Federated Data Analytics: Theory and Application

by

Xubo Yue

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in the University of Michigan
2023

Doctoral Committee:

Assistant Professor Raed Al Kontar, Chair
Professor Brian Denton
Professor Edward Ionides
Professor Judy Jin

Xubo Yue

maxyxb@umich.edu

ORCID iD: 0000-0001-9929-8895

Dedication

I would like to dedicate this dissertation to my beloved parents,
Xin Yue and Qi Xu

Acknowledgments

This acknowledgment expresses my utmost appreciation to all who have accompanied and supported me throughout my doctoral journey.

I want to sincerely thank my advisor, Dr. Raed Al Kontar, for his exceptional guidance, unwavering support, and remarkable patience throughout my doctoral studies. His invaluable contributions have significantly influenced and propelled my research forward. I am truly grateful to Dr. Kontar for his dedicated commitment to advising me, as it has played a pivotal role in helping me achieve this esteemed milestone in my doctoral journey. I consider myself fortunate to have Dr. Kontar as my mentor, and I am indebted to him for the positive impact he has had on both my academic and personal development.

I would also like to thank my committee members, Dr. Denton, Dr. Jin, and Dr. Ionides, for their valuable insights, time, and contribution to part of this research. Their generous support and constructive criticism have been a constant source of inspiration for me.

I also want to thank Seokhyun, Rohan, Haoming, Daniel, Kam, and other IOE friends who made my doctoral journey more fruitful and memorable. We have walked through hard but enjoyable and exciting times together.

My special thanks go to the friends I met in Ann Arbor. We have shared many enjoyable moments, like travels, parties, and visiting beautiful places around Michigan. We have created an unforgettable memory.

Finally, I thank my family. Thank you, mother and father, for your endless support and unconditional love throughout the past five years. I am grateful for the values and life lessons you have instilled in me, which have shaped my character and contributed to my success. Thank you, my beloved grandparents. Your love and support have been an enduring blessing in my life, and I cannot thank you enough for everything you have done for me.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF APPENDICES

LIST OF ALGORITHMS

ALGORITHM

ABSTRACT

A critical change is happening in today's Internet of Things (IoT). The computational power of edge devices is steadily increasing. AI chips are rapidly infiltrating the market, smart phones nowadays have compute power comparable to everyday use laptops, Tesla just boasted that its autopilot system has computing power of more than 3000 MacBook pros and small local computers such as Raspberry Pis have become common place in many applications especially manufacturing. This opens a new paradigm for data analytics in IoT; one that exploits local computing power to process more of the user's data where it is created. This new paradigm is coined as Federated Data Analytics (FDA).

I envision that many modern engineering systems are on the verge of shifting from a centralized regime to a distributed, smart and connected system where some of the data processing is deferred to the edge. This report sets research objectives to develop federated data analytics methodologies and working prototypes that may help facilitate this transition and allow practitioners to fully reap its benefits. While promising, FDA faces several fundamental challenges.

1. Edge devices with few data, limited bandwidth/memory, or unreliable connection may not be favored by conventional distributed learning algorithms. As a result, such device(s) can potentially incur higher error rate(s). This vicious cycle often relinquishes the opportunity of these devices to significantly contribute to the training process. Besides this aforementioned notion of individual fairness, group fairness also deserves attention. As FDA penetrates practical applications, it is important to achieve fair performance across groups of clients characterized by their gender, ethnicity, socio-economic status, geographic location, etc. Despite the importance of fairness, very limited work exists along this line. The key challenge is that, unlike traditional definitions such as demographic disparity or equal opportunity, there is no clear notion of an outcome that is "good" for a device in the FDA scenario.

2. In spite of some recent advances in FDA, most, if not all, literature focuses on deep neural networks and their applications for mobile networks. However, neural networks are not a panacea for all engineering problems. To date, very few papers have delivered a federated treatment of models that go beyond deep learning. Questions such as variable selection,

uncertainty quantification, hypothesis testing, and incorporating domain expert knowledge remain unanswered in FDA.

3. In engineering systems, data across devices possess trends that are often heterogeneous. For example, vehicles/machines operated under different environments typically yield heterogeneous condition monitoring data. As such, learning a global model, “one model that fits all”, may easily fail. Unfortunately, the vast majority of FDA work falls in this category due to the focus on mobile applications. Indeed some recent literature shows that global modeling approaches fail to provide reasonable predictions or classifications when heterogeneity exists.

This report develops three data analytics frameworks that solve the challenges above, with application to quality and reliability engineering. *(i) Developing FDA algorithms that target fairness*: the proposed framework – GIFAIR - imposes group and individual fairness to the FDA setting. By adding a regularization term, GIFAIR penalizes the spread in the loss of client groups to drive the optimizer to a fair solution. *(ii) Developing FDA algorithms that go beyond deep learning*: the proposed approach extends linear models and sparse linear models to federated scenarios, presenting solutions for hypothesis testing, uncertainty quantification, variable selection, and deriving engineering insight. *(iii) Tackling data heterogeneity through personalization*: the proposed approach builds personalized Bayesian models that tackle data heterogeneity among different devices. Furthermore, it also provides the first theoretical results on FDA convergence in correlated settings, which is very common in engineering situations. In turn, this may help researchers further investigate FDA within alternative stochastic processes built upon correlations, such as Lévy processes.

CHAPTER 1

Introduction

1.1 Motivation and Research Objectives

Nowadays, the sheer amount of data collected from edge devices such as mobile phones and self-driving vehicles is beginning to overwhelm traditional centralized data analytics regimes where data from the edge is continuously uploaded to a central server to be processed. Excessive communication traffic from data upload, significant central server storage needs, energy expenditures from centralized learning of big data models, and privacy concerns from sharing raw data are becoming critical challenges in centralized systems. Statista predicted that, by 2024, data produced on edge devices (e.g., cell phone data, self-driving vehicle data) would reach more than hundreds of zettabytes while the global central servers only have 10.4 zettabytes of storage (Morell and Alba 2022). Transmitting such a vast amount of edge data into a central server is infeasible. Adding to that, training a model with moderately large datasets results in significant budget costs and carbon emissions (Patterson et al. 2021). Furthermore, data-sharing comes with serious privacy concerns. According to Lawson et al. (2015), Canadian drivers who refused to enroll in the automotive telematics program demanded that their personal driving data (e.g., behavior, location, web-browsing history) should be respected by vehicle companies and that they be given control over the data collection process. These debates over data protection standards have not faded away over the past few years.

Fortunately, a critical change is happening in today's Internet of Things (IoT). The processing and computational power of edge devices is becoming increasingly powerful. AI chips are rapidly infiltrating the global market. Today's flagship cell phones are more powerful than many laptops, and Tesla has boasted that the computer that runs its Autopilot system is as powerful as hundreds of MacBook Pros. As such, we now have the opportunity to process more of our data where it is created - i.e. at the edge. This decentralized data analytics paradigm is often coined as **federated data analytics (FDA) or Federated Learning (FL)**. FDA resolves many of the aforementioned drawbacks. By exploiting edge computations, one can parallelize inference, reduce storage and communication costs, achieve faster alerts and decisions, and protect privacy, amongst many others.

I envision that many modern engineering systems are on the verge of shifting from a centralized regime to a distributed, smart and connected system where some of the data processing is deferred to the edge. My research focus is to develop methodologies and working prototypes that may help facilitate this transition and allow practitioners to fully reap its benefits. This report focuses on the development of federated and distributed data analytics for IoT-enabled systems.

While promising, FDA faces critical yet intrinsic challenges that need to be solved. First, in spite of some recent advances in FDA, most, if not all, literature focuses on deep neural networks and their applications for mobile networks. However, neural networks are not a panacea for all engineering problems. To date, very few papers have delivered a federated treatment of models that go beyond deep learning. Questions such as variable selection, uncertainty quantification, hypothesis testing, and incorporating domain expert knowledge remain unanswered in FDA. Second, edge devices with few data, limited bandwidth/memory, or unreliable connection may not be favored by conventional distributed learning algorithms. As a result, such device(s) can potentially incur higher error rate(s). This vicious cycle often relinquishes the opportunity of these devices to significantly contribute in the training process. Besides this aforementioned notion of individual fairness, group fairness also deserves attention. As FDA penetrates practical applications, it is important to achieve fair performance across groups of clients characterized by their gender, ethnicity, socio-economic status, geographic location, etc. Despite the importance of fairness, very limited work exists along this line. The key challenge is that, unlike traditional definitions such as demographic disparity or equal opportunity, there is no clear notion of an outcome which is “good” for a device in the FDA scenario. Third, in engineering systems, data across devices possess trends that are often heterogeneous. For example, vehicles/machines operated under different environments typically yield heterogeneous condition monitoring data. As such, learning a global model, “one model that fits all”, may easily fail. Unfortunately, the vast majority of FDA work fall in this category due to the focus on mobile applications. Indeed some recent literature shows that global modeling approaches fail to provide reasonable predictions or classifications when heterogeneity exists (Yu et al. 2020a, Zhu et al. 2021a).

To truly accomplish the promise of IoT-enabled systems, these challenges should be adequately addressed. This report is committed to developing and designing novel federated and distributed data analytics solutions that address the aforementioned challenges. To this end, the research objectives of this dissertation are set as follows.

1.2 Outline of Dissertation

This dissertation aims to achieve the above research objectives to address crucial challenges arising when developing data analytics frameworks for smart & connected systems. Here the dissertation is outlined as follows.

Chapter 2 – GIFAIR-FL: A Framework for Group and Individual Fairness in Federated Learning In this chapter, we propose GIFAIR-FL: a framework that imposes **G**roup and **I**ndividual **FAIR**ness to **F**ederated **L**earning settings. By adding a regularization term, our algorithm penalizes the spread in the loss of client groups to drive the optimizer to fair solutions. Our framework GIFAIR-FL can accommodate both global and personalized settings. Theoretically, we show convergence in non-convex and strongly convex settings. Our convergence guarantees hold for both $i.i.d.$ and non-$i.i.d.$ data. To demonstrate the empirical performance of our algorithm, we apply our method to image classification and text prediction tasks. Compared to existing algorithms, our method shows improved fairness results while retaining superior or similar prediction accuracy.

Chapter 3 – Federated Data Analytics: A Study on Linear Models Despite the recent success stories of FDA, most literature focuses exclusively on deep neural networks. In this work, we take a step back to develop an FDA treatment for one of the most fundamental statistical models: linear regression. Our treatment is built upon hierarchical modeling that allows borrowing strength across multiple groups. To this end, we propose two federated hierarchical model structures that provide a shared representation across devices to facilitate information sharing. Notably, our proposed frameworks are capable of providing uncertainty quantification, variable selection, hypothesis testing, and fast adaptation to new unseen data. We validate our methods on a range of real-life applications, including condition monitoring for aircraft engines. The results show that our FDA treatment for linear models can serve as a competing benchmark model for the future development of federated algorithms.

Chapter 4 – Federated Gaussian Process: Convergence, Automatic Personalization and Multi-fidelity Modeling In this chapter, we propose FGPR: a Federated Gaussian process ($\mathcal{GP}$) regression framework that uses an averaging strategy for model aggregation and stochastic gradient descent for local computations. Notably, the resulting global model excels in personalization as FGPR jointly learns a shared prior across all devices. The predictive posterior then is obtained by exploiting this shared prior and conditioning on local data, which encodes personalized features from a specific dataset. Theoretically, we show that FGPR converges to a critical point of the full log-marginal likelihood function, subject to statistical errors. This result offers standalone value as it brings federated learning theoretical results to correlated paradigms. Through extensive case studies, we show that FGPR excels in a wide range of applications and is a promising approach for privacy-preserving multi-fidelity data modeling.

Finally, Chapter 5 concludes this report by summarizing the unique contributions of the completed studies and introducing possible future directions.

For the sake of brevity, the notations and abbreviations defined in a particular chapter are only applicable to that chapter and its corresponding Appendix.

CHAPTER 2

GIFAIR-FL: A Framework for Group and Individual Fairness in Federated Learning

In this chapter, we propose `GIFAIR-FL`: a framework that imposes **G**roup and **I**ndividual **FAIR**ness to **F**ederated **L**earning settings. By adding a regularization term, our algorithm penalizes the spread in the loss of client groups to drive the optimizer to fair solutions. Our framework `GIFAIR-FL` can accommodate both global and personalized settings. Theoretically, we show convergence in non-convex and strongly convex settings. Our convergence guarantees hold for both $i.i.d.$ and non-$i.i.d.$ data. To demonstrate the empirical performance of our algorithm, we apply our method to image classification and text prediction tasks. Compared to existing algorithms, our method shows improved fairness results while retaining superior or similar prediction accuracy.

2.1 Introductory Remarks

A critical change is happening in today's Internet of Things (IoT). The computational power of edge devices is steadily increasing. AI chips are rapidly infiltrating the market, smart phones nowadays have compute power comparable to everyday use laptops (Samsung 2019), Tesla just boasted that its autopilot system has computing power of more than 3000 MacBook pros (CleanTechnica 2021) and small local computers such as Raspberry Pis have become common place in many applications especially manufacturing (Al-Ali et al. 2018). This opens a new paradigm for data analytics in IoT; one that exploits local computing power to process more of the user's data where it is created. This future of IoT has been recently termed as the "The Internet of Federated Things (IoFT)" (Kontar et al. 2021) where the term federated, refers to some autonomy for IoT devices and is inspired by the explosive recent interest in federated data science.

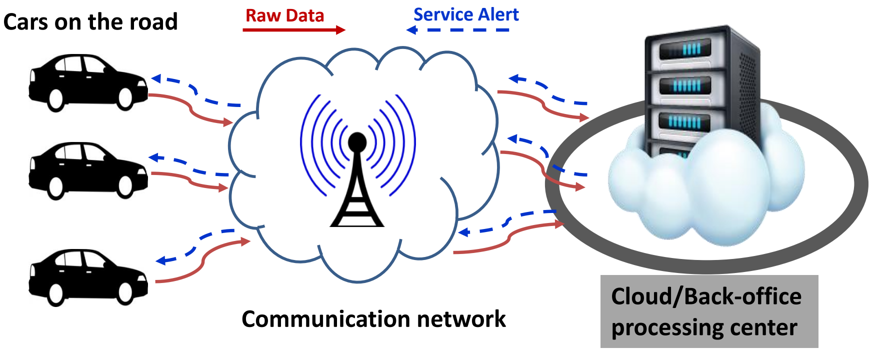

Figure 2.1: Example of the traditional IoT-enabled System

To give a microcosm of current IoT and its future, consider the IoT teleservice system shown in Fig. 2.1. Vehicles enrolled in this tele-service system often have their data in the form of condition monitoring signals uploaded to the cloud at regular intervals. The cloud acts as a data processing center that analyzes data for continuous improvement and to keep drivers informed about the health of their vehicles. IoT companies and services, such as Ford's SYNC and General Motors Onstar services, have long adopted this centralized approach to IoT. However, this state where data is amassed on the cloud yields significant challenges. The need to upload large amounts of data to the cloud incurs high communication and storage costs, demands large internet bandwidth (Jiang et al. 2020a, Yang et al. 2020), and leads to latency in deployment as well as reliability risks due to unreliable connection (Zhang et al. 2020a). Further, such a system does not foster trust or privacy as users need to share their raw data which is often sensitive or confidential (Li et al. 2020a).

With the increasing computational power of edge devices, the discussed challenges can be circumvented by moving part of the model learning to the edge. More specifically, rather than processing the data at the cloud, each device performs small local computations and only shares the minimum information needed to allow devices to borrow strength from each other and collaboratively extract knowledge to build smart analytics. In turn, such an approach (i) improves privacy as raw data is never shared, (ii) reduces cost and storage needs as less information is transmitted, (iii) enables learning parallelization and (iv) reduces latency in decisions as many decisions can now be achieved locally. Hereon we will use edge device and client interchangeably, also, the cloud or data processing center is referred to as the central server.

This idea of exploiting the computational power of edge devices by locally training models without recourse to data sharing gave rise to federated learning (FL). In particular, FL is a data analytics approach that allows distributed model learning without access to private data. Although the main concept of FL dates back a while ago, it was brought to the forefront of data science in 2017 by a team at Google which proposed Federated Averaging (`FedAvg`) (McMahan et al.

2017). In `FedAvg`, a central server distributes the model architecture (e.g., neural network, linear model) and a global model parameter (e.g., model weights) to selected devices. Devices run local computations to update model parameters and send updated parameters to the server. The server then takes a weighted average of the resulting local parameters to update the global model. This whole process is termed as one communication round and the process is iterated over multiple rounds until an exit condition is met. Figure 2.2 provides one illustrative example of `FedAvg`. Since then FL has seen immense success in various fields such as text prediction (Hard et al. 2018, Ramaswamy et al. 2019), Bayesian optimization (Dai et al. 2020, Khodak et al. 2021), Multi-fidelity modeling (Yue and Kontar 2021), environment monitoring (Hu et al. 2018, Jiang et al. 2020b) and healthcare (Li et al. 2020a, Xu et al. 2021).

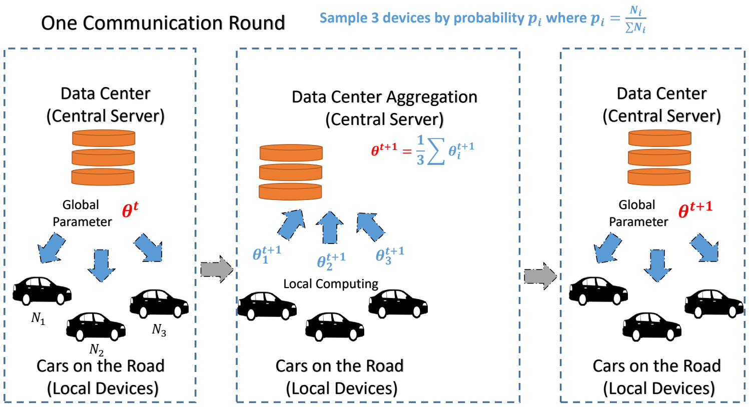

Figure 2.2: Illustrative Example of FL with `FedAvg`

Over the last few years, literature has been proposed to improve the performance of FL algorithms; be it speeding up FL algorithms to enable faster convergence (Karimireddy et al. 2020, Yuan and Ma 2020, Nguyen et al. 2020), tackling heterogeneous data both in size and distribution (Zhao et al. 2018, Li et al. 2018a, Sattler et al. 2019, Ghosh et al. 2019, Li and Wang 2019, Shi et al. 2023a,b), improving the parameter aggregation strategies at the central server (Pillutla et al. 2019, Wang et al. 2020a), designing personalized FL algorithms (Jiang et al. 2019, Fallah et al. 2020, Mansour et al. 2020), protecting federated systems from adversarial attacks (Bhagoji et al. 2019, Wang et al. 2020b), and promoting fairness (Mohri et al. 2019, Li et al. 2019a, Du et al. 2020, Hu et al. 2020, Huang et al. 2020, Zhang et al. 2020b). Please refer to Kontar et al. (2021) for a detailed literature review. Among those advances, fairness is a critical yet under-investigated area.

In the training phase of FL algorithms, devices with few data, limited bandwidth/memory, or unreliable connection may not be favored by conventional FL algorithms. For instance, as shown in Figure 2.2, `FedAvg` samples devices using the weight coefficient p_k proportional to the sample size on the device k. Scant data on device k will render p_k insignificant and this device less favorable by the resulting global model. As a result, such device(s) can potentially incur higher error rate(s).

This vicious cycle often relinquishes the opportunity of these devices to significantly contribute in the training process. Indeed, many recent papers have shown the large variety in model performance across devices under FL (Jiang et al. 2019, Hard et al. 2018, Wang et al. 2019, Smith et al. 2017, Kairouz et al. 2019), with some clients showing extremely bad model performance. Besides this aforementioned notion of individual fairness, group fairness also deserves attention in FL. As FL penetrates practical applications, it is important to achieve fair performance across groups of clients characterized by their gender, ethnicity, socio-economic status, geographic location, etc. Despite the importance of this notion of group fairness, unfortunately, *no work exists along this line in FL.*

Contribution: We propose a framework, `GIFAIR-FL`, that aims for fairness in FL. `GIFAIR-FL` resorts to regularization techniques by penalizing the spread in the loss of clients/groups to drive the optimizer to fair solutions. We show that our regularized formulation can be viewed as a dynamic client re-weighting technique that adaptively gives higher weights to low-performing individuals or groups. Our proposed method adapts the client weights at every communication round accordingly. One key feature of `GIFAIR-FL` is that it can handle both **group-level and individual-level fairness**. Also, `GIFAIR-FL` can be naturally tailored to either a global FL algorithm or a personalized FL algorithm. We then prove that, under reasonable conditions, our algorithm converges to an optimal solution for strongly convex objective functions and to a stationary solution for non-convex functions under **non-$i.i.d.$ settings**. Through empirical results on image classification and text prediction datasets, we demonstrate that `GIFAIR-FL` can **promote fairness while achieving superior or similar prediction accuracy** relative to recent state-of-the-art fair FL algorithms. Besides that, `GIFAIR-FL` can be easily plugged into other FL algorithms for different purposes.

Organization: The rest of the paper is organized as follows. In Sec. 2.2, we introduce important notations/definitions and briefly review FL. Related work is highlighted in Sec. 2.2.1. In Sec. 2.3, we present `GIFAIR-FL-Global` which is a global modeling approach for fairness in FL. We then briefly discuss the limitation of `GIFAIR-FL-Global` and introduce `GIFAIR-FL-Per` which is a personalized alternative for fairness, in Sec. 2.4. Meanwhile, we provide convergence guarantees for both methods. Experiments on image classification and text prediction tasks are then presented in Sec. 2.5. Finally, Sec. 2.6 concludes the paper with a brief discussion.

2.2 Background

We start by introducing needed background and notation for model development. Then we provide a brief overview of current literature.

Notation: Suppose there are $K \geq 2$ local devices and each device has N_k datapoints. Denote by $D_k = \big((x_{k,1}, y_{k,1}), (x_{k,2}, y_{k,2}), \ldots, (x_{k,N_k}, y_{k,N_k})\big)$ the data stored at device k where $x \in \mathcal{X}$ is the input, $\mathcal{X}$ is the input space, $y \in \mathcal{Y}$ is the output/label and $\mathcal{Y}$ is the output space. Denote by $\Delta_{\mathcal{Y}}$ the simplex over $\mathcal{Y}$, $h : \mathcal{X} \mapsto \Delta_{\mathcal{Y}}$ the hypothesis and $\mathcal{H}$ a family of hypotheses h. Let ℓ be a loss function defined over $\Delta_{\mathcal{Y}} \times \mathcal{Y}$. Without loss of generality, assume $\ell \geq 0$. The loss of h is therefore given by $\ell(h(x), y)$. Let $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ be a vector of parameters defining a hypothesis h and $\boldsymbol{\Theta}$ is a parameter space. For instance, $\boldsymbol{\theta}$ can be the model parameters of a deep neural network. In the following section we use $h_{\boldsymbol{\theta}}$ to represent the hypothesis.

Brief background on FL with `FedAvg`: In FL, clients collaborate to learn a model that yields better performance relative to each client learning in isolation. This model is called the global model where the global objective function is to minimize the average loss over all clients:

$$\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) \coloneqq \sum_{k=1}^{K} p_k F_k(\boldsymbol{\theta}),$$

where $p_k = \frac{N_k}{\sum_k N_k}$, $F_k(\boldsymbol{\theta}) = \mathbb{E}_{(x_{k,i}, y_{k,i}) \sim \mathcal{D}_k}[\ell(h_{\boldsymbol{\theta}}(x_{k,i}), y_{k,i})] \approx \frac{1}{N_k} \sum_{j=1}^{N_k} [\ell(h_{\boldsymbol{\theta}}(x_{k,j}), y_{k,j})]$ and $\mathcal{D}_k$ indicates the data distribution of the k-th device's data observations $(x_{k,i}, y_{k,i})$. During training, all devices collaboratively learn global model parameters $\boldsymbol{\theta}$ to minimize $F(\boldsymbol{\theta})$. The most commonly used method to learn the global objective is `FedAvg` (McMahan et al. 2017). Details of `FedAvg` are highlighted in Algorithm 2.1 as our work will build upon it for fairness. As shown in Algorithm 2.1, `FedAvg` aims to learn a global parameter $\boldsymbol{\theta}$, by iteratively averaging local updates $\boldsymbol{\theta}_k$ learned by performing E steps of stochastic gradient descent (SGD) on each client's local objective F_k.

```
Algorithm 2.1: FedAvg Algorithm
Data: number of communication rounds $C$, number of local updates $E$, SGD learning rate
      schedule $\{\eta^{(t)}\}_t$, initial model parameter $\boldsymbol{\theta}$
for $c = 0 : (C - 1)$ do
    Select some clients by sampling probability $p_k$ and denote by $\mathcal{S}_c$ the set of selected
      clients;
    Server broadcasts $\boldsymbol{\theta}$;
    for all selected devices do
        $\boldsymbol{\theta}_k^{(cE)} = \boldsymbol{\theta}$;
        for $t = cE : ((c + 1)E - 1)$ do
            Randomly sample a subset of data and denote it as $\zeta_k^{(t)}$;
            Local Training $\boldsymbol{\theta}_k^{(t+1)} = \boldsymbol{\theta}_k^{(t)} - \eta^{(t)} g_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)})$ ;
        end
    end
    Aggregation $\bar{\boldsymbol{\theta}}_c = \frac{1}{|\mathcal{S}_c|} \sum_{k \in \mathcal{S}_c} \boldsymbol{\theta}_k^{((c+1)E)}$, Set $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}_c$;
end
Return $\bar{\boldsymbol{\theta}}_C$.
```

In Algorithm 2.1, ζ_k denotes the set of indices corresponding to a subset of training data on device k and $g_k(\cdot;\zeta_k)$ denotes the stochastic gradient of $F_k(\cdot)$ evaluated on the subset of data indexed by ζ_k. Also, $|\mathcal{S}_c|$ denotes the cardinality of $\mathcal{S}_c$. One should note that it is also common for the central server to sample clients uniformly and then take a weighted average using p_k (Li et al. 2019b). Whichever method used, the resulting model may not be fair as small a p_k implies a lower weight for client k

Defining fairness in FL: Suppose there are $d \in [2, K]$ groups and each client can be assigned to one of those groups $s \in [d] \coloneqq \{1, \ldots, d\}$. **Note that clients from different groups are typically non-*i.i.d.*** Denote by $k^i, k \in [K], i \in [d]$ the index of k-th local device in group i. Throughout this paper, we drop the superscript i unless we want to emphasize i explicitly. Group fairness can be defined as follows.

Definition 1. *Denote by $\{a_1^i\}_{1\le i\le d}$ and $\{a_2^i\}_{1\le i\le d}$ the sets of performance measures (e.g., testing accuracy) of trained models 1 and 2 respectively. We say model 1 is more fair than model 2 if $Var(\{a_1^i\}_{1\le i\le d}) < Var(\{a_2^i\}_{1\le i\le d})$, where Var is variance.*

Definition 1 is straightforward: a model is fair if it yields small discrepancies among testing accuracies of different groups. It can be seen that when $d = K$, Definition 1 is equivalent to individual fairness (Li et al. 2019a). Definition 1 is widely adopted in FL literature (Mohri et al. 2019, Li et al. 2019a, 2020b, 2021). This notion of fairness might be different from traditional definitions such as demographic disparity (Feldman et al. 2015), equal opportunity and equalized odds (Hardt et al. 2016) in centralized systems. The reason is that those definitions cannot be extended to FL as there is no clear notion of an outcome which is "good" for a device (Kairouz et al. 2019). Instead, fairness in FL can be reframed as equal access to effective models (e.g., the accuracy parity (Zafar et al. 2017) or the representation disparity (Li et al. 2019a)). Specifically, the goal is to train models that incur a uniformly good performance across all devices (Kairouz et al. 2019).

2.2.1 Literature Overview

Now we briefly review existing state-of-the-art fair and personalized FL algorithms.

Fair FL: Mohri et al. (2019) propose a minimax optimization framework named agnostic FL (`AFL`). `AFL` optimizes the worst weighted combination of local devices and is demonstrated to be robust to unseen testing data. Du et al. (2020) further refine the notation of `AFL` and propose the `AgnosticFair` algorithm. Specifically, they linearly parametrize weight parameters by kernel functions and show that `AFL` can be viewed as a special case of `AgnosticFair`. Upon that, Hu et al. (2020) combine minimax optimization with gradient normalization techniques to produce a fair algorithm `FedMGDA+`. Motivated by fair resource allocation problems, Li et al. (2019a) propose q-Fair FL (`q-FFL`). `q-FFL` reweights loss functions such that devices with poor performance will be given relatively higher weights. The q-FFL objective is proved to encourage individual fairness in FL. However, this algorithm requires accurate estimation of a local Lipschitz constant L. Later, Li et al. (2020b) developed a tilted empirical risk minimization (TERM) algorithm to handle outliers and class imbalance in statistical estimation procedures. TERM has been shown to be superior to q-FFL in many FL applications. Along this line, Huang et al. (2020) propose to use training accuracy and frequency to adjust weights of devices to promote fairness. Zhang et al. (2020b) develop an algorithm to minimize the discrimination index of the global model to encourage fairness. Here we note that recent work study collaborative fairness in FL (Zhang et al. 2020c, Xu and Lyu 2020, Lyu et al. 2020). The goal of this literature, which is perpendicular to our purpose, is to provide more rewards to high-contributing participants while penalizing free riders.

