R. Yin, and J. Yuan, "Adaptive network pruning for wireless federated learning," *IEEE Wireless Commun. Letters*, vol. 10, no. 7, pp. 1572–1576, 2021.

[62] S. Bibikar, H. Vikalo, Z. Wang, and X. Chen, "Federated dynamic sparse training: Computing less, communicating less, yet learning better," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 36, no. 6, 2022, pp. 6080–6088.

[63] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," *arXiv preprint arXiv:1812.07210*, 2018.

[64] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," *arXiv preprint arXiv:2010.01264*, 2020.

[65] 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 Trans. Neural Networks and Learning Systems*, pp. 1–13, 2022.

[66] M. Kim, S. Yu, S. Kim, and S.-M. Moon, "Depthfl: Depthwise federated learning for heterogeneous clients," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2022.

[67] S. Horvath, S. Laskaridis, M. Almeida, I. Leontiadis, S. Venieris, and N. Lane, "Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 876–12 889, 2021.

[68] A.-J. Farcas, X. Chen, Z. Wang, and R. Marculescu, "Model elasticity for hardware heterogeneity in federated learning systems," in *Proc. ACM Workshop on Data Privacy and Federated Learning Technologies for Mobile Edge Network*, 2022, pp. 19–24.

[69] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.

[70] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2017.

[71] F. Ilhan, G. Su, and L. Liu, "Scalefl: Resource-adaptive federated learning with heterogeneous clients," in *Proc. IEEE/CVF Conf. Comput. Vis. and Pattern Recogn. (CVPR)*, June 2023, pp. 24 532–24 541.

[72] P. Li, G. Cheng, X. Huang, J. Kang, R. Yu, Y. Wu, and M. Pan, "Anycostfl: Efficient on-demand federated learning over heterogeneous edge devices," *arXiv preprint arXiv:2301.03062*, 2023.

[73] S. Lin, R. Ji, Y. Li, C. Deng, and X. Li, "Toward compact convnets via structure-sparsity regularized filter pruning," *IEEE Trans. Neural Networks and Learning Systems*, vol. 31, no. 2, pp. 574–588, 2020.

[74] Y. Mei, P. Guo, M. Zhou, and V. Patel, "Resource-adaptive federated learning with all-in-one neural composition," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 4270–4284.

[75] Ş. Cobzaş, R. Miculescu, A. Nicolae *et al.*, *Lipschitz functions.* Springer, 2019.

[76] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex analysis and optimization.* Athena Scientific, 2003, vol. 1.

[77] A. L. B. Cauchy, *Cours d'analyse de l'École Royale Polytechnique.* Imprimerie royale, 1821.

[78] A. Auslender and M. Teboulle, "Lagrangian duality and related multiplier methods for variational inequality problems," *SIAM Journal on Optimization*, vol. 10, no. 4, pp. 1097–1115, 2000.

[79] F. M. Dekking, *A Modern Introduction to Probability and Statistics: Understanding why and how.* Springer Science & Business Media, 2005.

[80] J. M. Steele, *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities.* Cambridge University Press, 2004.

[81] W. H. Young, "On classes of summable functions and their fourier series," *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 87, no. 594, pp. 225–229, 1912.

[82] G. Zhu, Y. Du, D. Gündüz, and K. Huang, "One-bit over-the-air aggregation for

communication-efficient federated edge learning: Design and convergence analysis," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2120–2135, 2021.

[83] K. B. Moses, "Mobile communication evolution," *International Journal of Modern Education and Computer Science*, vol. 6, no. 1, p. 25, 2014.

[84] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4734–4746, 2020.

[85] M. Gecer and B. Garbinato, "Federated learning for mobility applications," *ACM Computing Surveys*, vol. 56, no. 5, pp. 1–28, 2024.

[86] M. P. Friedlander and M. Schmidt, "Hybrid deterministic-stochastic methods for data fitting," *SIAM J. Sci. Comput.*, vol. 34, no. 3, pp. A1380–A1405, 2012.

[87] E. Abbasnejad, J. Shi, and A. van den Hengel, "Deep Lipschitz networks and dudley GANs," 2018. [Online]. Available: https://openreview.net/forum?id=rkw-jlb0W

[88] A. Virmaux and K. Scaman, "Lipschitz regularity of deep neural networks: analysis and efficient estimation," in *Proc. Neural Information Processing Systems (NeurIPS)*, 2018.

[89] A. Dedieu, "Improved error rates for sparse (group) learning with lipschitz loss functions," *arXiv preprint arXiv:1910.08880*, 2019.

[90] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *arXiv preprint arXiv:2010.13723*, 2020.

[91] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690–7703, 2020.