Personalized FL: One alternative to global modeling is personalized FL (Shi and Kontar 2022a, Liang et al. 2023, Shi and Kontar 2022b) which allows each client to retain their own individualized parameters $\{\boldsymbol{\theta}_k\}_{k=1}^K$. For instance, in Algorithm 2.1 and after training is done, each device i can use $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}_C$ as the initial weight and run additional SGD steps to obtain a personalized solution $\boldsymbol{\theta}_i$. Though personalization techniques do not directly target fairness, recent papers have shown that personalized FL algorithms may improve fairness. Arivazhagan et al. (2019) and Liang et al. (2020) use different layers of a network to represent global and personalized solutions. Specifically, they fit personalized layers to each local device such that each device will return a task-dependent solution based on its own local data. Wang et al. (2019), Yu et al. (2020b), Dinh et al. (2020) and Li et al. (2021) resort to fine-tuning techniques to learn personalized models. Notably, Li et al. (2021) develops a multi-task personalized FL algorithm Ditto. After optimizing a global objective function, Ditto allows local devices to run more steps of SGD, subject to some constraints, to minimize their own losses. Li et al. (2021) have shown that Ditto can significantly improve testing accuracy among local devices and encourage fairness.

Features of GIFAIR-FL: We here give a quick comparison to highlight the features of our proposed algorithm. The detailed formulation of GIFAIR-FL will be presented in the following section. GIFAIR-FL resorts to regularization to penalize the spread in the loss of client groups. Interestingly, GIFAIR-FL can be seen as a dynamic re-weighting strategy based on the statistical ordering of client/group losses at each communication round. As such, our approach aligns with FL literature that uses re-weighting client schemes, yet existing work faces some limitations. Specifically, AFL and its variants (Mohri et al. 2019, Li et al. 2019a, Hu et al. 2020, Du et al. 2020) exploit minimax formulations that optimize the worst-case distribution of weights among clients to promote fairness. Such approaches lead to overly pessimistic solutions as they only focus on the device with the largest loss. As will be shown in our case studies, GIFAIR-FL significantly outperforms such approaches. Adding to this key advantage, GIFAIR-FL enjoys convergence guarantees even for non-$i.i.d.$ data and is amenable to both global and personalized modeling.

2.3 GIFAIR-FL-Global: a Global Model for Fairness

We start by detailing our proposed global modeling approach - GIFAIR-FL-Global. In this approach, all local devices collaborate to learn one global model parameter $\boldsymbol{\theta}$. Our fair FL formulation aims at imposing group fairness while minimizing the training error. More specifically, our goal is to minimize the discrepancies in the average group losses while achieving a low training error. By penalizing the spread in the loss among client groups, we propose a regularization framework for computing optimal parameters $\boldsymbol{\theta}$ that balances learning accuracy and fairness. This translates to solving the following optimization problem

$$\min_{\boldsymbol{\theta}} \; H(\boldsymbol{\theta}) \triangleq \sum_{k=1}^{K} p_k F_k(\boldsymbol{\theta}) + \lambda \sum_{1\le i<j\le d} |L_i(\boldsymbol{\theta}) - L_j(\boldsymbol{\theta})|, \tag{2.1}$$

where λ is a positive scalar that balances fairness and goodness-of-fit, and

$$L_i(\boldsymbol{\theta}) \triangleq \frac{1}{|\mathcal{A}_i|} \sum_{k\in\mathcal{A}_i} F_k(\boldsymbol{\theta})$$

is the average loss for client group i, $\mathcal{A}_i$ is the set of indices of devices that belong to group i, and $|\mathcal{A}|$ is the cardinality of the set $\mathcal{A}$.

Remark 2. *Objective* (2.1) *aims at ensuring fairness by reducing client loss spread when losses are evaluated at a single global parameter* $\boldsymbol{\theta}$*. This achieves fairness from the server perspective. Specifically, the goal is to find a single solution that yields small discrepancies among* $\{L_i(\boldsymbol{\theta})\}_{i=1}^{d}$.

In typical FL settings, the global objective is given as

$$H(\boldsymbol{\theta}) = \sum_{k=1}^{K} p_k H_k(\boldsymbol{\theta})$$

where each client uses local data to optimize a surrogate of the global objective function. For instance, FedAvg simply uses the local objective function $H_k(\boldsymbol{\theta}) = F_k(\boldsymbol{\theta})$ for a given client k. Interestingly, our global objective in (2.1) can also be written as $H(\boldsymbol{\theta}) = \sum_{k=1}^{K} p_k H_k(\boldsymbol{\theta})$ as shown in Lemma 3 below.

Lemma 3. *Let* $s_k \in [d]$ *denote the group index of device* k*. For any given* $\boldsymbol{\theta}$*, the global objective function* $H(\boldsymbol{\theta})$ *defined in* (2.1) *can be expressed as*

$$H(\boldsymbol{\theta}) = \sum_{k=1}^{K} p_k \left(1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta})\right) F_k(\boldsymbol{\theta}), \tag{2.2}$$

where

$$r_k(\boldsymbol{\theta}) = \sum_{1\le j\neq s_k\le d} \text{sign}(L_{s_k}(\boldsymbol{\theta}) - L_j(\boldsymbol{\theta})).$$

Consequently,

$$H(\boldsymbol{\theta}) = \sum_{k=1}^{K} p_k H_k(\boldsymbol{\theta})$$

such that the local client objective is,

$$H_k(\boldsymbol{\theta}) \triangleq \left(1 + \frac{\lambda}{p_k\,|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta})\right) F_k(\boldsymbol{\theta}). \tag{2.3}$$

In Lemma 3, $r_k(\boldsymbol{\theta}) \in \{-d+1, -d+3, \ldots, d-3, d-1\}$ is a scalar directly related to the statistical ordering of L_{s_k} among client group losses. To illustrate that in a simple example, suppose that at a given $\boldsymbol{\theta}$, we have $L_1(\boldsymbol{\theta}) \ge L_2(\boldsymbol{\theta}) \ge \ldots \ge L_d(\boldsymbol{\theta})$. Then,

$$r_k(\boldsymbol{\theta}) = \begin{cases} d-1 & \text{if } s_k = 1 \\ d-3 & \text{if } s_k = 2 \\ \quad\vdots & \\ -d+1 & \text{if } s_k = d. \end{cases}$$

According to Lemma 1, one can view our global objective as a parameter-based weighted sum of the client loss functions. Particularly, rather than using uniform weighting for clients, our assigned weights are functions of the parameter $\boldsymbol{\theta}$. More specifically, for a given parameter $\boldsymbol{\theta}$, our objective yields higher weights for groups with higher average group loss; hence, imposing group fairness. To illustrate this idea, we provide a simple concrete example.

Example 2.3.1. *Without loss of generality and for a given $\boldsymbol{\theta}$, consider four different groups each having 10 clients with $L_1(\boldsymbol{\theta}) > L_2(\boldsymbol{\theta}) > L_3(\boldsymbol{\theta}) > L_4(\boldsymbol{\theta})$. Then our global objective function $H(\boldsymbol{\theta})$ in* (2.2) *can be expressed as*

$$\sum_{k=1}^{40} p_k F_k(\boldsymbol{\theta}) + \lambda\Big(|L_1(\boldsymbol{\theta}) - L_2(\boldsymbol{\theta})| + |L_1(\boldsymbol{\theta}) - L_3(\boldsymbol{\theta})| \\ + |L_1(\boldsymbol{\theta}) - L_4(\boldsymbol{\theta})| + |L_2(\boldsymbol{\theta}) - L_3(\boldsymbol{\theta})| + |L_2(\boldsymbol{\theta}) - L_4(\boldsymbol{\theta})| + |L_3(\boldsymbol{\theta}) - L_4(\boldsymbol{\theta})|\Big),$$

which is equivalent to

$$\sum_{k\in\mathcal{A}_1} p_k\left(1+\frac{3\lambda}{10p_k}\right)F_k(\boldsymbol{\theta})+\sum_{k\in\mathcal{A}_2} p_k\left(1+\frac{\lambda}{10p_k}\right)F_k(\boldsymbol{\theta})$$
$$+\sum_{k\in\mathcal{A}_3} p_k\left(1-\frac{\lambda}{10p_k}\right)F_k(\boldsymbol{\theta})+\sum_{k\in\mathcal{A}_4} p_k\left(1-\frac{3\lambda}{10p_k}\right)F_k(\boldsymbol{\theta}).$$

The objective clearly demonstrates a higher weight applied to clients that belong to a group with a higher average loss.

According to (2.3), the optimization problem solved by every selected client is a weighted version of the local objective in FedAvg. The objective imposes a higher weight for clients that belong to groups with higher average losses. These weights will be dynamically updated at every communication round. To assure positive weights for clients, we require the following bounds on λ

$$0\leq\lambda<\lambda_{max}\triangleq\min_k\left\{\frac{p_k|\mathcal{A}_{s_k}|}{d-1}\right\}.$$

When $\lambda=0$, our approach is exactly FedAvg. Moreover, a higher value of λ imposes more emphasis on fairness.

Now, the above formulation can be readily extended to individual fairness; simply through considering each client to be a group. This translates to the global objective in (2.2) to be given as

$$H(\boldsymbol{\theta})=\sum_{k=1}^{K}\left(1+\frac{\lambda}{p_k}r_k(\boldsymbol{\theta})\right)F_k(\boldsymbol{\theta}).$$

In essence our approach falls in line with FL literature that exploit re-weighting of clients. For instance, AFL proposed by Mohri et al. (2019) computes at every communication round the worst-case distribution of weights among clients. This approach promotes robustness but may be overly conservative in the sense that it focuses on the largest loss and thus causes very pessimistic performance to other clients. Our algorithm, however, adaptively updates the weight of clients at every communication round based on the statistical ordering of client/group losses. Moreover, the dynamic update of the weights can potentially avoid over-fitting by impeding updates for clients with low loss. We will further demonstrate the advantages of our algorithm in Sec. 2.5. In the next subsection, we provide our detailed algorithm for solving our proposed objective.

2.3.1 Algorithm

In this section, we describe our proposed algorithm GIFAIR-FL-Global which is detailed in Algorithm 2.2. We highlight the difference between GIFAIR-FL-Global and FedAvg in

Algorithm 2.2: `GIFAIR-FL-Global` Algorithm

```
Data: number of devices $K$, fraction $\alpha$, number of communication rounds $C$, number of
      local updates $E$, SGD learning rate schedule $\{\eta^{(t)}\}_t$, initial model parameter $\boldsymbol{\theta}$,
      regularization parameter $\lambda$, initial loss $\{L_i\}_{1 \leq i \leq d}$
for $c = 0 : (C - 1)$ do
    Select clients by sampling probability $p_k$ and denote by $\mathcal{S}_c$ the indices of these clients;
    Server broadcasts $\left(\boldsymbol{\theta}, \left\{\frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k^c(\boldsymbol{\theta})\right\}_{k \in \mathcal{S}_c}\right)$;
    for $k \in \mathcal{S}_c$ do
        $\boldsymbol{\theta}_k^{(cE)} = \boldsymbol{\theta}$;
        for $t = cE : ((c + 1)E - 1)$ do
            Randomly sample a subset of data and denote it as $\zeta_k^{(t)}$;
            $\boldsymbol{\theta}_k^{(t+1)} = \boldsymbol{\theta}_k^{(t)} - \eta^{(t)} \left(1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k^c(\boldsymbol{\theta})\right) g_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)})$ ;
            // Note that $r_k^c(\boldsymbol{\theta})$ is fixed during local update (See
               Remark 6)
        end
    end
    Aggregation $\bar{\boldsymbol{\theta}}_c = \frac{1}{|\mathcal{S}_c|} \sum_{k \in \mathcal{S}_c} \boldsymbol{\theta}_k^{((c+1)E)}$, Set $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}_c$;
    Calculate $L_i = \frac{1}{|\mathcal{A}_i|} \sum_{k \in \mathcal{A}_i} F_k(\boldsymbol{\theta}_k^{((c+1)E)})$ for all $i \in [d]$ and update $r_k^{c+1}(\boldsymbol{\theta})$;
    $c \leftarrow c + 1$;
end
Return $\bar{\boldsymbol{\theta}}_C$.
```

red color. At every communication round c, our algorithm selects a set of clients to participate in the training and shares r_k^c with each selected client. For each client, multiple SGD steps are then applied to a weighted client loss function. The updated parameters are then passed to the server that aggregates these results and computes r_k^{c+1}.

Computationally, our approach requires evaluating the client loss function at every communication round to compute r_k^c. Compared to existing fair FL approaches, `GIFAIR-FL-Global` is simple and computationally efficient. For instance, `q-FFL` proposed by Li et al. (2019a) first runs `FedAvg` to obtain a well-tuned learning rate and uses this learning rate to roughly estimate the Lipschitz constant L. Another example is `AFL` which requires running two gradient calls at each iteration to estimate the gradients of model and weight parameters. Similarly, `Ditto` requires running additional steps of SGD, at each communication round, to generate personalized solutions. In contrast, our proposed method can be seen as a fairness-aware weighted version of `FedAvg`.

Remark 4. *In Algorithm 2.2, we sample local devices by sampling probability p_k and aggregate model parameters by an unweighted average $\frac{1}{|\mathcal{S}_c|}\sum_{k\in\mathcal{S}_c}\boldsymbol{\theta}_k^{((c+1)E)}$. Alternatively, one may choose to uniformly sample clients. Then, the aggregation strategy should be replaced by $\bar{\boldsymbol{\theta}}_c = \frac{K}{|\mathcal{S}_c|}\sum_{k\in\mathcal{S}_c} p_k\boldsymbol{\theta}_k^{((c+1)E)}$ (Li et al. 2019b).*

Remark 5. *Instead of broadcasting p_k and $|\mathcal{A}_{s_k}|$ separately to local devices, the central server broadcasts the product $\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k^c(\boldsymbol{\theta})$ to client k. Hence, the local device k cannot obtain any information about p_k, $|\mathcal{A}_{s_k}|$ and $r_k^c(\boldsymbol{\theta})$. This strategy can protect privacy of other devices.*

Remark 6. *Notice that in* (2.3)*, $H_k(\boldsymbol{\theta})$ is not differentiable due to the $r_k(\boldsymbol{\theta})$ component. However, r_k^c is fixed during local client training as it is calculated on the central server. Also, local devices do not have any information about other devices hence they cannot update r_k^c during local training.*

Remark 7. *Despite introducing $O(d^2)$ regularizers to the main objective, our algorithm only requires computing group losses and r_k^c values which require sorting the losses. More specifically, once the central server collects the selected clients' losses $\{F_k\}_k$, it first calculates group losses $\{L_i\}_i$. This step only involves the summation of scalars. Afterward, the server runs a sort algorithm to rank loss values. One can use many built-in sort functions in the Python library and this sorting step is very fast even with millions of groups.*

2.3.2 Convergence Guarantees

In this section, we first show that, under mild conditions, `GIFAIR-FL-Global` converges to the global optimal solution at a rate of $\mathcal{O}(\frac{E^2}{T})$ for strongly convex functions and to a stationary point at a rate of $\mathcal{O}(\frac{(E-1)\log(T+1)}{\sqrt{T}})$, up to a logarithmic factor, for non-convex functions. Here $T := CE$ denotes the total number of iterations across all devices. **Our theorems hold for both $\boldsymbol{i.i.d.}$ and non-$\boldsymbol{i.i.d.}$ data.** Due to space limitation, we defer proof details to the Appendix.

2.3.2.1 Strongly Convex Functions

We assume each device performs E steps of local updates and make the following assumptions. Here, our assumptions are based on F_k rather than H_k. These assumptions are very common in many FL papers (Li et al. 2019c, 2018a, 2019b).

Assumption 7.1. *F_k is L-smooth and μ-strongly convex for all $k \in [K]$.*

Assumption 7.2. *The variance of stochastic gradient is bounded. Specifically,*

$$\mathbb{E}\bigg\{ \Big\| g_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) - \nabla F_k(\boldsymbol{\theta}_k^{(t)}) \Big\|^2 \bigg\} \leq \sigma_k^2, \ \ \forall k \in [K].$$

Assumption 7.3. *The expected squared norm of the stochastic gradient is bounded. Specifically,*

$$\mathbb{E}\bigg\{ \Big\| g_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) \Big\|^2 \bigg\} \leq G^2, \forall k \in [K].$$

Typically, data from different groups are **non-*$i.i.d.$*.** We modify the definition in Li et al. (2019b) to roughly quantify the degree of non-$i.i.d.$-ness. Specifically,

$$\Gamma_K = H^* - \sum_{k=1}^{K} p_k H_k^* = \sum_{k=1}^{K} p_k (H^* - H_k^*),$$

where $H^* \triangleq H(\boldsymbol{\theta}^*) = \sum_{k=1}^K H_k(\boldsymbol{\theta}^*)$ is the optimal value of the global objective function and $H_k^* \triangleq H_k(\boldsymbol{\theta}_k^*)$ is the optimal value of the local loss function. If data are $i.i.d.$, then $\Gamma_K \to 0$ as the number of samples grows. Otherwise, $\Gamma_K \neq 0$ (Li et al. 2019b). Given all aforementioned assumptions, we next prove the convergence of our proposed algorithm. We first assume all devices participate in each communication round (i.e., $|\mathcal{S}_c| = K, \forall c$).

Theorem 8. *Assume Assumptions 7.1-7.3 hold and $|\mathcal{S}_c| = K$. If $\eta^{(t)}$ is decreasing in a rate of $\mathcal{O}(\frac{1}{t})$ and $\eta^{(t)} \leq \mathcal{O}(\frac{1}{L})$, then for $\gamma, \mu, \epsilon > 0$, we have*

$$\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}_C) \Big\} - H^* \leq \frac{L}{2} \frac{1}{\gamma + T} \bigg\{ \frac{4\xi}{\epsilon^2 \mu^2} + (\gamma + 1) \left\| \bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^* \right\|^2 \bigg\},$$

where $\xi = 8(E-1)^2 G^2 + 4L\Gamma + 2\frac{\Gamma_{max}}{\eta^{(t)}} + 4\sum_{k=1}^K p_k^2 \sigma_k^2$ and $\Gamma_{max} := \sum_{k=1}^K p_k |(H^* - H_k^*)| \geq |\sum_{k=1}^K p_k (H^* - H_k^*)| = |\Gamma_K|$. Here $\bar{\boldsymbol{\theta}}^{(0)} := \boldsymbol{\theta}^{(0)}$ where $\boldsymbol{\theta}^{(0)}$ is the initial model parameter in the central server.

Remark 9. *Theorem 38 shows an $\mathcal{O}(\frac{E^2}{T})$ convergence rate which is similar to that obtained from `FedAvg`. However, the rate is also affected by ξ, which contains the degree of non-$i.i.d.$-ness.*

Under a fully i.i.d. settings where $\Gamma_K = \Gamma_{max} = 0$, we retain the typical `FedAvg` for strongly convex functions.

Next, we assume only a fraction of devices participate in each communication round (i.e., $|\mathcal{S}_c| = \alpha K, \forall c, \alpha \in (0,1)$). As per Algorithm 2.2, all local devices are sampled according to the sampling probability p_k (Li et al. 2018a). Our Theorem can similarly be extended to the scenario where devices are sampled uniformly (i.e., with the same probability). Recall, the aggregation strategy becomes $\bar{\boldsymbol{\theta}}_c = \frac{K}{|\mathcal{S}_c|}\sum_{k\in\mathcal{S}_c} p_k \boldsymbol{\theta}_k$ (Li et al. 2019b).

Theorem 10. *Assume at each communication round, the central server samples a fraction $|\mathcal{S}_c|$ of devices according to the sampling probability p_k. Additionally, assume Assumptions 7.1-7.3 hold. If $\eta^{(t)}$ is decreasing at a rate of $\mathcal{O}(\frac{1}{t})$ and $\eta^{(t)} \leq \mathcal{O}(\frac{1}{L})$, then for $\gamma, \mu, \epsilon > 0$, we have*

$$\mathbb{E}\Big\{H(\bar{\boldsymbol{\theta}}_C)\Big\} - H^* \leq \frac{L}{2}\frac{1}{\gamma + T}\Big\{\frac{4(\xi + \tau')}{\epsilon^2\mu^2} + (\gamma + 1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2\Big\},$$

where $\tau' = \frac{4G^2E^2}{|\mathcal{S}_c|}$.

Remark 11. *Under the partial device participation scenario, the same convergence rate $\mathcal{O}(\frac{E^2}{T})$ holds. The only difference is that there is a term $\tau' = \frac{4G^2E^2}{|\mathcal{S}_c|}$ that appears in the upper bound. This ratio slightly impedes the convergence rate when the number of sampled devices $|\mathcal{S}_c|$ is small.*

2.3.2.2 Non-convex Functions

To prove the convergence result on non-convex functions, we replace Assumption 7.1 by the following assumption.

Assumption 11.1. *F_k is L-smooth for all $k \in [K]$.*

Theorem 12. *Assume Assumptions 7.2-11.1 hold and $|\mathcal{S}_c| = K$. If $\eta^{(t)} = \mathcal{O}(\frac{1}{\sqrt{t}})$ and $\eta^{(t)} \leq \mathcal{O}(\frac{1}{L})$, then our algorithm converges to a stationary point. Specifically,*

$$\begin{aligned}\min_{t=1,\ldots,T} \mathbb{E}\Big\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\Big\} \\ \leq \frac{\Big\{2\big(1 + 2L^2\log(T+1)\big)\mathbb{E}\big\{H(\bar{\boldsymbol{\theta}}^{(0)}) - H^*\big\} + 2\xi_{\Gamma_K}\Big\}}{\sqrt{T}},\end{aligned}$$

where

$$\xi_{\Gamma_K} = \mathcal{O}\Big(\big(2L^2\Gamma_K + 8(E-1)LG^2 + 10L\sum_{k=1}^{K} p_k\sigma_k^2\big)\log(T+1)\Big),$$

and $\bar{\boldsymbol{\theta}}^{(t)} = \frac{1}{|\mathcal{S}_c|}\sum_{k\in\mathcal{S}_c}\boldsymbol{\theta}_k^{(t)}$.

Remark 13. *Our results show that* `GIFAIR-FL` *converges to a stationary point at a rate of* $\mathcal{O}(\frac{(E-1)\log(T+1)}{\sqrt{T}})$*. Similar to the strongly-convex setting, this convergence rate is affected by the degree of non-*$i.i.d.$*-ness* Γ_K*.*

2.3.3 Discussion and Limitations

We here note our theoretical results require exact computation of r_k. However, Algorithm 2.2 uses an estimate of r_k at every communication round using the local client loss prior to aggregation. This procedure might generate an inexact estimate of r_k as one cannot guarantee $F_k(\bar{\boldsymbol{\theta}}_c) = F_k(\boldsymbol{\theta}_k^{((c+1)E)})$ at each communication round. Here recall that r_k in Eq. (2.2) is calculated based on the order of $F_k(\bar{\boldsymbol{\theta}}_c)$. To guarantee exact an r_k, the server can ask clients to share the local losses evaluated at the global parameters. Specifically, after sharing $\bar{\boldsymbol{\theta}}_c$ to selected local devices, those devices calculate $\{F_k(\bar{\boldsymbol{\theta}}_c)\}_k$ and send loss values back to the central server to update r_k^{c+1}. This, however, requires more communication rounds. One approach to remedy the limitation of `GIFAIR-FL-Global` is to develop a personalized counterpart to circumvent additional communication rounds. We will detail this idea in the coming section.

2.4 `GIFAIR-FL-Per`: A Personalized Model for Fairness

In this section, we slightly tailor `GIFAIR-FL-Global` to a personalized fair algorithm `GIFAIR-FL-Per`. While still aiming to minimize the spread in the loss among client groups, our proposed objective evaluates the loss at the client-specific (i.e. personalized) solution . Formally speaking, our objective function is

$$\min_{\boldsymbol{\theta}} \; H(\boldsymbol{\theta}, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K) \triangleq \sum_{k=1}^{K} p_k F_k(\boldsymbol{\theta}) + \lambda \sum_{1 \le i < j \le d} \left| L_i(\{\boldsymbol{\theta}_k\}_{k \in \mathcal{A}_i}) - L_j(\{\boldsymbol{\theta}_k\}_{k \in \mathcal{A}_j}) \right|, \qquad (2.4)$$

where

$$L_i(\{\boldsymbol{\theta}_k\}_{k \in \mathcal{A}_i}) \triangleq \frac{1}{|\mathcal{A}_i|} \sum_{k \in \mathcal{A}_i} F_k(\boldsymbol{\theta}_k)$$

is the average loss for client group i and $\sum_{k=1}^{K} p_k \boldsymbol{\theta}_k = \boldsymbol{\theta}$.

Remark 14. *Different from* (2.1)*, objective* (2.4) *achieves fairness from the device perspective. By optimizing* (2.4)*, we can obtain device-specific solutions* $\{\boldsymbol{\theta}_k\}_{k=1}^{K}$ *that yield small discrepancies among* $\{L_i(\{\boldsymbol{\theta}_k\}_{k \in \mathcal{A}_i})\}_{i=1}^{d}$*. Although* (2.1) *and* (2.4) *have different perspectives, their ultimate goals are aligned with Definition 1.*

Objective (2.4) has many notable features: (i) first, compared to (2.1), the new objective function (2.4) evaluates group losses $\{L_i\}_{i=1}^d$ with respect to personalized solutions $\{\boldsymbol{\theta}_k\}_{k=1}^K$. This formulation circumvents the need to collect losses evaluated at global parameter and therefore requires no extra communication rounds to calculate r_k exactly; (ii) second, the global parameter $\boldsymbol{\theta} = \sum_{k=1}^K p_k \boldsymbol{\theta}_k$ ensures aggregation happens at every communication round. This can safeguard against over-fitting on local devices. Otherwise, each device will simply minimize its own local loss, without communication, and obtain a small loss value; (iii) before discussing the third property, we first need to present the convergence results of `GIFAIR-FL-Per`.

Theorem 15. *Assume Assumptions 7.1-7.3 hold and $|\mathcal{S}_c| = K$. If $\eta^{(t)}$ is decreasing in a rate of $\mathcal{O}(\frac{1}{t})$ and $\eta \leq \mathcal{O}(\frac{1}{L})$, then for $\gamma, \mu, \epsilon > 0$, we have*

$$\mathbb{E}\left\{ H(\bar{\boldsymbol{\theta}}_C, \{\boldsymbol{\theta}_k^{(T)}\}_{k=1}^K) \right\} - H^* \leq \mathcal{O}(\frac{E^2}{T})$$

where $\sum_{k=1}^K p_k \boldsymbol{\theta}_k^{(T)} = \bar{\boldsymbol{\theta}}_C$ and $H^* := H(\boldsymbol{\theta}^*, \{\boldsymbol{\theta}_k\}_{k=1}^K)$ such that $\sum_{k=1}^K p_k \boldsymbol{\theta}_k = \boldsymbol{\theta}^*$. A same convergence rate holds for the partial device participation scenario.

Under non-convex condition, we have

$$\min_{t=1,\dots,T} \mathbb{E}\left\{ \left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2 \right\} \leq \mathcal{O}(\frac{(E-1)\log(T+1)}{\sqrt{T}}).$$

The proof here follows a similar scheme to those in `GIFAIR-FL-Global`. Theorem 15 implies that `GIFAIR-FL-Per` drives aggregated parameter $\bar{\boldsymbol{\theta}}_C$ to the global optimal solution $\boldsymbol{\theta}^*$ at a rate of $\mathcal{O}(\frac{E^2}{T})$. This aggregated parameter is obtained from taking the weighed average of personalized solutions $\{\boldsymbol{\theta}_k^{(T)}\}_{k=1}^K$. This leads to a new interpretation of `GIFAIR-FL-Per`: once the optimizer reaches $\bar{\boldsymbol{\theta}}_C$, device k retains personalized solution $\boldsymbol{\theta}_k^{(T)}$ that stays in the vicinity of the global model parameter to balance each client's shared knowledge and unique characteristics. One can link this idea to `Ditto` (Li et al. 2021) - the recent state-of-the-art personalized FL algorithm. `Ditto` allows local devices to run more steps of SGD, subject to some constraints such that local solutions will not move far away from the global solution. `GIFAIR-FL-Per`, on the other hand, scales the magnitude of gradients based on the statistical ordering of client/group losses.

Finally, we detail `GIFAIR-FL-Per` in Algorithm 2.3. In the algorithm, r_k is defined as

$$r_k(\{\boldsymbol{\theta}_k\}_{k=1}^K) = \sum_{1 \leq j \neq s_k \leq d} \text{sign}(L_{s_k}(\{\boldsymbol{\theta}_m\}_{m \in \mathcal{A}_{s_k}}) - L_j(\{\boldsymbol{\theta}_m\}_{m \in \mathcal{A}_j})). \tag{2.5}$$

In other words, r_k is computed based on the ordering of losses evaluated **on the personalized solutions**.

Algorithm 2.3: `GIFAIR-FL-Per` Algorithm

```
Data: number of devices $K$, fraction $\alpha$, number of communication rounds $C$, number of
      local updates $E$, SGD learning rate schedule $\{\eta^{(t)}\}_{t=1}^{E}$, initial model parameter $\boldsymbol{\theta}$,
      regularization parameter $\lambda$, initial loss $\{L_i\}_{1\leq i\leq d}$
for $c = 0 : (C-1)$ do
    Select $|\mathcal{S}_c|$ clients by sampling probability $p_k$ and denote by $\mathcal{S}_c$ the indices of these
      clients;
    Server broadcasts $\left(\boldsymbol{\theta}, \left\{\frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k^c(\{\boldsymbol{\theta}_k\}_{k=1}^{K})\right\}_{k\in\mathcal{S}_c}\right)$;
    for $k \in \mathcal{S}_c$ do
        $\boldsymbol{\theta}_k^{(cE)} = \boldsymbol{\theta}$;
        for $t = cE : ((c+1)E-1)$ do
            $\boldsymbol{\theta}_k^{(t+1)} = \boldsymbol{\theta}_k^{(t)} - \eta^{(t)}\left(1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k^c(\{\boldsymbol{\theta}_k\}_{k=1}^{K})\right)\nabla F_k(\boldsymbol{\theta}_k^{(t)})$ ;
        end
    end
    Aggregation $\bar{\boldsymbol{\theta}}_c = \frac{1}{|\mathcal{S}_c|}\sum_{k\in\mathcal{S}_c}\boldsymbol{\theta}_k^{((c+1)E)}$, Set $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}_c$;
    Calculate $L_i = \frac{1}{|\mathcal{A}_i|}\sum_{k\in\mathcal{A}_i} F_k(\boldsymbol{\theta}_k^{((c+1)E)})$ for all $i \in [d]$;
    Set $\boldsymbol{\theta}_k = \boldsymbol{\theta}_k^{((c+1)E)}$ for all $k \in \mathcal{S}_c$. Remain $\boldsymbol{\theta}_k$ unchanged otherwise;
    update $r_k^{c+1}(\{\boldsymbol{\theta}_k\}_{k=1}^{K})$;
    $c \leftarrow c+1$;
end
Return $\{\boldsymbol{\theta}_k\}_{k=1}^{K}$.
```

2.5 Experiments

In this section, we test `GIFAIR-FL` on image classification and text prediction tasks.

We benchmark our model with the following algorithms: `q-FFL` (Li et al. 2019a), `TERM` and `TERM-Group` (Li et al. 2020b), `FedMGDA+` (Hu et al. 2020), `AFL` (Mohri et al. 2019), and `FedMGDA+` (Hu et al. 2020). To the best of our knowledge, those are the well-known current state-of-the-art FL algorithms for fairness. We also benchmark our model with `Ditto` (Li et al. 2021) which is a personalized FL approach using multi-task learning.

2.5.1 Image Classification

We start by considering a federated image classification dataset FEMNIST (Federated Extended MNIST) (Caldas et al. 2018). FEMNIST consists of images of digits (0-9) and English characters (A-Z, a-z) with 62 classes (Figure A.1) written by different people. Images are 28 by 28 pixels. All images are partitioned and distributed to 3,550 devices by the dataset creators (Caldas et al. 2018).

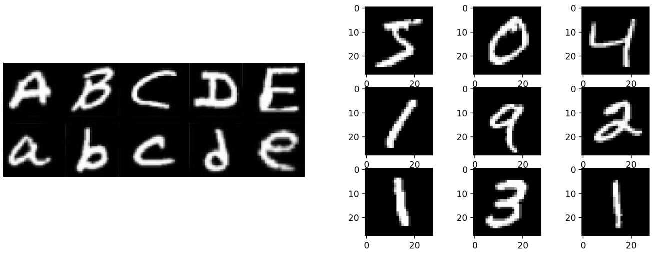

Figure 2.3: Example of Images from FEMNIST.

Individual Fairness (**FEMNIST-skewed**, $d = 100$) Following the setting in (Li et al. 2018a), we first sample 10 lower case characters ('a'-'j') from Extended MNIST (EMNIST) (Cohen et al. 2017) and distribute 5 classes of images to each device. Each local device has 500 images. There are 100 devices in total. Results are reported in Table 2.2. (**FEMNIST-original**, $d = 500$) Following the setting in (Li et al. 2021), we sample 500 devices and train models using the default data stored in each device. Results are reported in Table 2.3.

Group Fairness (**FEMNIST-3-groups**, $d = 3$) We manually divide FEMNIST data into three groups. See Table 2.1 for the detailed assignment. *This assignment is inspired by the statistic that most people prefer to write in lowercase letters while a small amount of people use capital letters or a mixed of two types (Jones and Mewhort 2004).* In such cases, it is important to assure that an FL algorithm is capable of achieving similar performance between such groups. Results are reported in Table 2.4.

Group	Data Type	Number of Images	Number of Devices
Group 1	Capital Letters + Digits	800	60
Group 2	Lowercase Letters + Digits	1,000	100
Group 3	Capital/Lowercase Letters + Digits	600	40

Table 2.1: Data Structure of FEMNIST-3-groups

Implementation: For all tasks, we randomly split the data on each local device into a 70% training set, a 10% validation set and a 20% testing set. This is a common data splitting strategy used in many FL papers (Li et al. 2018a, Chen et al. 2018, Reddi et al. 2020). The batch size is set to be 32. We use the tuned initial learning rate 0.1 and decay rate 0.99 for each method. During each communication round, 10 devices are randomly selected and each device will run 2 epochs of SGD. We use a CNN model with 2 convolution layers followed by 2 fully connected layers. All benchmark models are well-tuned. Specifically, we solve `q-FFL` with $q \in \{0, 0.001, 0.01, 0.1, 1, 2, 5, 10\}$ (Li et al. 2019a) in parallel and select the best q. Here, the best q is defined as the q value where the variance decreases the most while the averaged testing accuracy is superior or similar to `FedAvg`. This definition is borrowed from the original `q-FFL` paper (Li et al. 2019a). Similarly, we train `TERM` with $t \in \{1, 2, 5\}$ and select the best t (Li et al. 2020b). For `Ditto`, we tune the regularization parameter $\lambda_{Ditto} \in \{0.01, 0.05, 0.1, 0.5, 1, 2, 5\}$. In `GIFAIR-FL`, we tune the parameter $\lambda \in \{0, 0.1\lambda_{max}, 0.2\lambda_{max}, \ldots, 0.8\lambda_{max}, 0.9\lambda_{max}\}$. Here kindly note that λ_{max} is a function of p_k, $|\mathcal{A}_{s_k}|$ and d (i.e., data-dependent).

Performance metrics: Denote by a_k the prediction accuracy on device k. We define (1) individual-level mean accuracy as $\bar{a} := \frac{1}{K}\sum_{k=1}^{K} a_k$ and (2) individual-level variance as $Var(a) := \frac{1}{K}\sum_{k=1}^{K}(a_k - \bar{a})^2$.

Algorithm	`FedAvg`	`q-FFL`	`TERM`	`FedMGDA+`	`Ditto`	`GIFAIR-FL-Global`	`GIFAIR-FL-Per`
$\bar{a}$	79.2 (1.0)	84.6 (1.9)	84.2 (1.3)	85.0 (1.7)	92.5 (3.1)	87.9 (0.9)	**93.0** (1.1)
$\sqrt{Var(a)}$	22.3 (1.1)	18.5 (1.2)	13.8 (1.0)	14.9 (1.6)	14.3 (1.0)	**5.7** (0.8)	6.2 (0.9)

Table 2.2: Empirical results on FEMNIST-skewed. Each experiment is repeated 5 times.

Algorithm	`FedAvg`	`q-FFL`	`TERM`	`AFL`	`Ditto`	`GIFAIR-FL-Global`	`GIFAIR-FL-Per`
$\bar{a}$	80.4 (1.3)	80.9 (1.1)	81.0 (1.0)	82.4 (1.0)	83.7 (1.9)	83.2 (0.7)	**84.1 (1.2)**
$\sqrt{Var(a)}$	11.1 (1.4)	10.6 (1.3)	10.3 (1.2)	9.85 (0.9)	10.1 (1.6)	5.2 (0.8)	**4.5 (0.8)**

Table 2.3: Test accuracy on FEMNIST-original. Each experiment is repeated 5 times.

2.5.2 Text Data

Individual Fairness: We train a RNN to predict the next character using text data built from “The Complete Works of William Shakespeare”. In this dataset, there are about 1,129 speaking roles.

Algorithm	FedAvg	q-FFL	TERM	FedMGDA+	Ditto	TERM-Group	GIFAIR-FL-Global	GIFAIR-FL-Per
Group 1	79.72 (2.08)	81.15 (1.97)	81.29 (1.45)	81.03 (2.28)	82.37 (2.06)	82.01 (1.95)	83.41 (1.34)	**83.96 (1.22)**
Group 2	90.93 (2.35)	88.24 (2.13)	88.08 (1.09)	89.12 (1.74)	**92.05 (2.00)**	89.13 (1.00)	88.29 (1.22)	91.05 (1.31)
Group 3	80.21 (2.91)	80.93 (1.86)	81.84 (1.44)	81.33 (1.59)	83.03 (2.18)	81.75 (2.04)	84.37 (1.85)	**84.98 (0.99)**
Discrepancy	11.21	7.31	6.79	8.09	9.02	7.38	**6.07**	7.09

Table 2.4: Test accuracy on FEMNIST-3-groups. Each experiment is repeated 5 times. Discrepancy is the difference between the largest accuracy and the smallest accuracy.

Naturally, each speaking role in the play is treated as a device. Each device stored several text data and those information will be used to train a RNN on each device. The dataset is available on the LEAF website (Caldas et al. 2018).

Following the setting in McMahan et al. (2017) and Li et al. (2019a), we subsample 31 roles ($d = 31$). The RNN model takes a 80-character sequence as the input, and outputs one character after two LSTM layers and one densely-connected layer. For FedAvg, q-FFL and Ditto, the best initial learning rate is 0.8 and decay rate is 0.95 (Li et al. 2021). We also adopt this setting to GIFAIR-FL-Global and GIFAIR-FL-Per. The batch size is set to be 10. The number of local epochs is fixed to be 1 and all models are trained for 500 epochs. Results are reported in Table 2.5.

Algorithm	FedAvg	q-FFL	AFL	Ditto	GIFAIR-FL-Global	GIFAIR-FL-Per
$\bar{a}$	53.21 (0.31)	53.90 (0.30)	54.58 (0.14)	60.74 (0.42)	57.04 (0.23)	**61.58 (0.14)**
$\sqrt{Var(a)}$	9.25 (6.17)	7.52 (5.10)	8.44 (5.65)	8.32 (4.77)	**3.14 (1.25)**	4.33 (1.25)

Table 2.5: Mean and standard deviation of test accuracy on Shakespeare ($d = 31$). Each experiment is repeated 5 times.

Group Fairness:

We obtain the gender information from https://shakespeare.folger.edu/ and group speaking roles based on gender ($d = 2$). It is known that the majority of characters in Shakespearean drama are males. Simply training a FedAvg model on this dataset will cause implicit bias towards male characters. On a par with this observation, we subsample 25 males and 10 females from "The Complete Works of William Shakespeare". Here we note that each device in the male group implicitly has more text data. The setting of hyperparameters is same as that of individual fairness. Results are reported in Table 2.6.

Algorithm	FedAvg	q-FFL	FedMGDA+	Ditto	TERM-Group	GIFAIR-FL-Global	**GIFAIR-FL-Per**
Male	72.95 (1.70)	67.14 (2.18)	67.07 (2.11)	**74.19 (3.75)**	72.87 (1.01)	67.42(0.98)	73.95 (0.59)
Female	40.39 (1.49)	43.26 (2.05)	43.85 (2.32)	45.73 (4.01)	44.31 (0.96)	52.04 (1.10)	**54.88 (1.12)**
Discrepancy	32.56	23.88	23.22	28.46	28.56	**15.38**	19.07

Table 2.6: Test accuracy on Shakespeare ($d = 2$). Each experiment is repeated 5 times.

2.5.3 Analysis of Results

Based on Table 2.2-2.6, we can obtain important insights. First, compared to other benchmark models, GIFAIR-FL-Global/GIFAIR-FL-Per lead to significantly more fair solutions. As shown in Tables 2.2, 2.3 and 2.5, our algorithm significantly reduces the variance of testing accuracy of all devices (i.e., $Var(a)$) while the average testing accuracy remains consistent. Second, from Tables 2.4 and 2.6, it can be seen that GIFAIR-FL-Global/GIFAIR-FL-Per boosted the performance of the group with the worst testing accuracy and achieved the smallest discrepancy. Notably, this boost did not affect the performance of other groups. This indicates that GIFAIR-FL-Global/GIFAIR-FL-Per is capable of ensuring fairness among different groups while retaining a superior or similar prediction accuracy compared to existing benchmark models. Finally, we note that GIFAIR-FL-Global sometimes achieves lower prediction performance than Ditto. This is understandable as Ditto provides a personalized solution to each device k while our model only returns a global parameter $\bar{\boldsymbol{\theta}}$. Yet, as shown in the last column, if we use GIFAIR-FL-Per, then the prediction performance can be significantly improved without sacrificing fairness. However, even without personalization, GIFAIR-FL-Global achieves superior testing performance compared to existing fair FL benchmark models.

2.5.4 Sensitivity Analysis

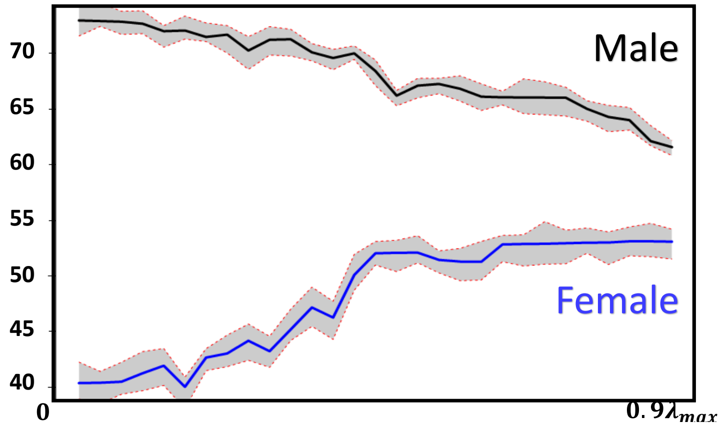

Figure 2.4: Sensitivity with respect to λ (Shakespeare Dataset).

In this section, we use GIFAIR-FL-Global to study the effect of the tuning parameter $\lambda \in$

$[0, \lambda_{max})$ using the Shakespeare dataset. A similar conclusion holds for `GIFAIR-FL-Per` and we therefore omit it. Results are reported in Figure 2.4. It can be seen that as λ increases, the discrepancy between male and female groups decreases accordingly. However, after λ passes a certain threshold, the averaged testing accuracy of the female group remained flat yet the performance of the male group significantly dropped. Therefore, in practice, it is recommended to consider a moderate λ value. Intuitively, when $\lambda = 0$, `GIFAIR-FL` becomes `FedAvg`. When λ is close to λ_{max}, the coefficient (i.e., $(1 + \lambda \frac{1}{p_k |\mathcal{A}_{s_k}|} r_k)$) of devices with good performance will be close to zero and the updating is, therefore, impeded. A moderate λ balances those two situations well. Besides this example, we also conducted additional sensitivity analysis. Due to space limitation, we defer those results to the Appendix.

2.6 Conclusion

In this paper, we propose `GIFAIR-FL`: a framework that imposes group and individual fairness to FL. Experiments show that `GIFAIR-FL` can lead to more fair solutions compared to recent state-of-the-art fair and personalized FL algorithms while retaining similar testing performance. To the best of our knowledge, fairness in FL is an under underinvestigated area and we hope our work will help inspire continued exploration into fair FL algorithms.

Also, real-life FL datasets for specific engineering or health science applications are still scarce. This is understandable as FL efforts have mainly focused on mobile applications. As such we only test on image classification and text prediction datasets. However, as FL is expected to infiltrate many applications, we hope that more real-life datasets will be generated to provide a means for model validation within different domains. We plan to actively pursue this direction in future research.

CHAPTER 3

Federated Data Analytics: A Study on Linear Models

As edge devices become increasingly powerful, data analytics are gradually moving from a centralized to a decentralized regime where edge compute resources are exploited to process more of the data locally. This regime of analytics is coined as federated data analytics (FDA). Despite the recent success stories of FDA, most literature focuses exclusively on deep neural networks. In this work, we take a step back to develop an FDA treatment for one of the most fundamental statistical models: linear regression. Our treatment is built upon hierarchical modeling that allows borrowing strength across multiple groups. To this end, we propose two federated hierarchical model structures that provide a shared representation across devices to facilitate information sharing. Notably, our proposed frameworks are capable of providing uncertainty quantification, variable selection, hypothesis testing, and fast adaptation to new unseen data. We validate our methods on a range of real-life applications, including condition monitoring for aircraft engines. The results show that our FDA treatment for linear models can serve as a competing benchmark model for the future development of federated algorithms.

3.1 Introductory Remarks

The sheer amount of data collected nowadays is beginning to overwhelm traditional centralized data analytics regimes where data from the edge is continuously uploaded to a central server to be processed. Excessive communication traffic from data upload, significant central server storage needs, energy expenditures from centralized learning of big data models, and privacy concerns from sharing raw data are becoming critical challenges in centralized systems. Statista, a German company specializing in market and consumer data, predicted that, by 2024, data produced on edge devices (e.g., cell phone data, self-driving vehicle data) would reach more than hundreds of zettabytes while the global central servers only have 10.4 zettabytes of storage (Morell and Alba 2022). Transmitting such a vast amount of edge data into a central server is infeasible. Adding to that, training a model with moderately large datasets results in significant budget costs and carbon emissions (Patterson et al. 2021). Furthermore, data-sharing comes with serious privacy concerns. According to Lawson et al. (2015), Canadian drivers who refused to enroll in the automotive telematics program demanded that their personal driving data (e.g., behavior, location, web-browsing history) should be respected by vehicle companies and that they be given control over the data collection process. These debates over data protection standards have not faded away over the past decade.

Fortunately, the Internet of Things (IoT) is undergoing a new revolution in which the compute power of edge devices is tremendously increasing (Hassan et al. 2018). AI Chips such as general-purpose chips (GPUs), semi-customized chips (FGPAs), and fully-customized chips (ASICs) are becoming readily available across many applications (Blanco-Filgueira et al. 2019, Rahman and Hossain 2021, Zhu et al. 2021b). Such AI chips are able to process a vast amount of data locally and provide timely responses and decisions (Shi et al. 2016). For instance, the autonomous vehicle company PerceptIn has released a real-time edge computing system, DragonFly+, that is three times more power efficient and delivers three to five times of the computing power of an Nvidia Tx1 and an Intel Core i7 processor (Liu et al. 2019). Another notable example is Tesla's autopilot system that has computing power on the car itself comparable to hundreds of MacBook pros (CleanTechnica 2021). As a consequence, traditional IoT is on the verge of shifting to a decentralized framework recently termed the Internet of Federated Things (IoFT) (Kontar et al. 2021) in which some of the data processing is deferred to the edge. In this future, the central server only acts as an orchestrator of the learning process and an integration point of model updates from different devices, rather than the central location where all data is processed. Indeed, IoFT is slowly infiltrating various fields such as manufacturing, transportation, and energy systems (Kontar et al. 2021).

The underlying data analytics framework in IoFT is federated data analytics (FDA), where edge devices exploit their own computation power to collaboratively extract knowledge and build smart analytics while keeping their personal data stored locally. Consequently, edge devices no longer need to upload their data to the cloud (or server), and, in turn, the cloud does not need to store that immense amount of data. As such, FDA resolves many of the aforementioned drawbacks of the centralized computing system and sets forth many intrinsic advantages, including privacy-preserving and reducing storage/computation/communication costs, among many others.

In spite of some recent advances in FDA, most, if not all, literature focuses on deep neural networks (Li et al. 2020c, Yue et al. 2020) (learned via first-order methods). To date, very few papers have delivered federated treatments of traditional statistical models. Perhaps the closest field where statistical models were investigated is distributed learning (DL) (Jordan et al. 2018), yet DL and FDA have several fundamental differences. Despite the terminology "distributed", DL is still a centralized computation approach where different compute nodes operate on all data (Fan et al. 2021). These nodes communicate often, observe each other's data, and can operate on different data partitions. The underlying philosophy for DL is "divide-and-conquer" where data is divided across the nodes (often dynamically), and then the nodes collaborate to "conquer" (learn) a single model. As a notable example, Zhang et al. (2015) propose a DL algorithm that solves a ridge regression problem. The basic idea of this paper is as follows: a server first evenly divides a set of data into m disjoint sets and assigns each set to a different node. Each node then solves a ridge regression problem and sends the optimal solutions back to the server. The server then aggregates local estimations. As a result, this approach returns a single global model. In contrast, in FDA, data resides at the edge and cannot be shuffled, randomized, or divided. Therefore, edge devices cannot see each others' data and data partitions for FDA are fixed and often heterogeneous. Besides that, devices have datasets with unique features as they correspond to different clients, components or systems (e.g., cars). Therefore, in FDA we cannot divide the data and there is often no single model to conquer, rather, our goal is to borrow strength across edge devices to improve inference and prediction.

In this work, we take a step back and move out of the regime of deep neural networks to study one of the most fundamental statistical models: linear regression (LR) (Yue et al. 2022a). Indeed, linear models may facilitate hypothesis testing, uncertainty quantification, variable selection, deriving engineering insight, and establishing a baseline to compare other models with. Needless to say, in reality, many real applications can be sufficiently characterized by linear models (Liu et al. 2013, Si et al. 2017, Li et al. 2018b, Schulz et al. 2020, Arashi et al. 2021, Şahın et al. 2022). In addition, building upon FDA for linear models, one may develop approaches for more complex derivatives such as logistic regression, mixed-effects, and kernel methods.

To this end, we exploit the properties and structure of linear models and develop an FDA treatment for linear regression with Gaussian noise, entitled `FedLin`. Our treatment is built upon hierarchical models (HM), which allow borrowing statistical knowledge across groups (i.e., devices or clients in FDA). Specifically, we propose two federated HM structures that provide a shared representation across devices to facilitate information transfer. The first structure establishes a shared representation defined through a structural prior over concatenated device parameters. The second structure is based on the assumption that all device parameters are generated from the same underlying distribution. This allows uncertainty quantification and, consequently, a Bayesian treatment for variable selection in `FedLin`. Our methods are validated on a range of real-life problems, including variable selection and condition monitoring. The results highlight the effective performance of our approaches and their ease of implementation, which may help them serve as benchmark models for many future developments of federated statistical algorithms.

Organization: The remainder of this paper is organized as follows. In Section 3.2, we conduct a literature review, and introduce the general setting and motivation. In Sections 3.3 and 3.4, we present our two model structures and their applications. We validate our proposed models on various simulated and real-life datasets in Sections 3.5 and 3.6. Finally, we conclude our paper in Section

3.7.

3.2 Background

3.2.1 Literature Overview

The idea of FDA was first brought to the forefront of deep learning by McMahan et al. (2017). In this work, they proposed the FDA algorithm termed federated averaging (`FedAvg`). The idea of `FedAvg` is simple: a central server distributes initial deep learning model parameters and the network structure to some selected devices, devices perform local stochastic gradient descent (SGD) steps using their data and send their updated parameters back. The server then takes an average of those parameters to update the global model. This process is termed as one communication round and is iterated several times. Although simple, `FedAvg` is still one of the most competitive benchmark models nowadays (Kairouz et al. 2021). To date, some work has been proposed to improve the performance of federated deep learning algorithms. For instance, Yuan and Ma (2020) and Liu et al. (2020) provided several provable techniques to accelerate `FedAvg` and enable faster convergence. Li et al. (2019a), Yu et al. (2020c), Yue et al. (2021), Du et al. (2021) developed variants of `FedAvg` that ensure uniformly good performance across all devices to achieve fairness. Another line of work aims to develop personalized solutions in federated data analytics as excessive heterogeneity can greatly impact the performance of a single global model (Deng et al. 2020, Fallah et al. 2020, Li et al. 2021). Such approaches usually either follow a train-then-personalize philosophy where a trained global model is fine-tuned on local devices or divide the layers of a neural network into shared and individualized ones (Tan et al. 2022), where devices collaborate to learn the common layers using methods such as `FedAvg`. From a theoretical perspective, Stich (2018), Li et al. (2020d) prove the convergence of `FedAvg` for convex functions and homogeneous (*i.i.d.*) datasets. Those results are then extended to a non-convex setting by Wang and Joshi (2021). On the other hand, Li et al. (2019b) extend the results of Stich (2018) to the non-*i.i.d.* setting. Furthermore, Shi et al. (2021) extend the convergence results to a kernel regime. For a comprehensive overview of current literature, please refer to Kontar et al. (2021).

The major trend of FDA exclusively focuses on neural networks and classification tasks. FDA for statistical models is still scant. Yue and Kontar (2021) extend the Gaussian process to a federated framework and show that their proposed algorithm can achieve state-of-the-art performance on multi-fidelity modeling problems. Yuan et al. (2021) develop a federated composite optimization framework that solves the federated Lasso problem. Tong et al. (2020) propose a federated iterative hard thresholding algorithm to tackle the non-convex 0-norm penalized regression problem. The two aforementioned papers mainly formulate penalized regression from a frequentist perspective. In Sec. 3.4, we will develop a Bayesian formulation built upon HM for federated penalized regression.

3.2.2 General Setting

We start by describing our problem setting. Suppose there exists $K \geq 2$ edge devices. For device $k \in [K] := \{1, \ldots, K\}$, the dataset is given as $\boldsymbol{D}_k = \{\boldsymbol{X}_k, \boldsymbol{Y}_k\}$ with N_k observations, where $\boldsymbol{Y}_k = [y_{k1}, \ldots, y_{kN_k}]^\intercal$ is a $N_k \times 1$ output vector, $\boldsymbol{X}_k = [\boldsymbol{x}_{k1}, \ldots, \boldsymbol{x}_{kN_k}]$ is a $d \times N_k$ input matrix and $\boldsymbol{x}_{k1} = ([\boldsymbol{x}_{k1}]_1, \ldots, [\boldsymbol{x}_{k1}]_d)^\intercal$. Here, d is the dimensionality of the input space. In this work, we focus on linear models. More specifically, data on device k is used to learn a linear model parameterized by $\boldsymbol{\theta}_k \in \mathbb{R}^d$. The distribution of y_{ki} is given as

$$y_{ki}|\boldsymbol{x}_{ki}, \boldsymbol{\theta}_k \sim \mathcal{N}(x_{ki}^\intercal \boldsymbol{\theta}_k, \sigma_k^2), \quad \forall i = 1, \ldots, N_k, \tag{3.1}$$

where σ_k^2 is a noise parameter. For the sake of compactness, denote by $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$ a $d \times K$ matrix concatenating all device parameters.

Further, we assume that a central server is connected to all edge devices and can facilitate the collaborative model learning process. As such, our goal in FDA is to let devices leverage their commonalities to better learn model parameters $\boldsymbol{\Theta}$; all while distributing the learning efforts and circumventing the need to share raw data.

3.2.3 Federated Data Analytics and Hierarchical Models

Since our goal is to borrow strength across devices, the first step is to create a shared representation across individual device models in order to facilitate the inductive transfer of knowledge. Here we adopt the natural hierarchy in FDA where a central server is connected to edge devices and can orchestrate the learning process. Specifically, we assume that individual device parameters $\boldsymbol{\theta}_k$ at the lower hierarchical level are generated from a set of shared parameters at the higher hierarchical level. Through collaboratively learning these shared parameters in a federated manner, devices induce an update on their personalized parameters $\boldsymbol{\theta}_k$ that uses information from all other devices.

Two hierarchical structures are proposed. The first defines a joint prior over $\boldsymbol{\Theta}$ parameterized by a cross-covariance matrix $\boldsymbol{\Omega}$. This allows learning a graph that achieves inductive transfer. Whereas, the second HM structure assumes that the $\boldsymbol{\theta}_k$'s are sampled from a common distribution (e.g., $\boldsymbol{\theta}_k|\boldsymbol{\phi} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$). This allows a Bayesian treatment capable of uncertainty quantification as well as learning a global random variable $\boldsymbol{\phi}$ that can be used to predict on new unseen devices.

We will detail our model formulations, inferences, and applications in the following two sections.

3.3 A Shared Representation via Correlation

In this section, we present our first hierarchical structure (denoted as `HM1`) that establishes a shared representation by defining a structural prior over $\boldsymbol{\Theta}$ (Figure 3.1). This prior is parameterized by $\mathbf{\Omega}$ - a $K \times K$ cross-covariance matrix. In Figure 3.1, the matrix $\mathbf{\Omega}$ acts as a graph on the central server that encodes a shared representation among the K devices and facilitates information sharing. By learning and exploiting the matrix $\mathbf{\Omega}$, devices can borrow information from each other to improve prediction performance.

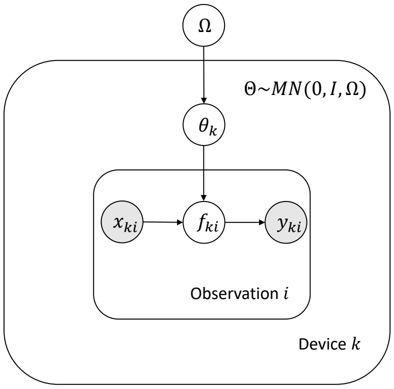

Figure 3.1: The first hierarchical model structure.