[92] Y. Sun, S. Zhou, Z. Niu, and D. Gündüz, "Dynamic scheduling for over-the-air federated edge learning with energy constraints," *IEEE J. Sel. Areas in Commun.*, vol. 40, no. 1, pp. 227–242, 2022.

[93] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[94] A. Rafiey and Y. Yoshida, "Fast and private submodular and k-submodular func-

tions maximization with matroid constraints," in *Proc Int. Conf. Machine Learning (ICML)*, 13–18 Jul 2020.

[95] A. Krause and D. Golovin, "Submodular function maximization." *Tractability*, vol. 3, pp. 71–104, 2014.

[96] X. Pan, S. Jegelka, J. E. Gonzalez, J. K. Bradley, and M. I. Jordan, "Parallel double greedy submodular maximization," in *Proc. Neural Information Processing Systems (NeurIPS)*, 2014.

[97] N. Buchbinder and M. Feldman, "Submodular functions maximization problems," in *Handbook of Approximation Algorithms and Metaheuristics, Second Edition.* Chapman and Hall/CRC, 2018, pp. 753–788.

[98] Z. Chen, W. Yi, A. S. Alam, and A. Nallanathan, "Dynamic task software caching-assisted computation offloading for multi-access edge computing," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6950–6965, 2022.

[99] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2019, pp. 691–706.

[100] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proc. ACM SIGSAC conf. comput. and commun. secur.*, 2017, pp. 603–618.

[101] S. Chen, C. Shen, L. Zhang, and Y. Tang, "Dynamic aggregation for heterogeneous quantization in federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6804–6819, 2021.

[102] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020.

[103] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[104] Q. Ma, Y. Xu, H. Xu, Z. Jiang, L. Huang, and H. Huang, "Fedsa: A semi-asynchronous federated learning mechanism in heterogeneous edge computing," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3654–3672, 2021.

[105] Z. Chen, Z. Zhou, and C. Chen, "Code caching-assisted computation offloading and resource allocation for multi-user mobile edge computing," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4517–4530, 2021.

[106] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[107] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *Proc Int. Conf. Machine Learning (ICML)*, 18–24 Jul 2021.

[108] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," *arXiv preprint arXiv:2003.13461*, 2020.

[109] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[110] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, 2020.

[111] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, 2021.

[112] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-IID data," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020.

[113] M. B. Cohen, Y. T. Lee, and Z. Song, "Solving linear programs in the current matrix multiplication time," *Journal of the ACM (JACM)*, vol. 68, no. 1, pp. 1–39, 2021.

[114] A. Schrijver *et al.*, *Combinatorial optimization: polyhedra and efficiency.* Berlin Germany:Springer-Velag, 2003.

[115] Z. Chen, W. Yi, and A. Nallanathan, "Exploring representativity in device scheduling for wireless federated learning," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2023.

[116] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Machine Learning and Systems*, 2020.

[117] Z. Chen, W. Yi, Y. Deng, and A. Nallanathan, "Device scheduling for wireless federated learning with latency and representativity," in *Proc. Int. Conf. Electrical, Computer, Commun. and Mechatronics Engineering (ICECCME)*, 2022, pp. 1–6.

[118] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," in *Proc. Int. Conf. Learning Representations (ICLR)*, April, 2017.

[119] M. B. Cohen, Y. T. Lee, and Z. Song, "Solving linear programs in the current matrix multiplication time," *J. ACM*, vol. 68, no. 1, jan 2021.

[120] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems.* Curran Associates, Inc., 2015.

[121] V.-D. Nguyen, S. K. Sharma, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Efficient federated learning algorithm for resource allocation in wireless iot networks," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3394–3409, 2021.

[122] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, "Splitfed: When federated learning meets split learning," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8485–8493.

[123] J. Wang, H. Qi, A. S. Rawat, S. Reddi, S. Waghmare, F. X. Yu, and G. Joshi, "Fedlite: A scalable approach for federated learning on resource-constrained clients," *arXiv preprint arXiv:2201.11865*, 2022.

[124] T. Feng, D. Bose, T. Zhang, R. Hebbar, A. Ramakrishna, R. Gupta, M. Zhang, S. Avestimehr, and S. Narayanan, "Fedmultimodal: A benchmark for multimodal federated learning," *arXiv preprint arXiv:2306.09486*, 2023.

[125] J. Chen and A. Zhang, "Fedmsplit: Correlation-adaptive federated multi-task learning across multimodal split networks," in *Proc. ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, 2022, pp. 87–96.

[126] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[127] Z. Li and D. Hoiem, “Learning without forgetting,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2018.

[128] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, “End-to-end incremental learning,” in *Proc. European conf. on computer vision (ECCV)*, 2018, pp. 233–248.

[129] D. Maltoni and V. Lomonaco, “Continuous learning in single-incremental-task scenarios,” *Neural Networks*, vol. 116, pp. 56–73, 2019.