Mathematically, we impose a structural prior $\mathcal{N}(\mathbf{0}, \mathbf{\Omega} \otimes \boldsymbol{I})$ on Vec($\mathbf{\Theta}$), where Vec($\cdot$) is a vectorization operation and $\otimes$ is a Kronecker product. This prior encodes the belief on the underlying distribution that generates components of $\mathbf{\Theta}$. More specifically, $\mathbf{\Omega}$ is a symmetric matrix whose (i, j)-th component captures the covariance between device i and j. Overall, the aforementioned description translates to the following formulation:

$$\mathbf{\Theta} \sim \mathcal{MN}(\mathbf{0}, \boldsymbol{I}, \mathbf{\Omega}), \tag{3.2}$$

where $\mathcal{MN}(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{B})$ denotes a matrix normal distribution with location (mean) parameter $\boldsymbol{M}$, row covariance $\boldsymbol{A}$, and column covariance $\boldsymbol{B}$. In this prior, the column covariance $\mathbf{\Omega}$ captures the covariance across devices. Our prior assumes that the covariance across devices is the same for different parameter components. In fact, this is a common practice in multitask learning literature (Zhang and Yeung 2012, Ruder 2017) for multiple reasons: 1) The goal here is to facilitate information transfer amongst devices and $\mathbf{\Omega}$ achieves exactly this goal. 2) This is only a prior and, in reality, it is hard to pre-define the within component covariance. 3) Posterior computations become rather challenging if the prior (basically a regularization) is complex. As we will show later, a single $\mathbf{\Omega}$ is capable of providing excellent performance in several prediction tasks.

By Bayes' rule and incorporating Eqs. 3.1-3.2, we can obtain the posterior distribution of $\mathbf{\Theta}$ as a product of the prior and the likelihood function. By omitting the constant terms and taking the negative logarithm, we can obtain the negative log-likelihood function:

$$-\log p(\boldsymbol{\Theta}|\boldsymbol{\Omega}, \{\boldsymbol{Y}_k\}_{k=1}^{K}) \propto -\log p(\{\boldsymbol{Y}_k\}_{k=1}^{K}|\boldsymbol{\Theta})p(\boldsymbol{\Theta}|\boldsymbol{\Omega}) = -\log\left(\prod_{k=1}^{K} p(\boldsymbol{Y}_k|\boldsymbol{\theta}_k)\right) - \log p(\boldsymbol{\Theta}|\boldsymbol{\Omega})$$

$$\propto \sum_{k=1}^{K} \frac{1}{\sigma_k^2} \|\boldsymbol{Y}_k - \boldsymbol{X}_k^{\intercal}\boldsymbol{\theta}_k\|_2^2 + \mathrm{Tr}(\boldsymbol{\Theta}\boldsymbol{\Omega}^{-1}\boldsymbol{\Theta}^{\intercal}) + d\log|\boldsymbol{\Omega}| := L(\boldsymbol{\Theta}, \boldsymbol{\Omega}).$$

Therefore, our goal is to find the maximum a posteriori (MAP) of $(\Theta, \boldsymbol{\Omega})$ that minimizes the negative log-likelihood function, in a federated fashion:

$$(\boldsymbol{\Theta}^*, \boldsymbol{\Omega}^*) = \arg\min_{\boldsymbol{\Theta},\boldsymbol{\Omega}} L(\boldsymbol{\Theta}, \boldsymbol{\Omega}).$$

To solve this, notice that the derivative with respect to $\boldsymbol{\theta}_k$, for all k, is

$$\frac{\partial L(\boldsymbol{\Theta}, \boldsymbol{\Omega})}{\partial \boldsymbol{\theta}_k} = \frac{-2}{\sigma_k^2}\boldsymbol{X}_k(\boldsymbol{Y}_k - \boldsymbol{X}_k^{\intercal}\boldsymbol{\theta}_k) + 2\sum_{i=1}^{K} \boldsymbol{\theta}_i \boldsymbol{\Omega}_{i,k}^{-1},$$

where $\boldsymbol{\Omega}_{i,k}^{-1}$ is defined as the component located in the i-th row and k-th column of $\boldsymbol{\Omega}^{-1}$.

In IoFT, the central server does not have access to datasets $\boldsymbol{D} = \{\boldsymbol{D}_1, \cdots, \boldsymbol{D}_K\}$, nor do devices have access to each other's datasets. Further, the central server cannot share $\boldsymbol{\Omega}$ and Θ with any device, due to the privacy constraint. Therefore, directly running gradient descent using $\frac{\partial L}{\partial \boldsymbol{\theta}_k}$ is not feasible. Yet, by scrutinizing $\frac{\partial L}{\partial \boldsymbol{\theta}_k}$, one can observe that a gradient update on each $\boldsymbol{\theta}_k$ is split into two gradients. The first term is an update from local data $\boldsymbol{D}_k$ while the second is a regularization term from all devices based on $\boldsymbol{\Omega}$. Therefore, the local parameter update can be done via the two local steps below

$$\text{Stage 1: Multiple local GD or SGD steps } \boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k + 2\frac{\eta_1}{\sigma_k^2}\boldsymbol{X}_k(\boldsymbol{Y}_k - \boldsymbol{X}_k^{\intercal}\boldsymbol{\theta}_k).$$

$$\text{Stage 2: Prior Shrinkage } \boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k - 2\eta_2\sum_{i=1}^{K} \boldsymbol{\theta}_i \boldsymbol{\Omega}_{i,k}^{-1}.$$

At the first stage, device k runs multiple steps of stochastic gradient descent (SGD) or gradient descent (GD) using the local gradient information $\frac{-2}{\sigma_k^2}\boldsymbol{X}_k(\boldsymbol{Y}_k - \boldsymbol{X}_k^{\intercal}\boldsymbol{\theta}_k)$. To compute this gradient value, one needs to estimate the local variance parameter σ_k^2. Yet, recall that our main goal is to estimate $\boldsymbol{\theta}_k$ by borrowing information from the covariance matrix $\boldsymbol{\Omega}$. Adding to that, $\boldsymbol{\theta}_k$ and σ_k^2 are independent. Therefore, it is not necessary to estimate σ_k^2 at each local step. Here observe that η_1 is a tunable learning rate parameter and we can thus view $\eta_2 := \frac{\eta_1}{\sigma_k^2}$ as a tuning parameter in stage 1. In other words, we define η_2 as the tunable learning rate during the optimization procedure. This circumvents the need to estimate σ_k^2 locally. Nevertheless, σ_k^2 can be easily estimated from the linear residual term if it is of practitioner's interest. We here note that, since each device k has a different σ_k^2, it is possible to use device-specific learning rates η_{2k} for all k. However, it incurs a heavy tuning cost. In this paper, we use the same learning rate for each device.

At the second stage, device k will then use the aggregated information from all devices $\sum_{i=1}^K \boldsymbol{\theta}_i \boldsymbol{\Omega}_{i,k}^{-1}$ broadcasted from the central server to update $\boldsymbol{\theta}_k$ by exploiting the covariance matrix $\boldsymbol{\Omega}$. One key notable feature of this updating framework is that the central server only needs to share an aggregated metric $\sum_{i=1}^K \boldsymbol{\theta}_i \boldsymbol{\Omega}_{i,k}^{-1}$. This operation is indeed reminiscent of federated averaging and can preserve privacy while allowing devices to borrow strength from each other.

Finally, we will discuss the updating rule of $\boldsymbol{\Omega}$ on the central server. The most straightforward way is to take the derivative of $L(\boldsymbol{\Theta}, \boldsymbol{\Omega})$ with respect to $\boldsymbol{\Omega}$. By doing so, we obtain

$$\frac{\partial L(\boldsymbol{\Theta}, \boldsymbol{\Omega})}{\partial \boldsymbol{\Omega}} = -\boldsymbol{\Omega}^{-1}\boldsymbol{\Theta}^{\intercal}\boldsymbol{\Theta}\boldsymbol{\Omega}^{-1} + d\boldsymbol{\Omega}^{-1} = 0 \Rightarrow \boldsymbol{\Omega} = \frac{\boldsymbol{\Theta}^{\intercal}\boldsymbol{\Theta}}{d}.$$

As a result, it is natural to update $\boldsymbol{\Omega}$ using this closed-form expression. Here one can view that $\boldsymbol{\Theta}^{\intercal}\boldsymbol{\Theta}$ encodes the information of device covariance. Unfortunately, this approach typically faces singularity issues when $\boldsymbol{\Theta}^{\intercal}\boldsymbol{\Theta}$ is not a positive definite matrix (e.g., contains zero elements). To resolve this, we propose an updating procedure that prevents an abrupt change in $\boldsymbol{\Omega}$ to safeguard against singularity. More specifically, we express the updated $\boldsymbol{\Omega}$ as a convex combination between $\boldsymbol{\Omega}$ from the previous communication round and the exact updating equation $\frac{\boldsymbol{\Theta}^{\intercal}\boldsymbol{\Theta}}{d}$. That being said, we have

$$\boldsymbol{\Omega} \leftarrow (1-\alpha)\boldsymbol{\Omega} + \frac{\alpha}{d}\boldsymbol{\Theta}^{\intercal}\boldsymbol{\Theta}.$$

Here, α is a parameter that controls the change in $\boldsymbol{\Omega}$ and $\frac{\boldsymbol{\Theta}^{\intercal}\boldsymbol{\Theta}}{d}$ encodes the devices' covariance. A small α renders a conservative updating rule while a large α under-weights the importance of the covariance matrix from the previous communication round. When $\alpha = 1$, we recover the closed-form updating equation.

Here note that, in the aforementioned framework, the central server selects all devices at each communication round. This scheme is known as full device participation. In reality, however, some local devices are often offline or unwilling to respond due to various reasons. To accommodate this situation, one can sample a subset of devices at each communication round. We term this scenario as partial device participation. We summarize the detailed algorithm in Algorithm 3.1.

As we will show in our numerical studies, this simple-to-implement algorithm requires very few communication rounds to recover the true parameters and excels at leveraging knowledge across all devices.

Algorithm 3.1: Improving Device Performance by Exploiting Structural Covariance

```
Data: Number of devices $K$, Set $\mathcal{S}$ that contains indices of the selected devices, number of communication rounds $C$, randomly
      initialized model parameter $\boldsymbol{\Theta}$, initial matrix $\boldsymbol{\Omega} = \boldsymbol{I}$, learning rate $\eta_2$ (selected by grid-search or other tuning methods), proportion
      $\alpha = 0.1$ (default), dimension $d$.
for $c = 0 : (C-1)$ do
    Server broadcasts column of $\boldsymbol{\Theta}$ (i.e., $\boldsymbol{\theta}_k$) and $\sum_{i=1}^{K} \boldsymbol{\theta}_i \boldsymbol{\Omega}_{i,k}^{-1}$ for all selected $k$;
    for $k \in \mathcal{S}$ do
        for $t = 0 : (T-1)$ do
            Device-side: (Sampling Batch) Sample a subset of data $(\boldsymbol{X}_k^b, \boldsymbol{Y}_k^b)$ from $\boldsymbol{D}_k$, where superscript $b$ means batch;
            Device-side: (SGD or GD) $\boldsymbol{\theta}_k^{(t+1)} = \boldsymbol{\theta}_k^{(t)} + 2\eta_2 \boldsymbol{X}_k^b (\boldsymbol{Y}_k^b - \boldsymbol{X}_k^{b\intercal} \boldsymbol{\theta}_k^{(t)})$;
        end
        Device-side: (Prior Shrinkage) $\boldsymbol{\theta}_k^{new} \leftarrow \boldsymbol{\theta}_k^{(T)} - 2\eta_2 \sum_{i=1}^{K} \boldsymbol{\theta}_i \boldsymbol{\Omega}_{i,k}^{-1}$;
    end
    Server-side: Combine all $\boldsymbol{\theta}_k$, for $k \notin \mathcal{S}$, and all $\boldsymbol{\theta}_k^{new}$, for $k \in \mathcal{S}$, to create a new matrix $\boldsymbol{\Theta}$;
    Server-side: $\boldsymbol{\Omega} \leftarrow (1-\alpha)\boldsymbol{\Omega} + \frac{\alpha}{d} \boldsymbol{\Theta}^\intercal \boldsymbol{\Theta}$.
end
Return $\boldsymbol{\Theta}, \boldsymbol{\Omega}$.
```

3.4 A Hierarchical Model based on the Distribution Assumption

So far, we have presented HM1, which exploits the relationship among devices to improve prediction performance. One drawback of Algorithm 3.1 is that it only returns a point estimate of $\boldsymbol{\Theta}$. In practice, it is also desirable to quantify the uncertainty in the parameter estimates. Additionally, the estimated $\boldsymbol{\Theta}$ and $\boldsymbol{\Omega}$ cannot provide any borrowable information for new devices, yet the idea of fast adaptation to new unseen data is crucial in many fields such as meta-learning (Vanschoren 2019). In this section, we will present an alternative model structure that is formulated from a Bayesian perspective to tackle the aforementioned issues.

3.4.1 Structure and Formulation

Our second structure (denoted as HM2) assumes all device parameters are generated from the same underlying distribution. To give a simple example, one can assume $\boldsymbol{\theta}_k | \boldsymbol{\phi} \sim \mathcal{N}(\boldsymbol{\mu}, \tau \boldsymbol{I})$ (Figure 3.2) where $\boldsymbol{\phi} = (\boldsymbol{\mu}, \tau)$ is a set of a global hyper-parameters on the central server. Here it is critical to note that $\tau \boldsymbol{I} \in \mathbb{R}^{d \times d}$ is a within covariance matrix and does not denote covariances across K devices as all θ_k's come from the same underlying distribution. This assignment indicates that $\{\boldsymbol{\theta}_k\}_{k=1}^K$ are related and generated from the same distribution, yet the degree of model similarity is controlled by the variance parameter τ. A small $|\tau|$ implies all model parameters are similar (i.e., homogeneous) and the hierarchical model is closely related to learning a single common parameter $\boldsymbol{\theta}$ that fits all devices' data. On the other hand, a large variance $|\tau|$ incurs more heterogeneity among devices. In the extreme case when $|\tau| \to \infty$, the hierarchical model is equivalent to a separate modeling approach where each device's data is fitted separately (Albert and Hu 2019).

Now, we formally define the HM2 formulation. From our hierarchical definition and taking a fully Bayesian treatment by placing a prior $p(\boldsymbol{\phi})$ on the global hyper-parameters, the joint posterior

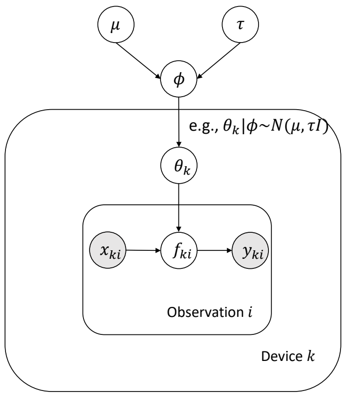

Figure 3.2: The second hierarchical model structure.

of $\{\boldsymbol{\theta}_k\}_{k=1}^K$ and $\boldsymbol{\phi}$ for `HM2` can be written as:

$$\begin{aligned} p(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \boldsymbol{\phi} | \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_K) &\propto p(\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_K | \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \boldsymbol{\phi}) p(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \boldsymbol{\phi}) \\ &= p(\boldsymbol{\phi}) \prod_{k=1}^{K} p(\boldsymbol{Y}_k | \boldsymbol{\theta}_k, \boldsymbol{\phi}) p(\boldsymbol{\theta}_k | \boldsymbol{\phi}) \,. \end{aligned} \tag{3.3}$$

To further contextualize `HM2`, we provide an example of a possible formulation.

Assume each device k fits a linear regression parameterized by $\boldsymbol{\theta}_k$. The local dataset is given as $(\boldsymbol{X}_k, \boldsymbol{Y}_k)$ for all k. Then one possible hierarchical formulation is:

$$\begin{aligned} \boldsymbol{Y}_k | \boldsymbol{\theta}_k, \boldsymbol{\phi} &\sim \mathcal{N}(\boldsymbol{X}_k^\intercal \boldsymbol{\theta}_k, \sigma_k^2 \boldsymbol{I}) \\ \boldsymbol{\theta}_k | \boldsymbol{\phi} &\sim \mathcal{N}(\boldsymbol{\mu}, \tau I) \\ \boldsymbol{\mu} &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \\ \tau &\sim \log \mathcal{N}(0, 1) \\ \boldsymbol{\phi} &= (\boldsymbol{\mu}, \tau). \end{aligned}$$

Clearly, inferring the joint posterior above is very challenging in a federated setting. Yet, in a hierarchical model, if we know the posterior over the upper hierarchical level $p(\boldsymbol{\phi}|\{\boldsymbol{Y}_k\}_{k=1}^K)$ (i.e., over global hyper-parameters $\boldsymbol{\phi}$), we can directly use this posterior as a prior to infer the lower level $p(\boldsymbol{\theta}_k|\{\boldsymbol{Y}_k\}_{k=1}^K)$ parameters locally.

More specifically, by integrating out device parameters from Eq. (3.3), we can derive that

$$\begin{aligned}
&\int p(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \boldsymbol{\phi} | \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_K) d\boldsymbol{\theta}_1 \ldots d\boldsymbol{\theta}_K \\
&= p(\boldsymbol{\phi} | \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_K) = p(\boldsymbol{\phi}) \prod_{k=1}^{K} \int p(\boldsymbol{Y}_k | \boldsymbol{\theta}_k, \boldsymbol{\phi}) p(\boldsymbol{\theta}_k | \boldsymbol{\phi}) d\boldsymbol{\theta}_k = p(\boldsymbol{\phi}) \prod_{k=1}^{K} f_k(\boldsymbol{\phi}),
\end{aligned}$$

where $f_k(\boldsymbol{\phi}) = \int p(\boldsymbol{Y}_k | \boldsymbol{\theta}_k, \boldsymbol{\phi}) p(\boldsymbol{\theta}_k | \boldsymbol{\phi}) d\boldsymbol{\theta}_k$. As a consequence, our main goal is to collaboratively learn $p(\boldsymbol{\phi} | \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_K)$ in a federated setting. One key challenge, however, is that the central server does not have any access to any edge dataset $\boldsymbol{D}_k$ and therefore directly computing $p(\boldsymbol{\phi} | \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_K)$ is infeasible. For this reason, we resort to a trick based on approximate inference methods to learn this posterior distribution.

3.4.2 Federated Bayesian Inference - Expectation Propagation

In this section, we will present the federated inference framework for `HM2`. Specifically, our goal is to learn the posterior density $p(\boldsymbol{\phi} | \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_K)$. In statistics, the most straightforward and popular approaches to do so are Markov chain Monte Carlo (MCMC) methods. Yet, as will be clear shortly, we argue that sampling methods are not practical in the federated hierarchical setting due to their sequential nature. Take Gibbs sampling as an example, device 1 needs to sample $\boldsymbol{\theta}_1$ from the density $p(\boldsymbol{\theta}_1 | \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K, \boldsymbol{\phi}, \boldsymbol{Y}_1)$ then passes those samples to the central server. The central server then needs to transmit those sampled $\boldsymbol{\theta}_1$ to device 2, and device 2 will sample from $p(\boldsymbol{\theta}_2 | \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \boldsymbol{\phi}, \boldsymbol{Y}_2)$. It can be seen that this sequential nature of MCMC significantly increases the communication cost and also slows down the federated optimization process when the total number of devices is large. Even if one can smartly parallelize the sampling process, the number of MCMC samples obtained locally will be large if the dimension is high due to the curse of dimensionality (Jordan et al. 2018).

To resolve the aforementioned issues, we resort to expectation propagation (EP) (Minka 2001) to approximate the posterior distribution. EP is one of the most widely-used algorithms for computing an approximate posterior distribution (Minka 2013, Vehtari et al. 2020). Here, we first briefly introduce the idea of EP in a centralized regime. Consider a posterior distribution with independent data points

$$\pi(\boldsymbol{\phi}) := p(\boldsymbol{\phi} | \boldsymbol{Y}) \propto p(\boldsymbol{\phi}) \prod_{i=1}^{N} p(y_i | \boldsymbol{\phi})$$

where $\boldsymbol{Y} = (y_1, \ldots, y_N)^\intercal$ is the data vector. EP approximates $\pi(\boldsymbol{\phi})$ by a density $q(\boldsymbol{\phi})$ such that

$$q(\boldsymbol{\phi}) = p(\boldsymbol{\phi}) \prod_{i=1}^{N} q_i(\boldsymbol{\phi}).$$

Intuitively, EP uses $q_i(\boldsymbol{\phi})$ to approximate $p(y_i|\boldsymbol{\phi})$, for all i. The most common choice is the normal density (Vehtari et al. 2020). To achieve this goal, at each iteration, EP first takes an approximation factor $q_i(\boldsymbol{\phi})$ out from the current $q(\boldsymbol{\phi})$ and replaces it with the true factor $p(y_i|\boldsymbol{\phi})$. This step yields a new density $q^{new}(\boldsymbol{\phi})$. This resulting new density can be used as an updated approximated posterior. This step is iterated over all i till convergence. Please, refer to Barthelme (2016) for a comprehensive summary of EP. It can be seen that EP can be naturally extended to FDA where each device can be viewed as an independent "data point". In the following paragraphs, we will detail the federated extension of EP.

The main idea is to approximate terms $f_k(\boldsymbol{\phi})$ by a local device approximation function $q_k(\boldsymbol{\phi})$ for all $k = 1, \ldots, K$. More specifically, we have

$$p(\boldsymbol{\phi}|\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_K) \approx p(\boldsymbol{\phi}) \prod_{k=1}^{K} q_k(\boldsymbol{\phi}) := q(\boldsymbol{\phi}). \tag{3.4}$$

Using the framework of EP, we gradually update $q(\boldsymbol{\phi})$ by iteratively renewing $q_k(\boldsymbol{\phi})$ at each device k. During each communication round, given the estimated $q(\boldsymbol{\phi})$ broadcasted from the server, device k first computes the cavity distribution

$$q_{-k}(\boldsymbol{\phi}) \propto \frac{q(\boldsymbol{\phi})}{q_k(\boldsymbol{\phi})} \tag{3.5}$$

and the tilted distribution

$$q_{\backslash k}(\boldsymbol{\phi}) \propto f_k(\boldsymbol{\phi}) q_{-k}(\boldsymbol{\phi}). \tag{3.6}$$

It then computes the updated posterior approximation such that

$$q^{new}(\boldsymbol{\phi}) \approx q_{\backslash k}(\boldsymbol{\phi}). \tag{3.7}$$

Intuitively, the cavity distribution $q_{-k}(\boldsymbol{\phi})$ removes the impact of the old $q_k(\boldsymbol{\phi})$ from the approximated posterior density $q(\boldsymbol{\phi})$ and the tilted distribution adds the true target density $f_k(\boldsymbol{\phi})$ to $q_{-k}(\boldsymbol{\phi})$. As a result, we use the tilted distribution as an updated approximation to the posterior density of $\boldsymbol{\phi}$. This step is typically done through a sampling method and is distribution-dependent. We will detail this approximation procedure in Sec. 3.4.2.2.

Afterward, device k calculates the change in its local approximation by

$$\Delta q_k(\boldsymbol{\phi}) = \frac{q^{new}(\boldsymbol{\phi})}{q(\boldsymbol{\phi})}. \tag{3.8}$$

Instead of sending q^{new} back to the server, we calculate the change in the global posterior imposed by device k via Eq. (3.8) and sends $\Delta q_k(\boldsymbol{\phi})$ to the central server. The server aggregates all device approximations by

$$q(\boldsymbol{\phi}) \leftarrow q(\boldsymbol{\phi}) \prod_{k=1}^{K} \Delta q_k(\boldsymbol{\phi}). \tag{3.9}$$

We summarize the EP algorithm in Algorithm 3.2.

Algorithm 3.2: The Federated Expectation Propagation Algorithm

```
Data: number of devices $K$, Set $\mathcal{S}$ that contains indices of the selected devices, number of communication rounds $C$, initial
      approximation $\{q_k(\boldsymbol{\phi})\}_{k=1}^{K}$, prior $p(\boldsymbol{\phi})$, learning rate $\eta$ (Selected by grid-search)
for $c = 0 : (C - 1)$ do
    Server broadcasts $q(\boldsymbol{\phi})$;
    for $k \in \mathcal{S}$ do
        Device-side: Calculate the cavity distribution $q_{-k}(\boldsymbol{\phi})$ using Eq. (3.5);
        Device-side: Calculate the tilted distribution $q_{\backslash k}(\boldsymbol{\phi})$ using Eq. (3.6);
        Device-side: Get new $q(\boldsymbol{\phi})$ from the tilted distribution using Eq. (3.7);
        Device-side: Calculate $\Delta q_k(\boldsymbol{\phi})$ using Eq. (3.8) and update local $q_k(\boldsymbol{\phi})$ ;
        Device-side: Send $\Delta q_k(\boldsymbol{\phi})$ to the central server;
    end
    Server-side: Update $q(\boldsymbol{\phi})$ using Eq. (3.9);
end
Return $q(\boldsymbol{\phi})$.
```

3.4.2.1 Posterior of Device Parameters

Once we obtain $q(\boldsymbol{\phi})$ that approximates $p(\boldsymbol{\phi}|\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_k)$, we can further estimate the posterior of device parameters. Specifically, given a device k,

$$\begin{aligned} p(\boldsymbol{\theta}_k|\{\boldsymbol{Y}_k\}_{k=1}^K) &= \int_{\boldsymbol{\phi}} \int_{\boldsymbol{\theta}_j, j\neq k} p(\boldsymbol{\theta}_k, \boldsymbol{\phi}|\{\boldsymbol{Y}_k\}_{k=1}^K) d\boldsymbol{\theta}_j d\boldsymbol{\phi} \\ &\propto \int p(\boldsymbol{\phi}) p(\boldsymbol{Y}_k|\boldsymbol{\theta}_k, \boldsymbol{\phi}) p(\boldsymbol{\theta}_k|\boldsymbol{\phi}) \prod_{j\neq k} \int p(\boldsymbol{Y}_j|\boldsymbol{\theta}_j, \boldsymbol{\phi}) p(\boldsymbol{\theta}_j|\boldsymbol{\phi}) d\boldsymbol{\theta}_j d\boldsymbol{\phi} \\ &\approx \int q_{-k}(\boldsymbol{\phi}) p(\boldsymbol{Y}_k|\boldsymbol{\theta}_k, \boldsymbol{\phi}) p(\boldsymbol{\theta}_k|\boldsymbol{\phi}) d\boldsymbol{\phi}. \end{aligned}$$

As a consequence, we can use the posterior of $\boldsymbol{\theta}_k$ to quantify uncertainties or conduct hypothesis testing. The posterior samples from $p(\boldsymbol{\theta}_k|\{\boldsymbol{Y}_k\}_{k=1}^K)$ can be obtained by off-the-shelf posterior sampling techniques (see `mcmc` package in R or `NumPyro` library in Python). Here we provide a simpler sampling trick. In the above equation, if we ignore the integral, we can obtain the joint posterior

$$p(\boldsymbol{\theta}_k, \boldsymbol{\phi} | \{\boldsymbol{Y}_k\}_{k=1}^K) \approx q_{-k}(\boldsymbol{\phi}) p(\boldsymbol{Y}_k | \boldsymbol{\theta}_k, \boldsymbol{\phi}) p(\boldsymbol{\theta}_k | \boldsymbol{\phi}). \tag{3.10}$$

As a result, one can ignore the integration and use sampling methods to jointly sample $(\boldsymbol{\theta}_k, \boldsymbol{\phi})$ from Eq. (3.10) and discard $\boldsymbol{\phi}$. Now, given M samples $\{\boldsymbol{\theta}_{ki}\}_{i=1}^M$, we can readily use the samples to estimate moments, coverage probability, and do hypothesis tests.

Here we should note that the estimated posterior density $q(\boldsymbol{\phi})$ encodes crucial information across all devices (recall the explanation in Sec. 3.4.1). One can exploit this information for a new device to achieve fast adaption. For example, we can treat the posterior mean of $\boldsymbol{\phi}$ as an initial model parameter for a new device. This idea is similar to meta-learning (Vanschoren 2019), where one tries to learn a global model that can quickly adapt to a new task.

3.4.2.2 Normal Approximation

In practice, it is common to model $p(\boldsymbol{\phi})$ and $q_k(\boldsymbol{\phi}), \forall k$ as normal densities. This is due to a very useful property of normal random variables.

Lemma 16. *(Williams and Rasmussen 2006) Suppose there are two normal random variables (with the same dimension) such that $\boldsymbol{\theta}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\boldsymbol{\theta}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$. Let $\boldsymbol{r}_i = \boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i$, $\boldsymbol{Q}_i = \boldsymbol{\Sigma}_i^{-1}$ for $i = 1, 2$. Define $p(\boldsymbol{\theta}_+) = p(\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_2)$ and $p(\boldsymbol{\theta}_-) = \frac{p(\boldsymbol{\theta}_1)}{p(\boldsymbol{\theta}_2)}$. We have that*

$$\boldsymbol{\theta}_+ \sim \mathcal{N}(\boldsymbol{r}_1 + \boldsymbol{r}_2, \boldsymbol{Q}_1 + \boldsymbol{Q}_2)$$
$$\boldsymbol{\theta}_- \sim \mathcal{N}(\boldsymbol{r}_1 - \boldsymbol{r}_2, \boldsymbol{Q}_1 - \boldsymbol{Q}_2).$$

Lemma 16 states that the product of two Gaussian densities gives another unnormalized Gaussian density. We use θ_+ to represent this new Gaussian random variable. Similarly, the quotient of two Gaussian densities gives an unnormalized Gaussian density and we use θ_- to represent this new random variable.

Using Lemma 16, one can efficiently implement the EP algorithm, as all components in Algorithm 3.2 can be computed in closed forms. Here, we detail the implementation technique. We model the prior of $\boldsymbol{\phi}$ as a multivariate normal random variable with mean $\boldsymbol{\mu}_0$ and variance $\boldsymbol{\Sigma}_0$. We also assume $q_k(\boldsymbol{\phi})$ has a normal density parameterized by $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$, for all k. If the support of some components in $\boldsymbol{\phi}$ does not lie in $\mathbb{R}$, one can always perform a logarithmic or logistic transformation to those components. By Gaussian properties, Eq. (3.4) can be computed in closed-form such that $q(\boldsymbol{\phi})$ has a normal density parametrized by mean $\boldsymbol{r}_0 + \sum_{k=1}^K \boldsymbol{r}_k$ and variance $\boldsymbol{Q}_0 + \sum_{k=1}^K \boldsymbol{Q}_k$, where $\boldsymbol{r}_j = \boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j$ and $\boldsymbol{Q}_j = \boldsymbol{\Sigma}_j^{-1}$, for all $j = 0, 1, \ldots, K$. Similarly, by Lemma 16, the cavity distribution in Eq. (3.5) can be computed in a closed-form by a subtraction operation. This results in $\boldsymbol{r}_{-k} \coloneqq \boldsymbol{r} - \boldsymbol{r}_k, \boldsymbol{Q}_{-k} \coloneqq \boldsymbol{Q} - \boldsymbol{Q}_k$. Compared to sampling approaches, one key advantage of EP is that the computation and communication steps are simple and efficient. The central server and devices only need to transmit the mean vector and variance matrix to perform model updating and aggregation. Notably, during each communication round, there is no need to estimate or transmit normalizing constants. We detail this idea in Algorithm 3.3. In Algorithm 3.3, one needs to compute the tilted distribution and obtain $\boldsymbol{r}_k^{new}, \boldsymbol{Q}_k^{new}$. These steps correspond to Eqs. (3.6)-(3.7). However, as $f_k(\boldsymbol{\phi})$ may not be a normal density, the resulting tilted distribution is not normal. Therefore, we need to use a normal distribution to approximate the tilted distribution. To do so, we can run a set of simulation draws (using any sampling method coded in R or Python libraries) from $f_k(\boldsymbol{\phi})q_{-k}(\boldsymbol{\phi})$ and estimate the mean and covariance of those draws. We set the resulting mean to be $\boldsymbol{\mu}_{\setminus k}$ and the resulting covariance to be $\boldsymbol{\Sigma}_{\setminus k}$.

Algorithm 3.3: The Federated Expectation Propagation Algorithm using Normal Approximation

```
Data: number of devices $K$, Set $\mathcal{S}$ that contains indices of the selected devices, number of communication rounds $C$, initial
      approximation $\{\boldsymbol{r}_k, \boldsymbol{Q}_k\}_{k=1}^K$, prior $\boldsymbol{r}_0, \boldsymbol{Q}_0$, initial posterior parameters $\boldsymbol{r} = \boldsymbol{r}_0 + \sum_{k=1}^K \boldsymbol{r}_k, \boldsymbol{Q} = \boldsymbol{Q}_0 + \sum_{k=1}^K \boldsymbol{Q}_k$, learning
      rate $\eta$ (Selected by grid-search)
for $c = 0 : (C - 1)$ do
    Server broadcasts $\boldsymbol{r}, \boldsymbol{Q}$;
    for $k \in \mathcal{S}$ do
        Device-side: Calculate the cavity distribution with parameters $\boldsymbol{r}_{-k} \coloneqq \boldsymbol{r} - \boldsymbol{r}_k, \boldsymbol{Q}_{-k} \coloneqq \boldsymbol{Q} - \boldsymbol{Q}_k$;
        Device-side: Calculate the tilted distribution $\boldsymbol{r}_{\setminus k}, \boldsymbol{Q}_{\setminus k}$ ;
        Device-side: Obtain new $\boldsymbol{r}_k^{new}, \boldsymbol{Q}_k^{new}$;
        Device-side: Calculate $\Delta\boldsymbol{r}_k = \boldsymbol{r}_k^{new} - \boldsymbol{r}_k, \Delta\boldsymbol{Q}_k = \boldsymbol{Q}_k^{new} - \boldsymbol{Q}_k$ ;
        Device-side: Send $\Delta\boldsymbol{r}_k, \Delta\boldsymbol{Q}_k$ to the central server;
    end
    Server-side: Update $\boldsymbol{r} = \boldsymbol{r} + \sum_{k\in\mathcal{S}} \Delta\boldsymbol{r}_k, \boldsymbol{Q} = \boldsymbol{Q} + \sum_{k\in\mathcal{S}} \Delta\boldsymbol{Q}_k$;
end
Return $\boldsymbol{r}, \boldsymbol{Q}$.
```

3.4.3 Federated & Penalized Regression for Variable Selection

To move a step further, we ask "is it possible to let devices exploit the shared representation structure to perform variable selection?" Indeed, there are some attempts to tackle this question from a frequentist perspective. Yuan et al. (2021) develop a federated composite optimization framework that solves the federated lasso problem. Tong et al. (2020) propose a federated iterative hard thresholding algorithm to tackle non-convex penalized regression. Despite these few efforts in exploring variable selection from a frequentist perspective, no literature exists in the Bayesian setting.

Tibshirani (1996) has shown that a Lasso estimate can be achieved when the regression parameters have *i.i.d.* Laplace priors. Since then, researchers have started to build Bayesian priors for many other penalized regressions such as the elastic net and fussed Lasso. Please refer to the work of Van Erp et al. (2019) for a detailed literature review. Inspired by the Bayesian interpretation of penalized regressions, we develop a hierarchical structure, based upon `HM2`, to perform federated variable selection. To proceed, we impose priors on $\boldsymbol{\theta}_k$, for all k, such that

$$\theta_{ki}|\boldsymbol{\phi} \sim \pi(\lambda, \sigma^2), \forall i = 1, \ldots, d$$

where $\boldsymbol{\phi} = (\lambda, \sigma^2)$, λ is a regularization parameter and σ^2 is a variance parameter. Here, $\pi(\lambda, \sigma^2)$ is a distribution parameterized by λ, σ^2. For instance, if we set $\pi(\lambda, \sigma^2)$ to be a Laplace distribution with zero mean and $\frac{\sigma}{\lambda}$ diversity, then we recover the Bayesian counterpart of Lasso regression (Tibshirani 1996). Another example is if we set $\pi(\lambda, \sigma^2)$ to be $\mathcal{N}(0, \frac{\sigma}{\lambda})$, then we recover Ridge regression. There are many possible choices of prior beliefs on σ^2 and λ. In this work, we impose log-normal priors on σ^2 and λ (Van Erp et al. 2019) and we set $\boldsymbol{\phi} = (\log \lambda, \log \sigma^2)$. Our framework can flexibly incorporate other priors such as a non-informative prior on σ^2 or a half-Cauchy prior on λ. In this work, we will use a log-normal prior as an illustrative example.

The posterior distribution of $\boldsymbol{\theta}_k$, for all k, and $\boldsymbol{\phi}$ can be learned by Algorithm 3.2. Here, one caveat is that, unlike frequentist penalized regressions, the Bayesian methods do not shrink regression coefficients to be exactly zero. As a result, we will calculate the credible interval (CI) for each parameter. If the CI of a parameter, say θ_{ki}, covers 0, we will exclude this predictor.

3.5 Simulation Studies

3.5.1 HM1 Proof of Concept using Algorithm 3.1

Case I: We assume that $K = 2$ and generate $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ from a matrix-variate normal distribution with zero mean and $\boldsymbol{I} \otimes \boldsymbol{\Omega} = \boldsymbol{I} \otimes \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$ covariance. The input space dimension is $d = 5$. Data on each device are generated from linear models using the generated parameters. We set noise to be 0.05. To demonstrate the benefits of our correlation-based construction in HM1, we create imbalanced sample sizes on the devices. Specifically, device 1 only has $N_1 = 20$ data points and device 2 has $N_2 = 200$ data points. We train Algorithm 3.1 with $C = 30, \eta_2 = 0.01, \alpha = 0.1$ and we set the number of local steps T to be 20. We compare the performance of Algorithm 3.1 with a separate modeling approach where each device fits its own model without communication. Specifically, each device runs 600 local SGD steps with learning rate 0.01.

Case II: We set $K = 100$ and generate a 100×100 positive definite matrix $\boldsymbol{\Omega}$ using the R package `clusterGeneration`. We then generate true device parameters based on the matrix $\boldsymbol{\Omega}$. We set $d = 8$ and generate data using the linear models with noise $\sigma^2 = 0.1$. For the first 30 devices, we assign 40 data points and for the remaining 70 devices, we assign 275 data points. Overall, we create an imbalanced data generation scenario. Case II can be viewed as a generalization of Case I with more devices. Again, we train models using the same hyperparameters as those in Case I.

Case III: We use the same setting as the one in Case II, but all devices have 20 data points (i.e., balanced data generation).

Case IV: We increase the sample size on each device to 200 and use the same setting as the one in Case III.

Performance Evaluation: Denote by $(\boldsymbol{X}_k^*, \boldsymbol{Y}_k^*)$ the testing dataset on device k where $\boldsymbol{X}_k^* = [x_{k1}^*, \ldots, x_{kN_k^*}^*]^\intercal$ and $\boldsymbol{Y}_k^* = [y_{k1}^*, \ldots, y_{kN_k^*}^*]$. The averaged Root-mean-square error (A-RMSE) across all devices is defined as

$$\text{A-RMSE} = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{\sum_{i=1}^{N_k^*}(f(x_{ki}^*) - y_{ki}^*)^2}{N_k^*}},$$

where $f_k(x^*) = x^{*\intercal}\boldsymbol{\theta}_k$. On each device, we generate $1,000$ data points using the true device parameters for testing. In Case I, the RMSE is calculated on device 1. In Case II, the A-RMSE is calculated using devices $k \in \{1, \ldots, 30\}$. Here our goal is to assess the prediction accuracy on devices with scant data. In Cases III/IV, we calculate A-RMSE using all 100 devices. We report our results in Table 3.1. It can be seen that Algorithm 3.1 yields much smaller A-RMSE under the imbalanced data scenario. This conveys the importance of borrowing strength from other devices under the FDA framework. In Case III, the local sample size is not enough such that each model alone cannot perform well. However, our FDA can still benefit devices' training by borrowing information from other devices. In case IV, Algorithm 3.1 does not offer a major improvement as all local devices have enough data. In this case, doing local training without collaboration should be sufficient.

Table 3.1: The A-RMSE of our proposed model and the separate model over 30 independent runs. We report standard deviations of A-RMSEs in brackets.

Case	Algorithm 3.1	Separate
I	$0.081(\pm 0.001)$	$0.094(\pm 0.001)$
II	$0.050(\pm 0.000)$	$0.056(\pm 0.001)$
III	$0.044(\pm 0.002)$	$0.072(\pm 0.004)$
IV	$0.035(\pm 0.000)$	$0.035(\pm 0.000)$

Accuracy of Parameter Estimation: The negative log-likelihood function $L(\boldsymbol{\Theta}, \boldsymbol{\Omega})$ is a non-convex function of $(\boldsymbol{\Theta}, \boldsymbol{\Omega})$. To test the impact of the initialization, we conducted a sensitivity analysis below. Denote by $\hat{\boldsymbol{\Theta}}$ the concatenated estimated device parameters and $\boldsymbol{\Theta}^*$ the concatenated true data-generating parameters. In Figure 3.3, we plotted $\frac{\|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*\|}{\sqrt{K}}$ versus communication round for Case II and III over 30 independent runs. Each independent run used a different initialized $\boldsymbol{\Theta}$. More specifically, we first generate $d * K$ random numbers from a standard normal distribution.

We then create a $d * K$ matrix $\boldsymbol{\Theta}$ using these random numbers. It can be seen that Algorithm 3.1 accurately recovers the true underlying model parameters. Furthermore, it can be observed that Algorithm 3.1 typically converges within 30-40 communication rounds. We observed that, in all simulations, Algorithm 3.1 could be trained within 5 seconds on a standard laptop. In conclusion, our proposed algorithm is easy to implement and optimize.

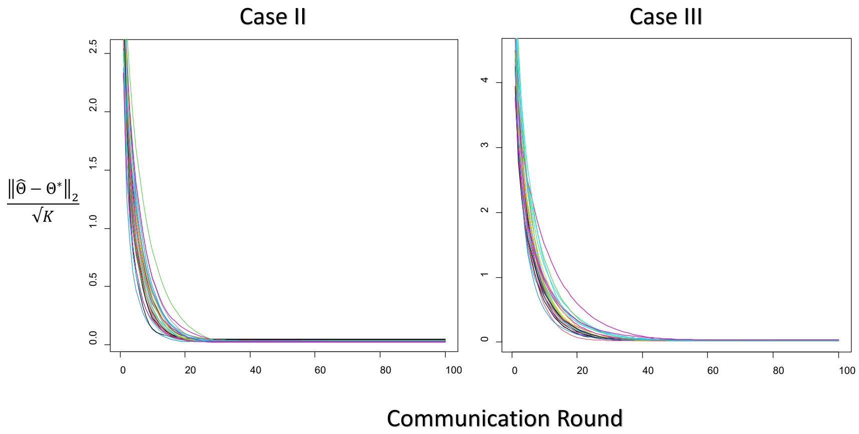

Figure 3.3: Plot of $\frac{\|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*\|}{\sqrt{K}}$ versus communication round. Each color represents one independent run.

3.5.2 HM2 Proof of Concept

In this section, we test the variable selection performance of `HM2`. Following the examples in Van Erp et al. (2019), we create several simulation cases below:

Case I: We set $K = 10$, $\boldsymbol{\theta}_{true} = (3, 1.5, 0, 0, 2, 0, 0, 0)^\intercal$ and generate all columns of $\{\boldsymbol{X}_k\}_{k=1}^K$ from a standard multivariate normal distribution. We then generate $\{\boldsymbol{Y}_k\}_{k=1}^K$ using $\boldsymbol{\theta}_{true}$ and $\{\boldsymbol{X}_k\}_{k=1}^K$ for all k. We set the noise to 0.05. Each device has 100 data points for training and 1000 data points for testing.

Case II: We use the same setting as the one in Case I. The difference is that the first 2 devices only have 20 data points each, while the other devices have 200 data points each. The number of testing data points is 1000.

Case III: We set $K = 20$, $\boldsymbol{\theta}_{true} = (\underbrace{3, \ldots, 3,}_{10} \underbrace{0, \ldots, 0,}_{10} \underbrace{3, \ldots, 3}_{10})^\intercal$. Each device has 40 observations for training and 400 observations for testing.

We evaluate the performance of our model based on prediction and variable selection accuracy. The prediction accuracy is evaluated by A-RMSE. Variable selection accuracy is based on the averaged correct and false inclusion rates. To decide whether to include a variable or not, we first calculate a 90% credible interval for each parameter. If the CI covers 0, we will exclude this predictor. Results are reported in Table 3.2. It can be seen that our proposed federated variable selection methods can correctly identify more than 85% of effective predictors while maintaining low false inclusion rates.

Table 3.2: The A-RMSE and averaged correct/false inclusion rates for different federated variable selection methods over 30 experimental runs.

Methods	A-RMSE	Averaged Correct Inclusion Rate	Averaged False Inclusion Rate
Lasso (HM2, Case I)	0.055(±0.001)	0.880(±0.003)	0.095(±0.001)
Lasso (HM2, Case II)	0.062(0.002)	0.875(±0.004)	0.101(±0.001)
Lasso (HM2, Case III)	0.088(0.001)	0.891(±0.005)	0.115(±0.001)

As mentioned in Sec. 3.4, one advantage of HM2 is that it can provide uncertainty quantification (UQ) for parameter estimation. We will provide two examples to demonstrate the UQ capability of HM2.

1. We collect the estimated posterior for parameters $\boldsymbol{\theta}_1$ from an independent run in case I and calculate the mean and 90% credible interval. The resulting plot is presented in Figure 3.4 (Left). It can be seen that the true parameters are included in the confidence interval generated by HM2.

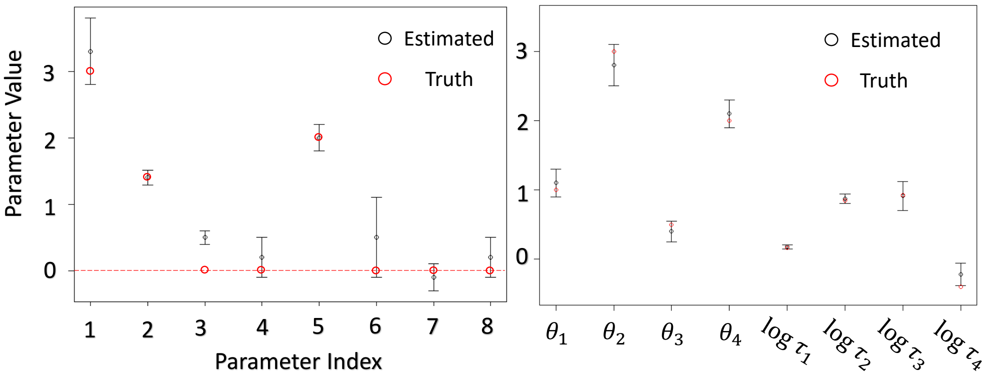

Figure 3.4: Plot of parameter estimation and confidence interval.

2. We create $K = 100$ devices and generate device parameter

$$\boldsymbol{\theta}_k|\boldsymbol{\phi} \sim \mathcal{N}(\boldsymbol{\mu}_{true}, \boldsymbol{\Sigma}_{true}) := \mathcal{N}\left(\begin{bmatrix} 1 \\ 3 \\ 0.5 \\ 2 \end{bmatrix}, \begin{bmatrix} 1.17 & 0 & 0 & 0 \\ 0 & 2.35 & 0 & 0 \\ 0 & 0 & 2.52 & 0 \\ 0 & 0 & 0 & 0.67 \end{bmatrix}\right)$$

for $k \in \{1, \ldots, 100\}$. We then use $\boldsymbol{\theta}_k$ to generate 100 data points for each device k. Our HM structure can be summarized as follows.

$$\begin{aligned}
\boldsymbol{Y}_k|\boldsymbol{\theta}_k, \boldsymbol{\phi} &\sim \mathcal{N}(\boldsymbol{X}_k^\intercal \boldsymbol{\theta}_k, \sigma_k^2 \boldsymbol{I}) \\
\boldsymbol{\theta}_k|\boldsymbol{\phi} &\sim \mathcal{N}(\boldsymbol{\mu}, \text{diag}(\tau_1, \dots, \tau_4)) \\
\boldsymbol{\mu} &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \\
\tau_i &\sim \log \mathcal{N}(0, 1), \forall i \\
\boldsymbol{\phi} &= (\boldsymbol{\mu}, \log \tau_1, \dots, \log \tau_4).
\end{aligned}$$

We calculate the posterior distribution of $\boldsymbol{\phi}$ using Algorithm 3.3 and plot the mean and 90% CI for each component in Figure 3.4 (Right). It can be seen that the mean of the posterior of $\boldsymbol{\phi}$ is close to the truth and the 90% credible interval also covers the true parameter. This estimated $q(\boldsymbol{\phi})$ can be used as an initialization for new devices to achieve fast adaption.

3.6 Real-World Case Studies

3.6.1 Student Performance Dataset

This is a public dataset that can be found at `archive.ics.uci.edu/ml/index.php`. The dataset contains information on student performance (measured by exam scores) in secondary education of two Portuguese schools, namely, Gabriel Pereira and Mousinho da Silveira. It includes 29 predictors covering gender, grades, demographic, and many other social/school-related features. Detailed information can be found in Cortez and Silva (2008). We treat each school as a "device" (i.e., $K = 2$). On each device, we randomly pick 60% of the students as the training dataset and another 40% of the students as the testing dataset. We create dummy variables for all nominal variables, such as job and guardian. All other numeric variables are standardized to a zero mean and one standard deviation, following the guide in Cortez and Silva (2008). This data processing yields 38 predictors.

Our first goal is to select relevant predictors using our federated penalized regression technique (See Sec. 3.4.3). We then use the selected predictors to predict the final exam grades of students. We consider the two most widely-used variable selection methods: Lasso and Ridge. Results are reported in Table 3.3. The model performance is evaluated based on the RMSE and the number of included predictors.

It can be seen that the variable selection performance of `HM2` is consistent with the centralized variable selection method such as Lasso and Ridge regressions. This implies that our framework can serve as a new paradigm for decentralized variable selection problems. Additionally, `HM2` also yields comparable A-RMSEs compared to centralized methods. This demonstrates the advantage of borrowing strength from other devices. However, please note that, in terms of prediction accuracy,

Table 3.3: The RMSE and number of included predictors for different federated variable selection methods.

Methods	A-RMSE	# included predictors (School 1)	# included predictors (School 2)
Lasso (HM2)	0.825	21	23
Ridge (HM2)	0.817	21	22
Methods	**RMSE**	**# included predictors**	
Lasso (Centralized)	0.820	21	
Ridge (Centralized)	0.815	21	

federated variable selection can rarely beat the centralized approach as the latter uses more data.

3.6.2 NASA Aircraft Gas Turbine Engines

In this case study, we consider condition monitoring data generated from aircraft gas turbine engines using the NASA commercial Modular Aero-Propulsion System Simulation (C-MAPSS) tools. The dataset is available at `https://ti.arc.nasa.gov/tech/dash/groups/pcoe/`. This dataset contains 100 engines. In each engine, 24 sensors are installed to collect time-series degradation signals. For each engine, we treat the first 60% of the time-series observations as the training dataset and the remaining 40% of the signals as the testing dataset. Within the training dataset, we sample 20% of the data as a validation dataset. Our goal is therefore to predict the sensor signal trajectory on each gas turbine engine by training Algorithm 3.1 using the training dataset. In this scenario, each engine can be viewed as a device (i.e., $K = 100$).

It can be observed that all signal trajectories exhibit polynomial patterns and therefore, many existing works resort to polynomial regression to analyze this dataset (Liu et al. 2013, Song and Liu 2018). Here we detail the modeling procedure. Given a specific sensor, for all $k \in \{1, \dots, K\}$, device k fits a $d = 6$-th order polynomial regression in the form of

$$\boldsymbol{Y}_k = \boldsymbol{X}_k^\intercal \boldsymbol{\theta}_k + \text{noise},$$

where the $(d+1) \times N_k$ design matrix $\boldsymbol{X}_k$ is in the form of

$$\boldsymbol{X}_k^\intercal = \begin{bmatrix} 1 & [x_{k1}]_1 & [x_{k1}]_2^2 & \dots & [x_{k1}]_d^d \\ 1 & [x_{k2}]_1 & [x_{k2}]_2^2 & \dots & [x_{k2}]_d^d \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & [x_{kN_k}]_1 & [x_{kN_k}]_2^2 & \dots & [x_{kN_k}]_d^d \end{bmatrix}.$$

In the above expression, $\boldsymbol{Y}_k$ represents the signal trajectory for device k and $\boldsymbol{x}_{k1}$ represents time. In HM1, device parameters $\{\boldsymbol{\theta}_k\}_k$ are estimated using Algorithm 3.1.

We benchmark our proposed model with the following algorithms:

- FedAvg: FedAvg is one of the most fundamental and competing benchmark models in the FDA. During each communication round, device k, for all selected k, first runs several steps of local SGD and then sends updated parameters $\boldsymbol{\theta}_k$ back to the server. The server the aggregates those parameters by calculating $\bar{\boldsymbol{\theta}} = \frac{1}{|\mathcal{S}|} \sum_{k \in \mathcal{S}} \boldsymbol{\theta}_k$. This step is repeated several times and ultimately, each device will use the global parameter $\bar{\boldsymbol{\theta}}$ to perform prediction.

- `Dis-Ridge` (Zhang et al. 2015): `Dis-Ridge` is a distributed ridge regression method. The server first evenly divides a set of data into m disjoint sets and assigns each set to a different node. Each node then solves a ridge regression problem, by optimizing $\frac{1}{N_k} \sum_{i=1}^{N_k} (y_{ki} - \boldsymbol{x}_{ki}^{\intercal} \boldsymbol{\theta}_k)^2 + \lambda \|\boldsymbol{\theta}_k\|^2$, and sends optimal solution back to the server. The server then aggregates local estimations.

- `Ditto`: `Ditto` is a personalized FL algorithm. The first stage of `Ditto` is the same as `FedAvg` and generates a global parameter $\bar{\boldsymbol{\theta}}$. Afterwards, each device k derives the personalized solution $\boldsymbol{v}_k$ by solving a constrained optimization problem $F_k(\boldsymbol{v}_k) + \frac{\lambda_{Ditto}}{2} \left\| \boldsymbol{v}_k - \bar{\boldsymbol{\theta}} \right\|_2^2$ where $F_k(\cdot)$ is the local loss function and λ_{Ditto} is a tuning parameter. The intuition is that each device can run several updating procedures to collect personalized solutions while this solution stays in the vicinity of the shared global model to retain useful information from a global model.

- `Separate`: In `Separate`, each device simply fits its own linear model without communication.

For all models, we set $T = 20, C = 100$, and use grid-search to tune the learning rate (and other model hyper-parameters). In Algorithm 3.1, we set $\alpha = 0.9$. We report the A-RMSE across all 100 devices in Table 3.4.

Table 3.4: The A-RMSE of all models over 30 independent runs. We report the standard deviation in the brackets.

Sensor	`HM1` ($\alpha = 0.9$)	`Separate`	`FedAvg`	`Ditto`	`Dis-Ridge`
Sensor 2	$\mathbf{0.270(\pm 0.001)}$	$0.299(\pm 0.003)$	$0.450(\pm 0.013)$	$0.281(\pm 0.001)$	$0.552(\pm 0.002)$
Sensor 3	$\mathbf{0.218(\pm 0.002)}$	$0.223(\pm 0.001)$	$0.303(\pm 0.009)$	$0.220(\pm 0.001)$	$0.288(\pm 0.002)$
Sensor 7	$\mathbf{0.369(\pm 0.004)}$	$0.405(\pm 0.003)$	$0.628(\pm 0.011)$	$0.388(\pm 0.005)$	$0.605(\pm 0.001)$
Sensor 8	$\mathbf{0.267(\pm 0.001)}$	$0.307(\pm 0.001)$	$0.395(\pm 0.008)$	$0.289(\pm 0.001)$	$0.390(\pm 0.003)$

From Table 3.4, it can be seen that `FedAvg` and `Dis-Ridge` consistently yield the worst prediction accuracy as one shared global parameter $\bar{\boldsymbol{\theta}}$ does not suit all devices, especially in a heterogeneous setting. Personalized approaches such as `Ditto` circumvent this disadvantage of global models and generate personalized solutions for each device. Those personalized methods, however, ignore related information amongst devices. Our method, on the other hand, improves the prediction accuracy by exploiting a joint structure for inductive transfer.

3.7 Conclusion

This paper proposes a federated treatment for linear regression by adopting a hierarchical modeling approach. We test our proposed framework on a range of simulated and real-world datasets. Despite the simplicity of our linear model framework, it can outperform many state-of-the-art federated algorithms and we argue that it can serve as a competing benchmark model for the future development of federated algorithms. One possible future direction is to extend our framework to generalized linear models such as linear mixed-effect models or to more complicated models such as Gaussian processes and tensor regression. We hope our work will help inspire continued exploration into the world of federated data analytics and its engineering applications.

CHAPTER 4

Federated Gaussian Process: Convergence, Automatic Personalization and Multi-fidelity Modeling

In this chapter, we propose `FGPR`: a Federated Gaussian process ($\mathcal{GP}$) regression framework that uses an averaging strategy for model aggregation and stochastic gradient descent for local computations. Notably, the resulting global model excels in personalization as `FGPR` jointly learns a shared prior across all devices. The predictive posterior then is obtained by exploiting this shared prior and conditioning on local data, which encodes personalized features from a specific dataset. Theoretically, we show that `FGPR` converges to a critical point of the full log-marginal likelihood function, subject to statistical errors. This result offers standalone value as it brings federated learning theoretical results to correlated paradigms. Through extensive case studies, we show that `FGPR` excels in a wide range of applications and is a promising approach for privacy-preserving multi-fidelity data modeling.

4.1 Introductory Remarks

The modern era of computing is gradually shifting from a centralized regime where data is stored in a centralized location, often a cloud or central server, to a decentralized paradigm that allows devices to collaboratively learn models while keeping their data stored locally (Kontar et al. 2021). This paradigm shift was set forth by the massive increase in compute resources at the edge device and is based on one simple idea: instead of learning models on a central server, edge devices execute small computations locally and only share the minimum information needed to learn a model. This modern paradigm is often coined as federated learning (FL). Though the prototypical idea of FL dates back decades ago, to the early work of Mangasarian and Solodov (1994), it was only brought to the forefront of deep learning after the seminal paper by McMahan et al. (2017). In their work, McMahan et al. (2017) propose Federated Averaging (`FedAvg`) for decentralized learning of a deep learning model. In `FedAvg`, a central server broadcasts the network architecture and a global model (e.g., initial weights) to selected devices; devices perform local computations (using stochastic gradient descent - SGD) to update the global model based on their local data, and the central server then takes an average of the resulting local models to update the global model. This process is iterated until an accuracy criterion is met.

Despite the simplicity of taking averages of local estimators in deep learning, `FedAvg` (McMahan et al. 2017) has seen immense success and has since generated an explosive interest in FL. To date, `FedAvg` for decentralized learning of deep neural networks (NN) was tailored to image classification, text prediction, wireless network analysis, and condition monitoring & failure detection (Smith et al. 2018, Brisimi et al. 2018, Tran et al. 2019, Kim et al. 2019, Yue and Kontar 2019, Li et al. 2020d). Besides that, building upon `FedAvg`'s success, literature has been proposed to: (i) tackle adversarial attacks in FL (Bhagoji et al. 2019, Wang et al. 2020b); (ii) allow personalization whereby each device retains its own individualized model (Li et al. 2021); (iii) ensure fairness in performance and participation across devices (Li et al. 2019a, Mohri et al. 2019, Yue et al. 2021, 2022b, Zeng et al. 2021); (iv) develop more complex aggregation strategies that accommodate deep convolution network (Wang et al. 2020a); (v) accelerate FL algorithms to improve convergence rate or reduce communication cost (Karimireddy et al. 2020, Yuan and Ma 2020); (vi) improve generalization through model ensembling (Shi et al. 2021).

Despite the aforementioned ubiquitous application of FL, most, if not all, FL literature lies within an empirical risk minimization (ERM) framework - a direct consequence of the focus on deep learning. To date, very few papers study FL beyond ERM, specifically when correlation exists. In this paper, we go beyond ERM and focus on the Gaussian process ($\mathcal{GP}$). We investigate both theoretically and empirically the (i) plausibility of federating model/parameter estimation in $\mathcal{GP}$s and (ii) applications where federated $\mathcal{GP}$s can be of immense value. Needless to say, the inherent capability to encode correlation, quantify uncertainty, and incorporate highly flexible model priors has rendered $\mathcal{GP}$s a key inference tool in various domains such as multi-fidelity modeling, experimental design (Rana et al. 2017, Yue and Kontar 2020a, Jiang et al. 2020c, Krishna et al. 2021, Yue and Kontar 2021), manufacturing (Tapia et al. 2016, Peng et al. 2017, Yue and Kontar 2020b, Chung and Kontar 2020, Chung et al. 2022a, Yue and Al Kontar 2023), healthcare (Imani et al. 2018, Ketu and Mishra 2021, Chung et al. 2022b), autonomous vehicles (Goli et al. 2018), energy (Liu et al. 2022) and robotics (Deisenroth et al. 2013, Jang et al. 2020). Therefore, the success of FL within $\mathcal{GP}$s may help pave the way for FL to infiltrate many new applications and domains.

The central challenge is that, unlike empirical risk minimization (see Sec. 4.3 for a formal definition), $\mathcal{GP}$s feature correlations across all data points such that any finite collection of which has a joint Gaussian distribution (Sacks et al. 1989, Currin et al. 1991). As a result, the objective function does not simply sum over the loss of individual data points. Adding to that, mini-batch gradients become biased estimators when correlation exists. The performance of FL in such a setting is yet to be understood and explored.

To this end, we propose `FGPR`: a **F**ederated $\boldsymbol{\mathcal{GP}}$ **R**egression framework that uses `FedAvg` (i.e., averaging strategy) for model aggregation and SGD for local devices computations. First, we show that, under some conditions, `FGPR` converges to a critical point of the full log-marginal likelihood function and recovers true parameters (or minimizes the global objective function) up to statistical errors that depend on the device's mini-batch size. Our results hold for kernel functions that exhibit exponential or polynomial eigendecay, which is satisfied by a wide range of kernels commonly used in $\mathcal{GP}$s such as the Matérn and radial basis function (RBF) kernels. Our proof offers standalone value as it is the first to extend the theoretical results of FL beyond ERM and to a correlated paradigm. In turn, this may help researchers further investigate FL within alternative stochastic processes built upon correlations, such as Lévy processes. Second, we explore `FGPR` within various applications to validate our results. Most notably, we propose `FGPR` as a privacy-preserving approach for multi-fidelity data modeling and show its advantageous properties compared to the state-of-the-art benchmarks. In addition, we find an interesting yet unsurprising observation. The global model in `FGPR` excels in personalization. This feature is due to the fact that ultimately `FGPR` learns a shared prior across all devices. The predictive posterior then is obtained by exploiting this shared prior and conditioning on local data, which encodes personalized features from a specific device. This notion of automatic personalization is closely related to meta-learning, where the goal is to learn a model that can achieve fast personalization.

4.1.1 Summary of Contributions & Findings

We briefly summarize our contributions below:

- **Convergence:** We explore two data-generating scenarios. (1) Homogeneous setting where local data is generated from the same underlying distribution or stochastic process across all devices; (2) Heterogeneous setting where devices have distributional differences. Under both scenarios and for a large enough batch size M, we prove that `FGPR` converges to a critical point of the full log-marginal likelihood function (from all data) for kernels that exhibit an exponential or polynomial eigendecay. We also provide uniform error bounds on parameter estimation errors and highlight the ability of `FGPR` to recover the underlying noise variance.
  - Interestingly, our derived bounds not only depend on iteration T, but also explicitly depend on batch size M, which is a direct consequence of correlation. Our results do not assume any specific functional structure, such as convexity, Lipschitz continuity, or bounded variance.
- **Automatic Personalization Capability:** We demonstrate that `FGPR` can automatically personalize the shared global model to each local device. Learning a global model by `FGPR` can be viewed as jointly learning a global $\mathcal{GP}$ prior. On the other hand, the posterior predictive

distribution of a $\mathcal{GP}$ depends both on this shared prior and the local training data. The latter one can be viewed as a personalized feature encoded in the $\mathcal{GP}$ model. This important personalization feature allows FGPR to excel in the scenario where data among each local device is heterogeneous (Sec. 4.6 and Sec. 4.7).

  - In addition to the personalization capability, we find that the prior class learned from FGPR excels in transfer learning (Appendix). This idea is similar to meta-learning, where one tries to learn a global model that can quickly adapt to a new task.

- **Multi-fidelity modeling and other applications:** We propose FGPR as a privacy-preserving approach for multi-fidelity data modeling, which combines datasets of varying fidelities into one unified model. We find that in such settings, not only does FGPR preserve privacy but also can improve generalization power across various existing state-of-the-art multi-fidelity and distributed learning (DL) approaches. We also validate FGPR on various simulated datasets and real-world datasets to highlight its advantageous properties.

The remainder of this paper is organized as follows. A detailed literature review can be found in Sec. 4.2. In Sec. 4.3, we present the FGPR algorithm. We study the theoretical properties of FGPR in Sec. 4.4. In Sec. 4.5-4.7, we present several empirical results over a range of simulated datasets and real-world datasets. We conclude our paper in Sec. 4.8 with a brief discussion.

4.2 Related Work

4.2.1 Federated Learning

Most of the existing FL literature has focused on developing deep learning algorithms and their applications in image classification and natural language processing. Please refer to (Kontar et al. 2021) for an in-depth review of FL literature. Here we briefly review some related papers that tackle data heterogeneity. One popular trend (Li et al. 2018a, Zhang et al. 2020d, Pathak and Wainwright 2020) uses regularization techniques to allay heterogeneity. For instance, FedProx (Li et al. 2018a) adds a quadratic regularizer to the device objective to limit the impact of heterogeneity by penalizing local updates that move far from the global model. Alternatively, personalized models were proposed. Such models usually follow an alternating train-then-personalize approach where a global model is learned, and the personalized model is regularized to stay within its vicinity (Kirkpatrick et al. 2017, Dinh et al. 2020, Li et al. 2021). Other approaches (Arivazhagan et al. 2019, Liang et al. 2020) use different layers of a network to represent global and personalized solutions. More recently, researchers have tried to remove the dependence on a global model for personalization by following a multi-task learning philosophy (Smith et al. 2017). Yet, such models can only handle simple convex formulations.

4.2.2 Distributed Learning

Table 4.1: Comparison between benchmark DL methods and our proposed FL approach. For Modular $\mathcal{GP}$, sparse representation of data entails the pseudo-targets, variational density, model parameters, and approximate likelihood value.

Models	**Theory**	**Inference**	**Comm. Frequency**	**Comm. Load**
DVI (Gal et al. 2014)	✗	Approximate	Every Iteration	Gradient Tensor
DGP (Deisenroth and Ng 2015)	✗	Approximate	One-Shot	Predicted Output
Modular $\mathcal{GP}$ (Moreno-Muñoz et al. 2021)	✗	Approximate	One-Shot	Sparse Representation of Data
FGPR	✓	Exact	Multiple local steps	Model Parameters

Our work focuses on developing federated models, specifically for $\mathcal{GP}$s, that go beyond deep learning. To date, little to no literature exists along this line. Perhaps, the closest field where various regression approaches were investigated is DL for distributed systems. Distributed approaches for MCMC, $\mathcal{GP}$s, PCA, logistic, and quantile regression have been proposed (Zhang et al. 2013, Wang and Dunson 2013, Lee et al. 2017, Lin et al. 2017, Chen et al. 2019, 2021a,b, Fan et al. 2021). However, DL and FL have several fundamental differences.

Distributed learning is a centralized computation approach where devices are compute nodes connected by large bandwidth. Nodes can communicate often and access any part of a dataset, as data partitions can be continuously adjusted. DL aims to parallelize computation tasks across different compute nodes to improve computational efficiency. In FL, data resides at the edge where the goal is to process more of the data at the origin of creation (the edge) and only share updated model parameters rather than entire datasets. In FL, we do not have the luxury to partition, shuffle, and randomize the data. In essence, each device in FL has its model, and all devices borrow strength from each other to improve model learning. One critical bottleneck in FL is communication (Zheng et al. 2020). Unlike centralized regimes, aggregation of local models cannot be done after every single optimization iteration, as this incurs huge communication needs between edge devices and the central server. Instead, each device runs multiple local optimization iterates before uploading the data. Indeed, the FedAvg algorithm that we discussed earlier (McMahan et al. 2017) was motivated by the ability to perform multiple optimization iterations locally before updating the global model - hence reducing communication needs. Interestingly, the number of local updates cannot be very large, as we will discuss in Sec. 4.4.

Along the line of DL, distributed $\mathcal{GP}$s are closely related to our proposed algorithm FGPR. (Cao and Fleet 2014) proposed a distributed $\mathcal{GP}$ approach that uses the product-of-experts (PoE) approximation (Ng and Deisenroth 2014) to partition a central dataset into several blocks so that the inference can be made in a distributed fashion. This approach often overestimates predictive variance. (Deisenroth and Ng 2015) proposed a new distributed $\mathcal{GP}$ counterpart (denoted as DGP) that alleviates the aforementioned drawback. The product-of-experts approximation assumes the independence of local experts and therefore ignores correlation among them. (Tavassolipour et al.

2019) overcame the limitation of PoE using vector quantization to learn correlation among experts. However, the proposed approach requires different nodes to transmit data to each other. (Gal et al. 2014) resorted to variational inference (VI) to approximate the $\mathcal{GP}$ marginal likelihood function and developed a distributed variational inference (DVI) framework that parallelizes inference procedures. (Moreno-Muñoz et al. 2021) developed a Modular $\mathcal{GP}$ that extended the VI-based framework into a multi-output scenario where one can model data from multiple sources. Due to space limitation, please refer to (Chen et al. 2022) for a comprehensive review of the distributed $\mathcal{GP}$ methods.

That said, the methods described above and our approach FGPR feature key differences. The differences are highlighted in Table 4.1.

First, DGP and Modular $\mathcal{GP}$ are one-shot approaches. Whereas our model FGPR is a collaborative process where the global model is updated over multiple communication rounds. In FL, a one-shot approach, where each device trains till convergence and then model aggregation happens, is sub-optimal. This is due to the well-known "Client-drift" phenomenon (Karimireddy et al. 2020) where many local steps can push the local solutions to different neighborhoods, and then the aggregation becomes sub-optimal, often giving meaningless predictions. This has also been shown from a theoretical perspective. For instance, in FedAvg, the number of local optimization steps at each communication round should be less than the order of communication rounds for convergence. A similar result is shown for our model in Sec. 4.4. Whilst Modular $\mathcal{GP}$ and DGP require only one-shot communication, DVI requires communication after every single optimization iterate. This is clearly not viable in FL. Adding to that, DVI needs to send a high-dimension tensor to a central server. This further amplifies communication loads and costs. FGPR, on the other hand, only shares model parameters.

Second, FGPR aims to optimize the exact marginal likelihood function. We alleviate the computational burden by using mini-batch SGD and accordingly show convergence on the exact likelihood. In contrast, DGP, DVI, and Modular $\mathcal{GP}$ are approximate inference methods that approximate the exact likelihood function and optimize the approximate objectives.

Third, our paper presents the first successful try at extending the theoretical results of FL beyond ERM and to a correlated paradigm, while existing work (Gal et al. 2014, Deisenroth and Ng 2015, Moreno-Muñoz et al. 2021) did not study the theoretical properties of their proposed algorithms.

More detailed explanations that shed light on the differences between all models can be found in our experiments (Sec. 4.6 and Sec. 4.7), where we benchmark our approach with DGP, DVI, and the Modular $\mathcal{GP}$ amongst others.

4.3 The FGPR Algorithm

In this section, we describe the problem setting in Sec. 4.3.1 and introduce `FGPR` - a federated learning scheme for $\mathcal{GP}$s in Sec. 4.3.2. We then provide insights on the advantages of `FGPR` in Sec. 4.3.3. Specifically, we will show that `FGPR` is capable of automatically personalizing the global model to each local device. This property allows `FGPR` to excel in many real-world applications, such as multi-fidelity modeling, where heterogeneity exists.

4.3.1 Background

We consider the Gaussian process regression. We first briefly review the centralized $\mathcal{GP}$ model. Suppose the training dataset is given as $D = \{\boldsymbol{X}, \boldsymbol{y}\}$, where $\boldsymbol{y} = [y_1, ..., y_N]^\intercal$, $\boldsymbol{X} = [x_1^\intercal, ..., x_N^\intercal]$ and N denotes the number of observations. In this paper, we use $|D| = N$ to denote the cardinality of set D. Here, $x \in \mathbb{R}^d$ is a d-dimensional input and $y \in \mathbb{R}$ is the output. We decompose the output as $y_i = f(x_i) + \epsilon_i$, where

$$f \sim \mathcal{GP}(0, \mathcal{K}(\cdot, \cdot; \boldsymbol{\theta}_{\mathcal{K}})), \quad \epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2),$$

and $\mathcal{K}(\cdot, \cdot; \boldsymbol{\theta}_{\mathcal{K}})$ is the prior kernel function parameterized by kernel parameters $\boldsymbol{\theta}_{\mathcal{K}}$. The prior encodes a belief about the data-generating process and incurs correlations across all data points.

Given a new observation x^*, the goal of $\mathcal{GP}$ regression is to predict $f(x^*)$. By definition, any finite collection of observations from a $\mathcal{GP}$ follows a multivariate normal distribution. Therefore, the joint distribution of $\boldsymbol{y}$ and $f(x^*)$ is given as

$$\begin{bmatrix} \boldsymbol{y} \\ f(x^*) \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I} & \boldsymbol{K}(\boldsymbol{X}, x^*) \\ \boldsymbol{K}(x^*, \boldsymbol{X}) & \boldsymbol{K}(x^*, x^*) \end{bmatrix}\right)$$

where $\boldsymbol{K}(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a covariance matrix whose entries are determined by the kernel function $\mathcal{K}(\cdot, \cdot; \boldsymbol{\theta}_{\mathcal{K}})$. Therefore, the conditional distribution (also known as the posterior predictive distribution) of $f(x^*)$ is given as $\mathcal{N}\left(\mu_{pred}(x^*), \sigma^2_{pred}(x^*)\right)$, where

$$\begin{aligned} &\mu_{pred}(x^*) = \boldsymbol{K}(x^*, \boldsymbol{X}) \left(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{y}, \\ &\sigma^2_{pred}(x^*) \\ &= \boldsymbol{K}(x^*, x^*) - \boldsymbol{K}(x^*, \boldsymbol{X}) \left(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{K}(\boldsymbol{X}, x^*). \end{aligned} \tag{4.1}$$

Here, $\mu_{pred}(x^*)$ is often used as a point estimate of $f(x^*)$ and $\sigma^2_{pred}(x^*)$ quantifies the variance. It can be seen that our predictions will depend on the kernel parameters that parameterize $\boldsymbol{K}(\cdot, \cdot)$ and on the noise parameter σ^2. In this paper, we denote by $\boldsymbol{\theta} := (\boldsymbol{\theta}_{\mathcal{K}}, \sigma^2)$ the $\mathcal{GP}$ model parameters.

Therefore, predicting an accurate output $f(x^*)$ critically depends on finding a good estimate of $\boldsymbol{\theta}$. To estimate $\boldsymbol{\theta}$, the most popular approach is to minimize the negative log-marginal likelihood in the form of

$$\begin{aligned} -\log p(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\theta}) &= -\log \int p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{f};\boldsymbol{\theta})p(\boldsymbol{f}|\boldsymbol{X};\boldsymbol{\theta})df \\ &= \frac{1}{2}[\boldsymbol{y}^\intercal(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y} \\ &\quad + \log\left|\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})+\sigma^2\boldsymbol{I}\right| + N\log(2\pi)], \end{aligned} \tag{4.2}$$

where $\boldsymbol{f} = (f(x_1),\ldots,f(x_N))$, $\boldsymbol{y}|\boldsymbol{X},\boldsymbol{f} \sim \mathcal{N}(0,\sigma^2\boldsymbol{I})$ and $p(\boldsymbol{f}|\boldsymbol{X};\boldsymbol{\theta})$ is a prior density function. There are numerous optimizers that are readily available to minimize $-\log p(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\theta})$. In this paper, we resort to stochastic optimization methods such as SGD or Adam (Kingma and Ba 2014).

Remark 17. *Needless to say, a current critical challenge in FL is that edge devices have limited compute power. SGD offers an excellent scalability solution to the computational complexity of $\mathcal{GP}s$, which has been a long-standing bottleneck since $\mathcal{GP}s$ require inverting a covariance matrix $\boldsymbol{K}(\cdot,\cdot)$ at each iteration of an optimization procedure (see Eq. (4.2)). This operation, in general, incurs a $\mathcal{O}(N^3)$ time complexity. In SGD, only a mini-batch with a size of $M \ll N$ is taken at each iteration; hence allowing $\mathcal{GP}s$ to scale to big data regimes. Besides that, and as will become clear shortly, our approach only requires edge devices to do a few steps on SGD on their local data. Another notable advantage of SGD is that it offers good generalization power (Keskar et al. 2016, Gnanasambandam et al. 2022). In deep learning, it is well-known that SGD can drive solutions to a flat minimizer that generalizes well (Wu et al. 2018). Although this statement is still an open problem in $\mathcal{GP}$, Chen et al. (2020) empirically validate that the solution obtained by SGD generalizes better than other deterministic optimizers.*

In the non-federated setting, applying stochastic inference to $\mathcal{GP}$ is not new. Indeed, prior work (Hensman et al. 2013) introduced $N_z < N$ inducing points and employed stochastic VI that optimizes an approximate log-marginal likelihood function. As a result, the computation burden is reduced to $\mathcal{O}(N_z^2 N)$. Unfortunately, (Stein 2014, Burt et al. 2019) show that the VI approximation does not work well when the underlying process is not smooth and requires many inducing points to achieve a satisfactory approximation accuracy. Even for a smooth kernel such as the RBF kernel, $\mathcal{O}(\log^d N)$ inducing points are needed. On the other hand, our work directly applies SGD to the exact log-marginal likelihood function without using an approximation. In Sec. 4.4, we also support our approach with theoretical guarantees.

Now to use SGD on the exact log-marginal likelihood in Eq. (4.2) in a centralized regime, we can derive the stochastic gradient given mini-batch of size M as

$$
\begin{aligned}
& g(\boldsymbol{\theta}; \xi) \\
&= \frac{1}{2}\Bigg[-\boldsymbol{y}_\xi^\intercal \boldsymbol{K}^{-1}(\boldsymbol{X}_\xi, \boldsymbol{X}_\xi)\frac{\partial \boldsymbol{K}(\boldsymbol{X}_\xi, \boldsymbol{X}_\xi)}{\partial \theta}\boldsymbol{K}^{-1}(\boldsymbol{X}_\xi, \boldsymbol{X}_\xi)\boldsymbol{y}_\xi \\
&\qquad + \operatorname{Tr}\left(\boldsymbol{K}^{-1}(\boldsymbol{X}_\xi, \boldsymbol{X}_\xi)\frac{\partial \boldsymbol{K}(\boldsymbol{X}_\xi, \boldsymbol{X}_\xi)}{\partial \theta}\right)\Bigg] \\
&= \frac{\operatorname{Tr}\left[\boldsymbol{K}^{-1}(\boldsymbol{X}_\xi, \boldsymbol{X}_\xi)\left(\boldsymbol{I} - \boldsymbol{y}_\xi \boldsymbol{y}_\xi^T \boldsymbol{K}^{-1}(\boldsymbol{X}_\xi, \boldsymbol{X}_\xi)\right)\frac{\partial \boldsymbol{K}(\boldsymbol{X}_\xi, \boldsymbol{X}_\xi)}{\partial \theta}\right]}{2},
\end{aligned}
$$

where ξ is the set of indices corresponding to a subset of training data with mini-batch size M and $\boldsymbol{X}_\xi, \boldsymbol{y}_\xi$ is the respective subset of inputs and outputs indexed by ξ. At each iteration t, a subset of training data is taken to update model parameters as

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta^{(t)} g(\boldsymbol{\theta}^{(t)}; \xi^{(t)}),$$

where $\eta^{(t)}$ is the learning rate at iteration t. This step is repeated several times till some exit condition is met.

Although SGD was a key propeller for deep learning, it faces a fundamental challenge in $\mathcal{GP}$s. In deep learning, the empirical risk function is given as $\hat{R}(\boldsymbol{\theta}; D) \approx \frac{1}{N}\sum_{i=1}^{N} \ell(f_{\boldsymbol{\theta}}(x_i), y_i)$, where $D = \{(x_i, y_i)\}_{i=1}^{N}$ is the training dataset, $f_{\boldsymbol{\theta}}$ is the neural network to be learned and $\ell(\cdot, \cdot)$ is a loss function. Therefore, the stochastic gradient of $\hat{R}(\boldsymbol{\theta}; D)$ evaluated using a batch of data, indexed by a set ξ, is $\nabla \hat{R}_k(\boldsymbol{\theta}; \xi) = \frac{1}{|\xi|}\sum_{i \in \xi} \ell(f_{\boldsymbol{\theta}}(x_i), y_i)$. As a result, $\mathbb{E}[\nabla \hat{R}(\boldsymbol{\theta}; \xi)] = \nabla \hat{R}(\boldsymbol{\theta}; D)$, which means the stochastic gradient is an unbiased estimator of the full gradient. This is a direct consequence of the fact that the objective $\hat{R}(\boldsymbol{\theta}; D)$ is given as a summation over the training data. On the other hand, $\mathcal{GP}$s feature correlations where any finite collection of data points has a joint Gaussian distribution. Therefore, the objective, $-\log p(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\theta})$, to be minimized in a $\mathcal{GP}$ does not simply sum over individual data points. Consequently, stochastic gradients become biased estimators when correlation exists. Mathematically, this implies $\mathbb{E}[g(\boldsymbol{\theta}; \xi)] \neq \nabla(-\log p(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\theta}))$.

Despite this challenge, we will show in the following sections that our federated SGD approach for learning a $\mathcal{GP}$ converges to a critical point of $\log p(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\theta})$, subject to statistical errors.

4.3.2 The `FGPR` Framework

Suppose there exists $K \geq 2$ local devices. In this paper, we will use (edge) devices and clients interchangeably. For client $k \in [K]$, the local dataset is given as $D_k = \{\boldsymbol{X}_k, \boldsymbol{y}_k\}$ with cardinality N_k. We let $N = \sum_{k=1}^{K} N_k$. Denote by $L_k(\boldsymbol{\theta}; D_k) := -\log p(\boldsymbol{y}_k|\boldsymbol{X}_k; \boldsymbol{\theta})$ the negative log-marginal likelihood function for device k and $g_k(\boldsymbol{\theta}; \xi_k)$ the SG of this negative log-marginal function with respect to a mini-batch of size M indexed by ξ.

In FL, our goal is to collaboratively learn a global parameter $\boldsymbol{\theta}$ that minimizes the global objective function in the form of

$$L(\boldsymbol{\theta}) := \sum_{k=1}^{K} p_k L_k(\boldsymbol{\theta}; D_k) \tag{4.3}$$

where $p_k = \frac{N_k}{\sum_{k=1}^{K} N_k}$ is the weight parameter for device k such that $\sum_{k=1}^{K} p_k = 1$. To fulfill this goal, during each communication period, each local device k runs E steps of SGD and updates model parameters as

$$\boldsymbol{\theta}_k^{(t+1)} \leftarrow \boldsymbol{\theta}_k^{(t)} - \eta^{(t)} g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}).$$

At the end of each communication round, the central server aggregates model parameters as

$$\bar{\boldsymbol{\theta}} = \sum_{k=1}^{K} p_k \boldsymbol{\theta}_k.$$

The aggregated parameter $\bar{\boldsymbol{\theta}}$ is then distributed back to local devices. This cycle is repeated several times till convergence. In this training framework, all devices participate during each communication round. We define this framework as synchronous updating. In reality, however, some local devices are frequently offline or reluctant/slow to respond due to various unexpected reasons. To resolve this issue, we develop an asynchronous updating scheme. Specifically, at the beginning of each communication round (c), we select $K_{\text{sample}} \in [1, K)$ clients by sampling probability p_k and denote by $\mathcal{S}$ the indices of these clients. During the communication round, the central server aggregates model parameter as

$$\bar{\boldsymbol{\theta}} = \frac{1}{K_{sample}} \sum_{k \in \mathcal{S}} \boldsymbol{\theta}_k.$$

The detailed procedure is given in Algorithm 4.1.

Remark 18. *The aggregation strategy used in Algorithm 4.1 is known as* `FedAvg` *(McMahan et al. 2017). Despite being the first proposed aggregation scheme for FL,* `FedAvg` *has stood the test in the past couple of years as one of the most robust and competitive approaches for model aggregation. That being said, it is also possible to extend our algorithm to different strategies, such as different sampling or weighting schemes.*

Algorithm 4.1: The `FGPR` algorithm

```
Data: number of sampled devices $K_{\text{sample}}$, number of communication rounds $R$, initial model parameter $\boldsymbol{\theta}$
for $c = 0 : (R - 1)$ do
    Select $K_{\text{sample}}$ clients by sampling probability $p_k$ and denote by $\mathcal{S}$ the indices of these clients;
    Server broadcasts $\boldsymbol{\theta}$;
    for $k \in \mathcal{S}$ do
        $\boldsymbol{\theta}_k^{(0)} = \boldsymbol{\theta}$;
        Update model parameter (e.g., using Algorithm 4.2);
    end
    Aggregation $\bar{\boldsymbol{\theta}}_c = \frac{1}{K_{\text{sample}}} \sum_{k \in \mathcal{S}} \boldsymbol{\theta}_k^{(E)}$, Set $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}_c$;
end
Return $\bar{\boldsymbol{\theta}}_R$.
```

Algorithm 4.2: Local update using SGD

```
Data: index of device $k$, number of local updates $E$, SGD learning rate schedule $\{\eta^{(t)}\}_{t=1}^{E}$, initial model parameter $\boldsymbol{\theta}_k^{(0)}$
for $t = 0 : (E - 1)$ do
    Randomly sample a subset of data from $D_k$ and denote it as $\xi_k^{(t)}$;
    $\boldsymbol{\theta}_k^{(t+1)} = \boldsymbol{\theta}_k^{(t)} - \eta^{(t)} g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)})$ ;
end
Return $\boldsymbol{\theta}_k^{(E)}$;
```

4.3.3 Why a Single Global $\mathcal{GP}$ Model Works?

In this paper, we will demonstrate the viability of `FGPR` in cases where data across devices are both homogeneous or heterogeneous. In heterogeneous settings, it is often the case that personalized FL approaches are developed where clients eventually retain their own models while borrowing strength from one another. Popular personalization methods usually fine-tune the global model based on local data while encouraging local weights to stay in a small region in the parameter space of the global model (Li et al. 2021). This allows a balance between the client's shared knowledge and unique characteristic. This literature, however, is mainly focused on deep learning.

One natural question is: why does a single global model learned from Algorithm 4.1 work in `FGPR`? Here it is critical to note that, unlike deep learning, estimating $\boldsymbol{\theta}$ in a $\mathcal{GP}$ is equivalent to learning a prior through which predictions are obtained by conditioning on the observed data, and the learned prior. Here, by "learning a prior", we refer to estimating hyper-parameters of $\mathcal{GP}$s by maximizing the global objective.

More specifically, in the $\mathcal{GP}$, we impose a prior on f_k such that $f_k \sim \mathcal{GP}(0, \mathcal{K}(\cdot,\cdot;\boldsymbol{\theta}_{\mathcal{K}}))$. The kernel function is parameterized by $\boldsymbol{\theta}_{\mathcal{K}}$. Therefore, learning a global model by `FGPR` can be viewed as learning a common model prior over $f_k, \forall k$. On the other hand, the posterior predictive distribution at a testing point x^* is given as

$$\begin{aligned} p(f_k^*|\boldsymbol{X}_k, \boldsymbol{y}_k, x^*) &= \int p(f_k^*|x^*, \boldsymbol{f}_k) p(\boldsymbol{f}_k|\boldsymbol{X}_k, \boldsymbol{y}_k) d\boldsymbol{f}_k \\ &= \int p(f_k^*|x^*, \boldsymbol{f}_k) \frac{p(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{f}_k)\overbrace{p(\boldsymbol{f}_k)}^{\text{prior}}}{p(\boldsymbol{y}_k|\boldsymbol{X}_k)} d\boldsymbol{f}_k \\ &= \mathcal{N}(\mu_{k,pred}(x^*), \sigma^2_{k,pred}(x^*)), \end{aligned}$$

where $\boldsymbol{f}_k$ is defined in Eq. (4.2), the predictive mean $\mu_{k,pred}(x^*)$ and the predictive variance $\sigma^2_{k,pred}(x^*)$ are defined in Eq. (4.1). From this posterior predictive equation, one can see that the predicted trajectory (and variance) of $\mathcal{GP}$ in device k is affected by both prior distribution and training data $(\boldsymbol{X}_k, \boldsymbol{y}_k)$ explicitly. For a specific device, the local data themselves embody the personalization role. Therefore, `FGPR` can automatically tailor a shared global model to a personalized model for each local device. This idea is similar to meta-learning, where one tries to learn a global model that can quickly adapt to a new task.

To see this, we create a simple and stylized numerical example. Another example can be found in the Appendix. Suppose there are two local devices. Device 1 has data that follows $y = \sin(x)$ while device 2 has data that follows $y = -\sin(x)$. Each device has 100 training points uniformly spread in $[0, 10]$. We use `FedAvg` to train a 2-layer neural network. Unfortunately, a single global model of a neural network simply returns a line, as shown in Figure 4.1. Mathematically, this example solves

$$\min_{\boldsymbol{\theta}} \left(\|f_{\boldsymbol{\theta}} - \sin(x)\|_2^2 + \|f_{\boldsymbol{\theta}} + \sin(x)\|_2^2 \right),$$

where $f_{\boldsymbol{\theta}}$ is a global neural network parametrized by $\boldsymbol{\theta}$ and $\|\cdot\|_2^2$ is a functional on $[0, 10]$ defined as $\|f\|_2^2 = \int_0^{10} f(x)^2 dx$.

By taking the derivative of the above objective and setting it to zero, we can find that the solution is $f_{\boldsymbol{\theta}} = 0$. This implies that the global model cannot provide meaningful predictions on both devices.

To remedy this issue, one needs to implement an additional personalization step that fine-tunes the global model from local data. This comes with its own challenges, such as starting with a bad global model (as is the case above) and introducing extra computational costs and parameters. On the other hand, a single $\mathcal{GP}$ model learned from FGPR can provide good interpolation performance for both devices. This demonstrates the advantage of automatic personalization intrinsic to FGPR.

Remark 19. *Despite* FGPR *being a global modeling approach, in our empirical section, we will compare with personalized FL using NNs when the data distributions are heterogeneous.*

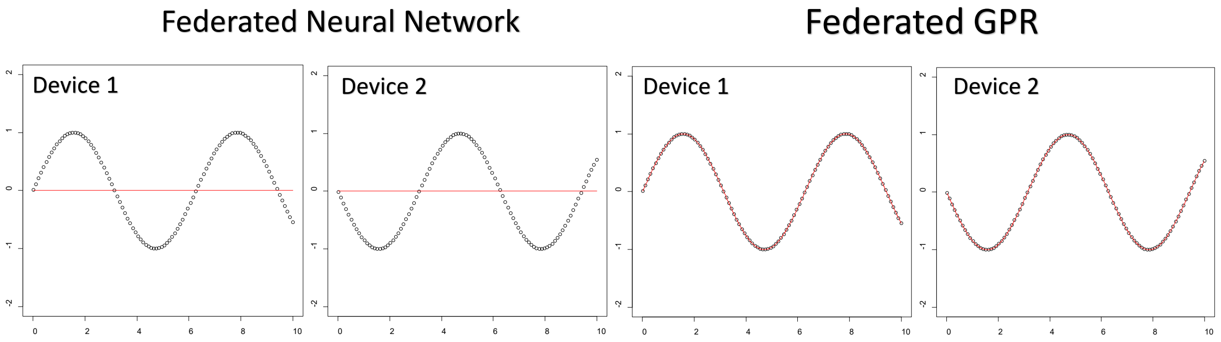

Figure 4.1: A simple example that is used to demonstrate the automatic personalization feature of FGPR. In the plot, the black dots are original data, and the red lines are fitted curves.

4.4 Theoretical Results

Proving convergence of FGPR introduces new challenges due to correlation and the decentralized nature of model estimation.

In $\mathcal{GP}$s, the objective function cannot be approximated by a summation form since all data points are correlated. This correlation renders the stochastic gradient a biased estimator of the full gradient. To the best of our knowledge, only a recent work from (Chen et al. 2020) has shown theoretical convergence results of centralized $\mathcal{GP}$ in a correlated setting. Adding to that, FGPR aggregates parameters that are estimated on only a partial dataset.

In this section, we take a step forward in understanding the theoretical properties of $\mathcal{GP}$ estimated in a federated fashion. Specifically, we provide several probabilistic convergence results of FGPR under both homogeneous and heterogeneous clients and under both full and partial device participation settings.

To proceed, we define $\boldsymbol{\theta}_{\mathcal{K}} = (\theta_1, l)$ such that $\boldsymbol{\theta} = (\theta_1, \theta_2, l)$. Here, θ_1 is the signal variance parameter, $\theta_2 = \sigma$ is the noise parameter, and l is the length parameter. Denote by $\boldsymbol{\theta}^* := (\theta_1^*, \theta_2^*, l^*)$ the true data-generating parameter. We impose a structure on the kernel function such that $\mathcal{K}(\cdot, \cdot; \boldsymbol{\theta}_{\mathcal{K}}) = \theta_1^2 \mathrm{k}_f(\cdot, \cdot)$ where $\mathrm{k}_f(\cdot, \cdot)$ is a known function. Now, we define $\mathcal{C}(x_1, x_2) = \mathcal{K}(x_1, x_2; \boldsymbol{\theta}_{\mathcal{K}}) + \sigma_2^2 \mathbb{I}_{x_1 = x_2}$ as a covariance function, where $\mathbb{I}$ is an indicator function. This form of covariance function is ubiquitous and widely adopted. For instance, the Matérn covariance is in the form of

$$\begin{aligned}
&\mathcal{C}_v(x_1, x_2) \\
&= \theta_1^2 \frac{2^{1-v}}{\Gamma(v)} \left(\sqrt{2v} \frac{\|x_1 - x_2\|}{l} \right)^v K_v \left(\sqrt{2v} \frac{\|x_1 - x_2\|}{l} \right) \\
&\quad + \theta_2^2 \mathbb{I}_{x_1 = x_2}
\end{aligned}$$

where v is a positive scalar and K_v is the modified Bessel function of the second kind. In this example, $\mathrm{k}_f(x_1, x_2) = \frac{2^{1-v}}{\Gamma(v)} \left(\sqrt{2v} \frac{\|x_1 - x_2\|}{l} \right)^v K_v \left(\sqrt{2v} \frac{\|x_1 - x_2\|}{l} \right)$. Another example is the RBF covariance:

$$\mathcal{C}_{RBF}(x_1, x_2) = \theta_1^2 \mathsf{exp} \left(\frac{\|x_1 - x_2\|^2}{2l^2} \right) + \theta_2^2 \mathbb{I}_{x_1 = x_2}.$$

There are also many other examples, such as the Ornstein–Uhlenbeck covariance and the periodic covariance (Williams and Rasmussen 2006).

Remark 20. *A more general setting is to consider the compound kernel function that is in the form of* $\mathcal{K}(\cdot, \cdot; \boldsymbol{\theta}_{\mathcal{K}}) = \sum_{i=1}^{A} \theta_{1i}^2 \mathrm{k}_{f_i}(\cdot, \cdot)$*. For simplicity, in the theoretical analysis, we assume* $A = 1$*. However, our proof techniques can be easily extended to the scenario where* $A > 1$*.*

In the theoretical analysis, we will show the explicit convergence bounds on θ_1 and θ_2. The convergence behavior of the length parameter l is still an open problem (Chen et al. 2020). The key reason is that one needs to apply the eigendecomposition technique to the kernel function and carefully analyze the lower and upper bounds of eigenfunctions. The length parameter l, however, lies in the denominator of a kernel function. In this case, it is extremely challenging to write the kernel function in the form of eigenvalues and bound them. To the best of our knowledge, the work that studies convergence results of l is still vacant even in centralized regimes.

4.4.1 Assumptions

To derive our convergence results, we make the following assumptions.

Assumption 20.1. *The parameter space $\mathbf{\Theta}$ is a compact and convex subset of $\mathbb{R}^2$. Moreover, $(\theta_1^*, \theta_2^*)^\intercal \in \mathbf{\Theta}^\circ$ and $\sup_{(\theta_1,\theta_2)^\intercal \in \mathbf{\Theta}} \|(\theta_1, \theta_2)^\intercal - (\theta_1^*, \theta_2^*)^\intercal\| > 0$, where $\mathbf{\Theta}^\circ$ is the interior of set $\mathbf{\Theta}$.*

This assumption indicates that all parameter iterates are bounded, and the global minimizer $(\theta_1^*, \theta_2^*)^\intercal$ exists. Without loss of generality, assume the lower (or upper) bound of the parameter space on each dimension is θ_{min} (or θ_{max}).

Assumption 20.2. *The norm of the stochastic gradient is bounded. Specifically,*

$$0 \leq \left\| g_k(\cdot; \xi_k^{(t)}) \right\| \leq G, \textit{ for all } k \in [K], t \in [T].$$

Here T is defined as the total number of iteration indices on each device. Mathematically, $T = R(E-1)$ and $[T] = \{0, \ldots, T\}$.

Remark 21. *It is very common to assume the local functions are L-smooth, (strongly-)convex, or the variance of the stochastic gradient is bounded. Here we do not make those assumptions.*

In the $\mathcal{GP}$ setting, the explicit convergence bound depends on the rate of decay of eigenvalues from a specific type of kernel function. In this paper, we study two types of kernel functions: (1) kernel functions with exponential eigendecay rates; and (2) kernel functions with polynomial eigendecay rates. Those translate to the following assumptions.

Assumption 21.1. *For each $k \in [K]$, the eigenvalues of function k_f with respect to probability measure μ are $\{\lambda_{1j}\}_{j=1}^\infty = \{C_k e^{-b_k j}\}_{j=1}^\infty$, where $b_k > 0$ and $C_k < \infty$. Without loss of generality, assume $C_k \leq 1$.*

Assumption 21.2. *For each $k \in [K]$, the eigenvalues of function k_f with respect to probability measure μ are $\{\lambda_{1j}\}_{j=1}^\infty = \{C_k j^{-2b_k}\}_{j=1}^\infty$, where $b_k > \frac{\sqrt{21}+3}{4}$ and $C_k < \infty$. Without loss of generality, assume $C_k \leq 1$.*

Remark 22. *Assumption 21.1 is satisfied by smooth kernels such as RBF kernels and Assumption 21.2 is satisfied by the non-smooth kernels such as Matérn kernels.*

4.4.2 Homogeneous Setting

We first assume that data across all devices are generated from the same underlying process or distribution (i.e., homogeneous data). Mathematically, it indicates (Li et al. 2019b)

$$\lim_{N_1,\ldots,N_k\to\infty}\left|\sum_{k=1}^{K} p_k L_k(\boldsymbol{\theta}^*; D_k) - \sum_{k=1}^{K} p_k L_k(\boldsymbol{\theta}_k^*; D_k)\right| = 0.$$

We briefly parse this expression. Since the data distribution across all devices is homogeneous, we know, for each k, $\boldsymbol{\theta}_k^* = \boldsymbol{\theta}^*$ as $N_k \to \infty$. Therefore, $\sum_{k=1}^{K} p_k L_k(\boldsymbol{\theta}^*; D_k) = \sum_{k=1}^{K} p_k L_k(\boldsymbol{\theta}_k^*; D_k)$. Later, we will consider the heterogeneous data settings, which are often more realistic in real-world applications.

To derive the convergence result, we divide $[g_k(\boldsymbol{\theta}; \xi_k)]_1$ by a constant factor $s_1(M_k) = \tau \log M_k$ and $[g_k(\boldsymbol{\theta}; \xi_k)]_2$ by $s_2(M_k) = M_k$, where $[g_k(\boldsymbol{\theta}; \xi_k)]_i$ is the i-th component in the stochastic gradient. Those scaling factors are introduced to ensure $[g_k(\boldsymbol{\theta}; \xi_k)]_1$ and $[g_k(\boldsymbol{\theta}; \xi_k)]_2$ have the same scale in the theoretical analysis.

Remark 23. *The aforementioned scaling factors are only needed for convergence results. In practice, we observe that those factors $s_1(M_k), s_2(M_k)$ have minimal influence on the model performance.*

Our first Theorem shows that `FGPR` using RBF kernels converges if all devices participated in the training.

Theorem 24. *(RBF kernels, synchronous update) Suppose Assumptions 20.1-21.1 hold. At each communication round, assume $|\mathcal{S}| = K$. If $\eta^{(t)} = \mathcal{O}(\frac{1}{t})$ (i.e., a decay learning rate scheduler), then for some constants $\beta_1, C_{\boldsymbol{\theta}}, c_{\boldsymbol{\theta}} > 0, \epsilon_k \in (0, \frac{1}{2})$, when $M_k > C_{\boldsymbol{\theta}}$, at iteration T, with probability at least $\min_k \left(1 - C_{\boldsymbol{\theta}} T \mathsf{exp}\left\{-c_{\boldsymbol{\theta}} \left(\log M_k\right)^{2\epsilon_k}\right\}\right)$,*

$$\begin{aligned}&\left|\bar{\theta}_1^{(T)} - \theta_1^*\right|^2 + \left|\bar{\theta}_2^{(T)} - \theta_2^*\right|^2 \\ &\quad \leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{T+1} \\ &\quad + \mathcal{O}\left(\max_k \frac{\log M_k}{M_k} + \sum_{k=1}^{K} p_k (\log M_k)^{\epsilon_k - \frac{1}{2}}\right),\end{aligned}$$

and with probability at least

$$\min_k \left(1 - C_{\boldsymbol{\theta}}\left(\log\left(M_k^{\epsilon_k - \frac{1}{2}}\right)\right)^4 T \mathsf{exp}\left\{-c_{\boldsymbol{\theta}} M_k^{2\epsilon_k}\right\}\right),$$

$$\left\|\bar{\theta}_2^{(T)}-\theta_2^*\right\|_2^2 \leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{T+1} + \mathcal{O}\left(\max_k \frac{\log M_k}{M_k}+\sum_{k=1}^K p_k M_k^{\epsilon_k-\frac{1}{2}}\right).$$

Here, constants $\beta_1, C_{\boldsymbol{\theta}}, c_{\boldsymbol{\theta}}$ only depend on $\theta_{min}, \theta_{max}$ and $\{b_k\}_{k=1}^K$.

Remark 25. *Recall that T is the number of iterations. Theorem 24 implies that, when the batch size is large enough, then with a high probability, the parameter iterate converges to the global optimal parameter at a rate of $\mathcal{O}(\frac{1}{T})$. This is credited to the unique structure of the $\mathcal{GP}$ objective function, which we refer to as relaxed convexity (See Lemma 4 and Lemma 5 in the Appendix).*

Remark 26. *In the upper bound, there is a term $\frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{T+1} \sim \frac{(E-1)^2}{T+1}$, where E is the number of local SGD steps. To ensure this term decreases with respect to T, one needs to ensure E does not exceed $\Omega(\sqrt{T})$. Otherwise, the `FGPR` will not converge. For instance, if $E=T$, then the `FGPR` is equivalent to the one-shot communication approach (Zhang et al. 2013).*

Remark 27. *In addition to the $\mathcal{O}(\frac{1}{T})$ term, there is also a statistical error term $\mathcal{O}(\max_k \frac{\log M_k}{M_k}+\sum_{k=1}^K p_k M_k^{\epsilon_k-\frac{1}{2}})$ that appeared in the upper bound. Theoretically, it indicates that a large batch size is capable of reducing errors in parameter estimation.*

Remark 28. *From Theorem 24, it can be seen that $\left\|\bar{\theta}_2^{(T)}-\theta_2^*\right\|_2^2$ has smaller error term than $\mathcal{O}\left(\sum_{k=1}^K p_k(\log M_k)^{\epsilon_k-\frac{1}{2}}\right)$. This implies that the noise parameter θ_2 is easier to estimate than θ_1. This is intuitively understandable due to the different eigenvalue structures dictated by k_f compared to $\mathbb{I}_{x_1=x_2}$.*

Next, we study the convergence behavior under the asynchronous update (i.e., partial device participation) framework. In this scenario, only a portion of devices is actively sending their model parameters to the central server at each communication round.

Theorem 29. *(RBF kernels, asynchronous update) Suppose Assumptions 20.1-21.1 hold. At each communication round, assume $|\mathcal{S}|=K_{sample}<K$ number of devices are sampled according to the sampling probability p_k. If $\eta^{(t)}=\mathcal{O}(\frac{1}{t})$, then for some constants $C_{\boldsymbol{\theta}}, c_{\boldsymbol{\theta}}>0, \epsilon_k \in (0,\frac{1}{2})$, when*

$M_k > C_{\boldsymbol\theta}$, at iteration T, with probability at least $\min_k \left(1 - C_{\boldsymbol\theta} T \mathsf{exp}\left(-c_{\boldsymbol\theta} \left(\log M_k\right)^{2\epsilon_k}\right)\right)$,

$$\mathbb{E}_{\mathcal{S}}\left\{\left|\bar\theta_1^{(T)} - \theta_1^*\right|^2 + \left|\bar\theta_2^{(T)} - \theta_2^*\right|^2\right\}$$
$$\leq \frac{2\beta_1^2\left(\frac{1}{|\mathcal{S}|}4E^2 + 8(E-1)^2 + 2\right)G^2}{T+1}$$
$$+\mathcal{O}\left(\max_k \frac{\log M_k}{M_k} + \sum_{k=1}^{K} p_k (\log M_k)^{\epsilon_k - \frac{1}{2}}\right),$$

and with probability at least

$$\min_k \left(1 - C_{\boldsymbol\theta}\left(\log\left(M_k^{\epsilon_k - \frac{1}{2}}\right)\right)^4 T\mathsf{exp}\left\{-c_{\boldsymbol\theta} M_k^{2\epsilon_k}\right\}\right),$$

$$\mathbb{E}_{\mathcal{S}}\left\{\left\|\bar\theta_2^{(T)} - \theta_2^*\right\|_2^2\right\}$$
$$\leq \frac{2\beta_1^2\left(\frac{1}{|\mathcal{S}|}4E^2 + 8(E-1)^2 + 2\right)G^2}{T+1}$$
$$+\mathcal{O}\left(\max_k \frac{\log M_k}{M_k} + \sum_{k=1}^{K} p_k M_k^{\epsilon_k - \frac{1}{2}}\right),$$

where the expectation is taken over the set $\mathcal{S}$, and please refer to Appendix 6.3 for a rigorous definition.

Remark 30. *Under the asynchronous update setting, a similar convergence guarantee holds. The only difference is that the number of active devices $|\mathcal{S}|$ plays a role in the upper bound. Numerically, the ratio $\frac{E^2}{|\mathcal{S}|}$ enlarges the upper bound and impedes the convergence rate. As $|\mathcal{S}|$ grows (i.e., more devices participate in the training), the ratio $\frac{E^2}{|\mathcal{S}|}$ decreases.*

Our next theorem provides explicit convergences rate for `FGPR` with Matérn kernels under both a synchronous and asynchronous update scheme.

Theorem 31. *(Matérn kernels) Suppose Assumptions 20.1-20.2 and 21.2 hold,*

(1) At each communication round, assume $|\mathcal{S}| = K$. If $\eta^{(t)} = \mathcal{O}(\frac{1}{t})$, then for some constants $C_{\boldsymbol\theta}, c_{\boldsymbol\theta} > 0$, $\beta_1 > 0, b_k > \frac{(\sqrt{21}+3)}{4}$ and $0 < \alpha_k < \frac{1}{2}$, when $M_k > C_{\boldsymbol\theta}$, with probability at least

$\min_k \left(1 - C_{\boldsymbol{\theta}} T (\log(M_k^{\epsilon_k - \frac{1}{2}}))^4 \mathsf{exp}\{-c_{\boldsymbol{\theta}} M_k^{2\epsilon_k}\}\right)$,

$$\left\|\bar{\theta}_2^{(T)} - \theta_2^*\right\|_2^2 \le \frac{2\beta_1^2 \left(8(E-1)^2 + 2\right) G^2}{T+1} + \mathcal{O}\left(\max_k M_k^{-\frac{8b_k^2 - 12b_k - 6 - 3\alpha_k - 4\alpha_k b_k}{8b_k^2 - 4b_k}}\right) + \mathcal{O}\left(\sum_{k=1}^K p_k M_k^{\epsilon_k - \frac{1}{2}}\right).$$

Additionally,

$$\left\|\nabla L(\bar{\boldsymbol{\theta}}^{(T)})\right\|_2^2 \le \frac{2\beta_1^2 \left(8(E-1)^2 + 2\right) G^2}{4\theta_{min}^4 (T+1)} + \mathcal{O}\left(\max_k \left\{ M_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)} - 1} + \sum_{k=1}^K p_k M_k^{\epsilon_k - \frac{1}{2}} \right\}\right).$$

(2) At each communication round, assume $|\mathcal{S}| = K_{sample}$, number of devices are sampled according to the sampling probability p_k. If $\eta^{(t)} = \mathcal{O}(\frac{1}{t})$, then for some constants $C_{\boldsymbol{\theta}}, c_{\boldsymbol{\theta}} > 0$, $\beta_1 > 0, b_k > \frac{(\sqrt{21}+3)}{4}$ and $0 < \alpha_k < \frac{1}{2}$, when $M_k > C_{\boldsymbol{\theta}}$, with probability at least

$$\min_k \left(1 - C_{\boldsymbol{\theta}} T \left(\log\left(M_k^{\epsilon_k - \frac{1}{2}}\right)\right)^4 \mathsf{exp}\{-c_{\boldsymbol{\theta}} M_k^{2\epsilon_k}\}\right),$$

$$\mathbb{E}_{\mathcal{S}}\left\{\left\|\bar{\theta}_2^{(T)} - \theta_2^*\right\|_2^2\right\} \le \frac{2\beta_1^2 \left(\frac{4E^2}{|\mathcal{S}|} + 8(E-1)^2 + 2\right) G^2}{T+1} + \mathcal{O}\left(\max_k M_k^{-\frac{8b_k^2 - 12b_k - 6 - 3\alpha_k - 4\alpha_k b_k}{8b_k^2 - 4b_k}}\right) + \mathcal{O}\left(\sum_{k=1}^K p_k M_k^{\epsilon_k - \frac{1}{2}}\right).$$

Additionally,

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S}}\left\{\left\|\nabla L(\bar{\boldsymbol{\theta}}^{(T)})\right\|_2^2\right\} \\
&\leq \frac{2\beta_1^2\left(\frac{4E^2}{|\mathcal{S}|}+8(E-1)^2+2\right)G^2}{4\theta_{min}^4(T+1)} \\
&\quad + \mathcal{O}\left(\max_k\left\{M_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1}+\sum_{k=1}^{K}p_kM_k^{\epsilon_k-\frac{1}{2}}\right\}\right).
\end{aligned}$$

Remark 32. *It can be seen that the* `FGPR` *using Matérn kernel has a larger statistical error than the one using RBF kernel. In the RBF kernel, the statistical error is partially affected by* $\mathcal{O}\left(\max_k \frac{logM_k}{M_k}\right)$ *(Theorems 24,29) while this term becomes* $\mathcal{O}\left(\max_k M_k^{-\frac{8b_k^2-12b_k-6-3\alpha_k-4\alpha_kb_k}{8b_k^2-4b_k}}\right)$ *in the Matérn kernel. The latter one is larger since* $b_k > \frac{(\sqrt{21}+3)}{4}$ *and* $\alpha_k \in (0, 0.5)$*. This difference arises from the fact that the Matérn kernel has a slower eigenvalue decay rate (determined by* b_k*) than the RBF kernel (i.e., polynomial vs. exponential). This slow decay rate leads to slower convergence and larger statistical error. When* b_k *becomes larger, the decay rate becomes faster, and the influence of* $\mathcal{O}\left(\max_k M_k^{-\frac{8b_k^2-12b_k-6-3\alpha_k-4\alpha_kb_k}{8b_k^2-4b_k}}\right)$ *gets smaller. In this case, the statistical error is dominated by* $\mathcal{O}\left(\sum_{k=1}^K p_kM_k^{\epsilon_k-\frac{1}{2}}\right)$*, which is the same as the one in the RBF kernel.*

Remark 33. *In addition to the convergence bound on parameter iterates, we also provide an upper bound on the full gradient* $\left\|\nabla L(\bar{\boldsymbol{\theta}}^{(T)})\right\|_2^2$*. This bound scales the same as the bound for* $\left\|\bar{\theta}_2^{(T+1)}-\theta_2^*\right\|_2^2$*.*

Remark 34. *For Matérn kernel, there is no explicit convergence guarantee for parameter* $\bar{\theta}_1$*. The reason is that it is very hard to derive the lower and upper bounds for the SG for Matérn kernel. However, Theorem 31 shows that both* $\bar{\theta}_2$ *and the full gradient converge at rates of* $\mathcal{O}(\frac{1}{T})$ *subject to statistical errors.*

4.4.3 Heterogeneous Setting

Besides the homogeneous setting, we further consider the scenario where data from all devices are generated from several different processes or distributions. Equivalently, this indicates

$$\mathbb{P}\left(\left|\sum_{k=1}^{K}p_kL_k(\boldsymbol{\theta}^*;D_k)-\sum_{k=1}^{K}p_kL_k(\boldsymbol{\theta}_k^*;D_k)\right|=0\right)=0.$$

Since the data are heterogeneous, we know $\boldsymbol{\theta}_k^* \neq \boldsymbol{\theta}^*$. As a result, the weighted average of $L_k(\boldsymbol{\theta}_k^*; D_k)$ can be very different from $L(\boldsymbol{\theta}^*)$. We here note that convergence results for the heterogeneous setting are moved to Appendix due to space limitations.

Overall, in this theoretical section, we show that the `FGPR` is guaranteed to converge under both homogeneous setting (Sec. 4.2) and heterogeneous setting, regardless of the synchronous updating or the asynchronous updating.

4.5 Proof of Concept

We start by validating the theoretical results obtained in Sec. 4.4.2. We also provide sample experiments that shed light on key properties of `FGPR`.

Example 1: Homogeneous Setting with Balanced Data. We generate data from a $\mathcal{GP}$ with zero-mean and both a RBF and Matérn$-3/2$ kernel. We consider $\theta_1 \in [0.1, 10]$, $\theta_2 \in [0.01, 1]$ and a length parameter $\boldsymbol{l} \in [0.01, 1]^d$. The input space is a d-dimensional unit cube $[0, 1]^d$ in $\mathbb{R}^d$ with $d \in \{1, \dots, 10\}$ and the dimension of the output is one. We conduct 20 independent experiments. In each experiment, we first randomly sample $\theta_1, \theta_2, \boldsymbol{l}$ and d to generate data samples from the $\mathcal{GP}$. In each scenario, we set $N_k = \frac{N}{K}$. This setting is homogeneous and balanced as the number of data points across K clients is equal and they all come from the same underlying stochastic process. We consider three scenarios: (1) $K = 20, N = 5000$, (2) $K = 50, N = 2000$, (3) $K = 100, N = 800$. Results from the RBF kernel are provided in Figure 4.2. Due to space limitation, we move plots of the Matérn Kernel into Appendix 1. It can be seen that the convergence rate follows a $\mathcal{O}(\frac{1}{T})$ pattern. In some runs, the values of $\left\|\bar{\theta} - \theta^*\right\|_2^2$ are very large at the beginning. Those imply that initial parameters are far away from true parameters. However, after 20-40 communication rounds, those values quickly diminish. In 4.2, we also observe that plots in (c) are more dispersed and fluctuated than (a) and (b). This is because each device only has fewer data points ($N/K = 2000/100 = 20$).

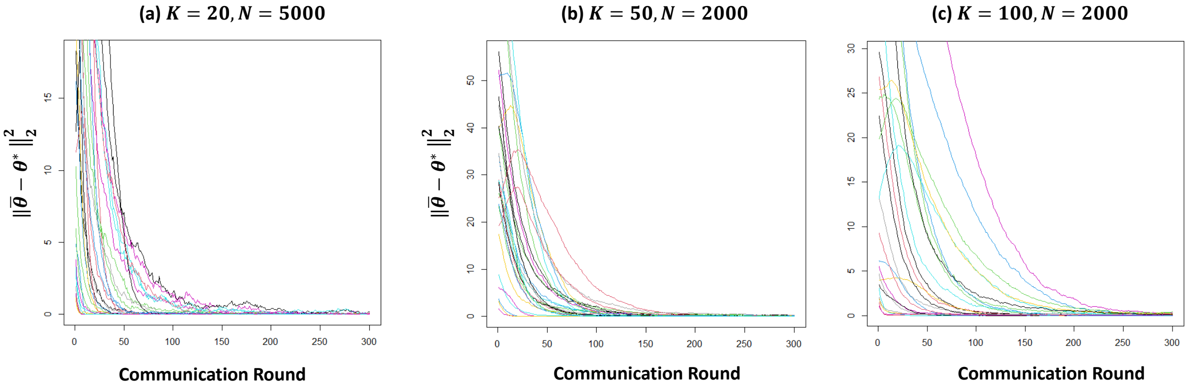

Figure 4.2: (RBF kernel) Evolution of $\left\|\bar{\theta} - \theta^*\right\|_2^2$ over training epochs. In the plot, each color represents an independent run. The input dimension d is different for each run and $d \in \{1, \dots, 10\}$.

Example 2: Homogeneous Setting with Unbalanced Data. We use the same data-generating strategy as Example 1, but the sample sizes are unbalanced. Specifically, the number of data points in each device ranges from 10 to 10,000. The histogram of data distribution from one experiment is given in Figure 4.4. The convergence curves are plotted in Figure 4.3. Again, the convergence rate agrees with our theoretical finding. This simple example reveals a critical property of `FGPR`: `FGPR` can help devices with few observations recover true parameters (subject to statistical errors) or reduce prediction errors. We will further demonstrate this advantage in the heterogeneous setting in Sec. 4.6.

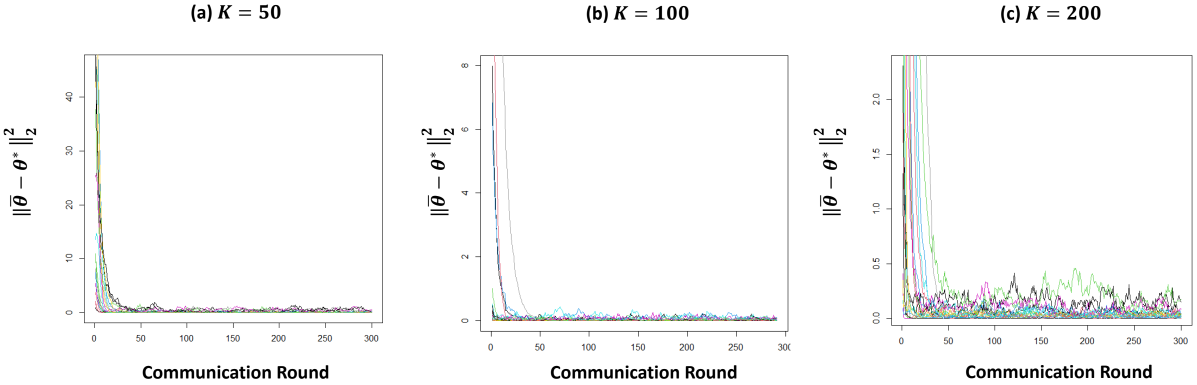

Figure 4.3: (RBF kernel) Evolution of $\left\|\bar{\theta} - \theta^*\right\|_2^2$ over training epochs using unbalanced data. In the plot, each color represents an independent run. The input dimension d is different for each run and $d \in \{1, \ldots, 10\}$.

Example 3: The Ability to Recover Accurate Predictions for a Badly Initialized $\mathcal{GP}$. When training an FL algorithm, it is not uncommon to initialize the model parameters $\boldsymbol{\theta}$ near a bad stationary point. Here we provide one toy example. We simulate data from $y = sin(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.2)$ and create two clients ($K = 2$). Each client has 100 training data points and $1,000$ testing data points that are uniformly sampled from $[0, 1]$. We artificially find a bad initial parameter $\boldsymbol{\theta}$ such that the fitted curve is just a flat line. This can be achieved by finding a $\boldsymbol{\theta}$ whose noise parameter θ_2 is large. In this case, $\boldsymbol{\theta} = (1, 10, 1)$ where the $\mathcal{GP}$ interprets all data as noise and simply returns a flat line.

We evaluate the predictive performance of `FGPR` using the averaged root-mean-square error (RMSE) metric. The RMSE for each device is evaluated on the local testing data, and the averaged RMSE averages RMSEs across all devices. We find that `FGPR` is robust to parameter initialization. We plot the evolution of averaged RMSE versus training epoch in Figure 4.5. It can be seen that, even when the parameter is poorly initialized, `FGPR` can still correct the wrong initialization after several communication rounds. This credits to the stochasticity in the SGD method. It is known that, in ERM, SGD can escape bad stationary solutions and converge to solutions with good

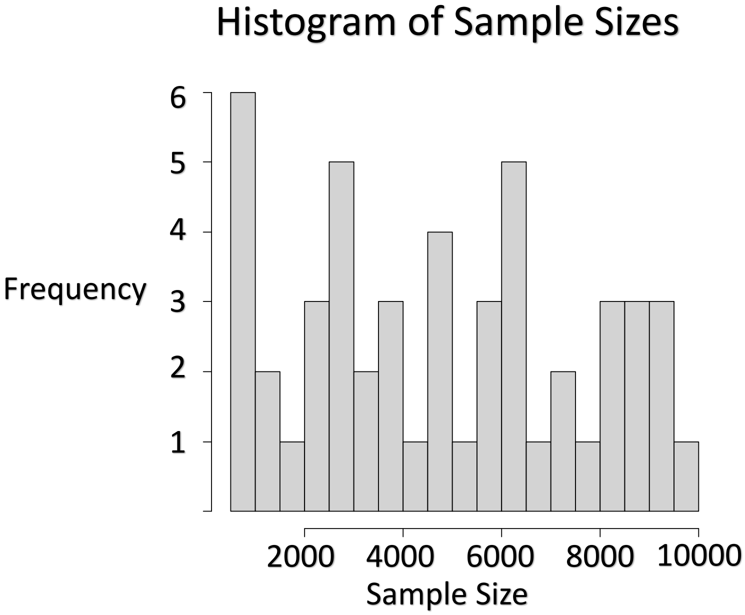

Figure 4.4: Histogram of Sample Sizes (Example 2).

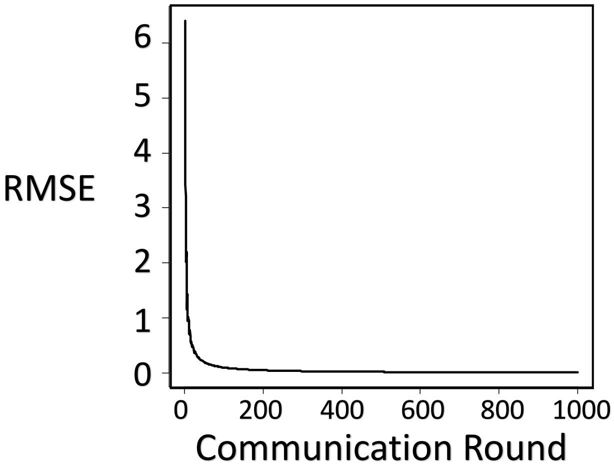

Figure 4.5: Evolution of the averaged RMSE in Example 3.

generalization (often flat ones) (Wu et al. 2018).

4.6 Application I: Multi-fidelity Modeling

For many computer experiments, high-fidelity numerical simulations of complex physical processes typically require a significant amount of time and budget. This limits the number of data points researchers can collect and affects the modeling accuracy due to insufficient data. A major work trend has been proposed to augment the expensive data source with cheaper surrogates to overcome this hindrance. Multi-fidelity models are designed to fuse scant but accurate observations (i.e., high-fidelity, HF) with cheap and biased approximations (i.e., low-fidelity, LF) to improve the HF model performance.

Denote by f_h a high-fidelity function and f_l a low-fidelity function. Multi-fidelity approaches (Bailly and Bailly 2019, Cutajar et al. 2019, Brevault et al. 2020) aim to use f_l to better predict f_h. During the past decades, many multi-fidelity models have been proposed to fulfill this goal. We refer to (Fernández-Godino et al. 2016) and (Peherstorfer et al. 2018) for detailed literature reviews. Among all the methods, $\mathcal{GP}$-based approaches have caught the most attention due to their ability to incorporate prior beliefs, interpolate complex functional patterns and quantify uncertainties (Fernández-Godino et al. 2016). The last ability is critical to fuse observations across different fidelities effectively.

Within many applications, two specific models have been shown to be very competitive (Brevault et al. 2020); the auto-regressive (`AR`) and the Deep $\mathcal{GP}$ (`Deep`) approaches. Both approaches model f_h as shown below

$$f_h(x) = \rho(f_l(x), x) + \Delta(x),$$

where $\rho(\cdot, \cdot)$ is a space-dependent non-linear transformation and $\Delta(x)$ is a bias term modeled through a $\mathcal{GP}$.

More specifically, the `AR` model (Kennedy and O'Hagan 2000) sets the transformation as a linear mapping such that $\rho(f_l(x), x) = \rho_c f_l(x)$, where ρ_c is a constant. It then imposes a $\mathcal{GP}$ prior on f_l and accordingly obtains its posterior f_l^*. As a result, one can derive the closed-form posterior distribution $p(f_h|f_l^*, x, y)$ and obtain the posterior predictive equation of the high-fidelity model. On the other hand, the `Deep` model (Cutajar et al. 2019) treats $\rho(f_l(x), x)$ as a deep Gaussian process to uncover highly complex relationships among f_l and f_h . `Deep` is one of the state-of-the-art multi-fidelity models. For more details, please refer to (Brevault et al. 2020).

Nowadays, as data privacy gains increased importance, having access to data across multiple fidelities is often impractical as multiple clients can own data. This imposes a key challenge in multi-fidelity modeling approaches as effective inference on expensive high-fidelity models often necessitates the need to borrow strength from other information sources. Fortunately, in such a case, FGPR is a potential candidate that learns a $\mathcal{GP}$ prior without sharing data.

In this section, we test the viability of FGPR in multi-fidelity modeling. We test our approach using settings where local devices contain data with different fidelities. We then use Algorithm 4.1 to train our FGPR algorithm. Specifically, each device runs several steps of SGD and then sends its model parameter to the central orchestrator. The orchestrator then aggregates model parameters and sends the aggregated parameter back to each device. This procedure is repeated several times till some exit condition is met. Upon estimating the model parameters, we then test the local predictive accuracy using the predictive equation (4.1) for device k.

We benchmark FGPR with several state-of-the-art models. Interestingly, our results (Table 4.2) show that FGPR not only preserves privacy but also can provide superior performance than centralized multi-fidelity approaches.

Below we detail the benchmark models: (1) Separate which fits a single $\mathcal{GP}$ to the HF dataset without any communication. This means the HF dataset does not use any information from the LF dataset; (2) the AR method (Kennedy and O'Hagan 2000). AR is the most classical and widely-used multi-fidelity modeling approach (Laurenceau and Sagaut 2008, Fernández-Godino et al. 2016, Bailly and Bailly 2019); (3) the Deep model (Damianou and Lawrence 2013) highlighted above; (4) Modular $\mathcal{GP}$ (Moreno-Muñoz et al. 2021) that models each fidelity-level as an output. For this method, we introduce 20 inducing points for each device and 3 global latent variables. All output values are standardized to mean 0 and variance 1.

We start with two simple illustrative examples from (Cutajar et al. 2019) and then benchmark all models on five well-known models in the multi-fidelity literature.

Example 1: Linear Example - We first present a simple one dimensional linear example where $x \in [0, 1]$. The low and high-fidelity models are given by (Cutajar et al. 2019)

$$\begin{aligned} y_l(x) &= \frac{1}{2} y_h(x) + 10(x - \frac{1}{2}) + 5, \\ y_h(x) &= (6x-2)^2 \sin(12x-4), \end{aligned}$$

where $y_l(\cdot)$ is the output from the LF model and $y_h(\cdot)$ is the output from the HF model. We simulate 100 data points from the LF model and 20 data points from the HF model. The number of testing data points is 1,000.

Example 2: Nonlinear Example - The one dimensional non-linear example for $x \in [0, 2]$ is given as

$$\begin{aligned} y_l(x) &= \cos(15x), \\ y_h(x) &= x\mathsf{exp}^{y_l(2x-0.2)} - 1. \end{aligned}$$

We use the same data-generating strategy in Example 1.

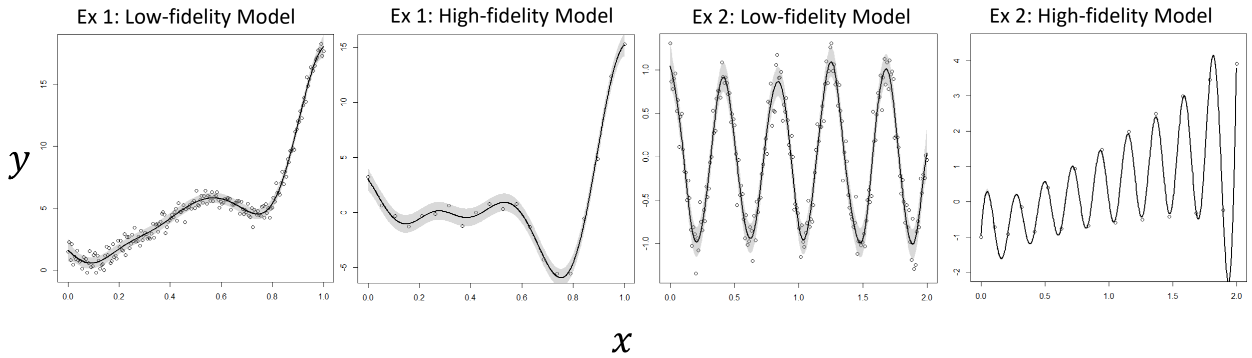

Figure 4.6: Results of Example 1 and 2. The solid black line denotes the predicted mean, and the grey area is a 95% confidence interval.

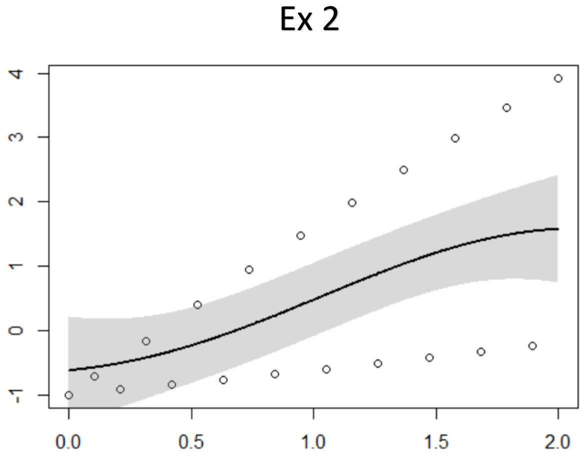

Figure 4.7: Results of Example 2 using `separate` on the HF data only.

The results from both examples are plotted in Figure 4.6. The results provide a simple proof-of-concept that the learned `FGPR` is able to accurately predict the HF model despite sparse observations. Additionally, `FGPR` can also adequately capture uncertainties (grey areas in Figure 4.6) in predictions. The results also confirm our insights on automatic personalization in Sec. 4.3.3 whereby a single global model was able to adequately fit both HF and LF datasets. Here we conduct one additional comparison study on Example 2. We train a $\mathcal{GP}$ model solely using a high-fidelity dataset. The fitted curve is plotted in Figure 4.7. It can be seen that, without borrowing any information from the LF dataset, the fitted $\mathcal{GP}$ curve fails to recover the true underlying pattern. This example further demonstrates the advantage of `FGPR`: the shared global model parameter encodes key information (e.g., trend, pattern) from the low-fidelity dataset such that the high-fidelity dataset can exploit this information to fit a more accurate surrogate model.

Table 4.2: RMSEs and standard deviations compared to the true HF model. Each experiment is repeated 30 times. The sample size is in a format of HF/MF/LF, where MF represents a medium-fidelity model.

RMSE-HF	Sample Size	**FGPR**	Separate	AR	Deep	Modular $\mathcal{GP}$
CURRIN	$40/0/200$	$\mathbf{0.148 \pm 0.056}$	0.301 ± 0.080	0.295 ± 0.052	0.252 ± 0.064	0.243 ± 0.033
PARK	$50/0/300$	$\mathbf{0.012 \pm 0.002}$	0.052 ± 0.006	0.035 ± 0.001	0.013 ± 0.001	0.039 ± 0.001
BRANIN	$20/40/200$	0.260 ± 0.065	0.374 ± 0.089	0.335 ± 0.070	$\mathbf{0.213 \pm 0.085}$	0.365 ± 0.076
Hartmann-3D	$50/100/200$	$\mathbf{0.365 \pm 0.074}$	0.456 ± 0.087	0.412 ± 0.067	0.383 ± 0.092	0.438 ± 0.085
Borehole	$50/0/200$	$\mathbf{0.604 \pm 0.006}$	0.633 ± 0.006	0.615 ± 0.005	0.622 ± 0.007	0.621 ± 0.004

Next, we consider a range of benchmark problems that are widely used in the multi-fidelity literature (Cutajar et al. 2019, Brevault et al. 2020). We defer the full specifications of those problems to the Appendix. For each experiment, we generate 1,000 testing points uniformly on the input domain.

- **CURRIN:** CURRIN (Currin et al. 1991, Xiong et al. 2013) is a two-dimensional function that is widely used for multi-fidelity computer simulation models.

- **PARK:** The PARK function (Cox et al. 2001, Xiong et al. 2013) lies in a four-dimensional space ($\boldsymbol{x} \in (0, 1]^4$). This function is often in testing for parameter calibration and design of experiments.

- **BRANIN:** BRANIN is widely used as a test function for metamodeling in computer experiments. In this example, there are three fidelity levels (Perdikaris et al. 2017, Cutajar et al. 2019).

- **Hartmann-3D:** Similar to BRANIN, this is a 3-level multi-fidelity dataset where the input space is $[0, 1]^3$.

- **Borehole Model:** The Borehole model is an 8-dimensional physical model that simulates water flow through a borehole (Moon et al. 2012, Gramacy and Lian 2012, Xiong et al. 2013).

Each experiment is repeated 30 times, and we report RMSEs of the model performance on the true HF model, along with the standard deviations in Table 4.2. The training data size is highlighted in the table.

First, it can be seen in Table 4.2, FGPR consistently yields smaller RMSE than Separate. This confirms that FGPR is able to borrow strength across multi-fidelity datasets. More importantly, we find that FGPR can even achieve superior performance compared to the AR and Deep benchmarks. This implies that one can avoid centralized approaches without compromising accuracy. Finally, the inferior performance of Modular $\mathcal{GP}$s is because: (1) Modular $\mathcal{GP}$ optimizes an approximate likelihood instead of the exact likelihood. FGPR, on the other hand, directly performs stochastic optimization on the exact likelihood; (2) Modular $\mathcal{GP}$ is a one-shot approach. For instance, the convergence bound of FedAvg follows $\mathcal{O}(E^2/T)$ where E is the number of local steps and $T = R(E-1)$ where R is the number of communication rounds. Clearly, E should be small (less than the order of $\mathcal{O}(R)$) to guarantee convergence. A similar result is shown in FGPR in Sec. 4.4. Whereas our model FGPR is a collaborative process where the global model is updated over R communication rounds; (3) Modular $\mathcal{GP}$ require one additional layer of approximation that sacrifices accuracy (Moreno-Muñoz et al. 2021). As a side note, as mentioned in Table 4.1, in Modular $\mathcal{GP}$, a sparse representation of the local data is shared, which entails the pseudo-targets, variational density, model parameters, and approximate likelihood value. Clearly, if the sparse approximation is close to the true local posterior, there is an infringement on local privacy. FGPR, on the other hand, only shares model parameters.

In summary, the results show that FGPR can serve as a compelling candidate for privacy-preserving multi-fidelity modeling in the modern era of statistics and machine learning.

Below, we also detail an interesting technical observation.

Remark 35. *In our settings, the weight coefficient p_k for the HF is low compared to LF, as HF clients have fewer data. For instance, in the CURRIN example, the HF coefficient is $p_1 = \frac{40}{240} = 0.17$. Therefore, the global parameter is averaged with higher weights for the LF model. Yet, the model excels in predicting the HF model. This again goes back to the fact that, unlike deep learning based FL approaches, FGPR is learning a joint prior on the functional space. The scarce HF data alone cannot learn a strong prior, yet, with the help of the LF data, such prior can be learned effectively. That being said, it may be interesting to investigate the adaptive assignment of p_k, yet this requires additional theoretical analysis.*

On par with Remark 17, we conduct an ablation study on p_i using the CURRIN function. Specifically, we use the same sample size (i.e., $N_1 = 40$, $N_2 = 200$), but we gradually increase p_1 from 0.17 to 1 and decrease p_2 from 0.83 to 0. We plot the RMSE versus p_1 in Figure 4.8. It can be seen that the RMSE remains consistent when we moderately increase p_1. However, once p_1 passes a threshold, the RMSE increases sharply. Again this is because the increased weight to HF can be misleading due to the scarcity of HF data.

4.7 Application II: Robotics

We now test the performance of FGPR on a robotic dataset [Link].

To enable accurate robot movement, one needs to control the joint torques (Nguyen-Tuong et al. 2008). Joint torques can be computed by many existing inverse dynamics models. However, in real-world applications, the underlying physical process is highly complex and often hard to derive using first principles. Data-driven models were proposed as an appealing alternative to

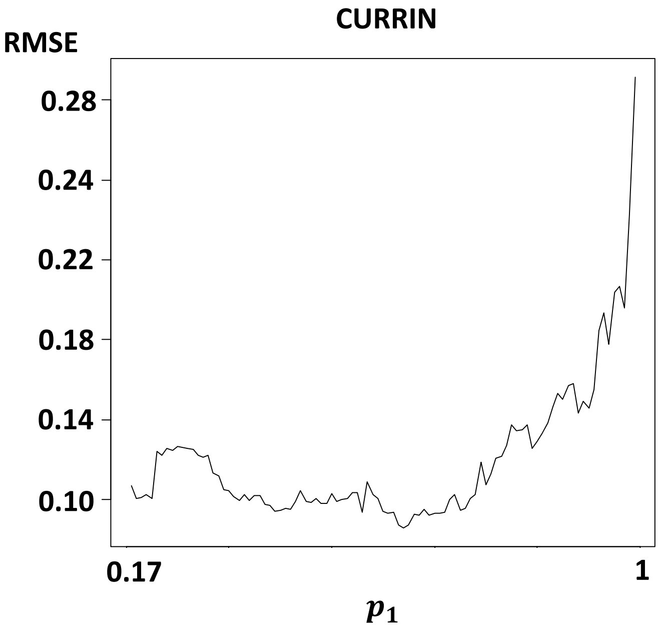

Figure 4.8: Ablation Study (CURRIN).

handle complex functional patterns and, more importantly, quantify uncertainties (Nguyen-Tuong and Peters 2011). The goal of this section is to test FGPR as a data-driven approach to accurately compute joint torques at different joint positions, velocities, and accelerations.

Table 4.3: For FGPR & Neural we report averaged RMSE and the standard deviation (std) of RMSEs across all testing devices for the robotics data. Each experiment is repeated 30 times. The standard deviation of each performance measure is reported in brackets. For DVI/DGP, we report the RMSE on a central server

Averaged RMSE ×10 std of RMSE ×10	Output 1	Output 3	Output 5	Output 7
FGPR	**2.75 (0.00)** **1.84 (0.01)**	**2.42 (0.03)** **1.57 (0.01)**	**2.20 (0.05)** **1.29 (0.02)**	**2.38 (0.01)** **1.44 (0.02)**
Neural	3.01 (0.01) 1.70 (0.00)	3.05 (0.06) 2.11 (0.02)	2.89 (0.09) 1.37 (0.02)	2.90 (0.02) 1.50 (0.01)
DVI	2.85 (0.02)	3.32 (0.06)	2.57 (0.03)	2.98 (0.02)
DGP	2.99 (0.03)	3.17 (0.04)	2.62 (0.01)	2.77 (0.02)

To this end, we test FGPR using a Matérn-$3/2$ kernel on learning an inverse dynamics problem for a seven degrees-of-freedom SARCOS anthropomorphic robot arm (Williams and Rasmussen 2006, Bui et al. 2018). This task contains $d = 21$ dimensional input and 7 dimensional output with 44,484 points for training and 4,449 points for testing. Since FGPR is a single-output FL framework, we only use one output each time (See Table 4.3). Our goal is to accurately predict the forces used at different joints given the joints' input information. We randomly partition the data into 25 devices. Overall, each device has around 1850 training points and 180 testing points each.

We benchmark FGPR with (1) neural network; (2) DGP (Deisenroth and Ng 2015) that uses the product-of-experts approximation and distributes learning tasks to different experts (i.e., nodes); (3) DVI (Gal et al. 2014) that performs distributed variational inference.

We found that neural network trained from a simple FedAvg failed. This is due to the large heterogeneity. To resolve this issue, we train Neural using a state-of-the-art personalized FL framework Ditto (Li et al. 2021). In Ditto, each local device solves two optimization problems. The first is the same as FedAvg and to find $\boldsymbol{\theta}$, while the second derives personalized parameters $\boldsymbol{v}_k$ for each client k by solving

$$\min_{\boldsymbol{v}_k} h_k(\boldsymbol{v}_k; \boldsymbol{\theta}) := \hat{R}_k(\boldsymbol{v}_k; D_k) + \frac{\lambda}{2} \|\boldsymbol{v}_k - \boldsymbol{\theta}\|_2^2$$

where λ is a regularization parameter and $\boldsymbol{\theta}$ is the shared global parameter. The idea behind Ditto is clear: in addition to updating a shared global parameter $\boldsymbol{\theta}$, each device also maintains its own personalized solution $\boldsymbol{v}_k$. Yet, the regularization term ensures that this $\boldsymbol{v}_k$ should be close to $\boldsymbol{\theta}$ such that one can retain useful information learned from a global model.

For DVI and DGP, we use Matérn-3/2 kernels and introduce 1024 inducing points for the former method.

In Table 4.3, we present results for outputs 1, 3, 5, and 7. Here, note that the RMSEs of DVI and DGP are evaluated on the central location using all testing data rather than on each node. This is because the goal of DVI or DGP is to distribute learning tasks and speed up training rather than improve the model performance on each local node. Whilst for FGPR and Neural, we can additionally obtain the standard error of RMSEs across devices since predictions are performed on local devices.

Under the heterogeneous setting, FGPR still provides lower averaged RMSE than the personalized Neural, DGP, and DVI benchmark models. This credits to (1) the flexible prior regularization in the $\mathcal{GP}$ regression that can avoid potential model over-fitting; (2) the intrinsic personalization capability of FGPR; (3) FGPR does exact inference whereas DVI and DGP use approximate objectives that may often be inadequate. Recall that DGP uses the product-of-experts approximation that induces a notion of independence across local experts (devices). DVI uses VI that faces several drawbacks, per our earlier discussion in Sec. 4.3; (4) DGP is a one-shot approach that is not optimal, as discussed earlier. Here we note that DVI requires each device to send an $N_z \times N_z \times d$ dimension tensor to the server after every single optimization step. This incurs very heavy communication loads and high costs. Also, DGP shares local predicted output to a central server, and the server can re-construct the data pattern from each device. This clearly leaks the local data information.

An additional case study on NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) tools dataset (Saxena and Goebel 2008) that involves multiple engines is deferred to the appendix due to space limitation. In this case study, we also benchmarked with federated polynomial regression models.

4.8 Conclusion

In this paper, we extend the standard $\mathcal{GP}$ regression model to a federated setting, `FGPR`. We use both theory and a wide range of experiments to justify the viability of our proposed framework. We highlight the unique capability of `FGPR` to provide automatic personalization and strong transferability on untrained devices.

`FGPR` may find value in meta-learning as it provides an inherent Bayesian perspective on this topic (Yue and Kontar 2020a). Other interesting research directions include extending the current framework to a multi-output $\mathcal{GP}$ model. The challenge lies in capturing the correlation across output in a federated paradigm. Another possible direction arises from the theoretical perspective of `FGPR`. In this work, we only provide theoretical guarantees on noise/variance parameters and the gradient norm. Studying the convergence behavior of length parameters is another crucial but challenging future research direction.

CHAPTER 5

Conclusion

The contribution of the completed studies can be summarized as follows. In Chapter 2, we propose GIFAIR-FL: a framework that imposes **G**roup and **I**ndividual **FAIR**ness to **F**ederated **L**earning settings. By adding a regularization term, our algorithm penalizes the spread in the loss of client groups to drive the optimizer to fair solutions. Our framework GIFAIR-FL can accommodate both global and personalized settings. Theoretically, we show convergence in non-convex and strongly convex settings. Our convergence guarantees hold for both $i.i.d.$ and non-$i.i.d.$ data. To demonstrate the empirical performance of our algorithm, we apply our method to image classification and text prediction tasks. Compared to existing algorithms, our method shows improved fairness results while retaining superior or similar prediction accuracy. In Chapter 3, we develop an FDA treatment for one of the most fundamental statistical models: linear regression. Our treatment is built upon hierarchical modeling that allows borrowing strength across multiple groups. To this end, we propose two federated hierarchical model structures that provide a shared representation across devices to facilitate information sharing. Notably, our proposed frameworks are capable of providing uncertainty quantification, variable selection, hypothesis testing, and fast adaptation to new unseen data. We validate our methods on a range of real-life applications, including condition monitoring for aircraft engines. The results show that our FDA treatment for linear models can serve as a competing benchmark model for the future development of federated algorithms. In Chapter 4, we propose FGPR: a Federated Gaussian process ($\mathcal{GP}$) regression framework that uses an averaging strategy for model aggregation and stochastic gradient descent for local computations. Notably, the resulting global model excels in personalization as FGPR jointly learns a shared prior across all devices. The predictive posterior then is obtained by exploiting this shared prior and conditioning on local data, which encodes personalized features from a specific dataset. Theoretically, we show that FGPR converges to a critical point of the full log-marginal likelihood function, subject to statistical errors. This result offers standalone value as it brings federated learning theoretical results to correlated paradigms. Through extensive case studies, we show that FGPR excels in a wide range of applications and is a promising approach for privacy-preserving multi-fidelity data modeling.

FDA is still at its infancy phase. Many methodological questions are yet to be answered.

Further, as FDA infiltrates new domains, the domains themselves will pose the fundamental research challenges. To this end, my research plan involves:

1. **Building a framework for distributed and collaborative design.** Optimal design of process parameters is a critical yet challenging task within many domains. This challenge arises from the need for trial and error. This process is typically done through simulations that mimic the underlying process, yet for complex engineering processes, the dimensionality of process parameters are usually high. That being said, a high fidelity simulation model might take days or weeks to obtain outputs from a single setting of process parameters. Fortunately, the increased connectivity and computation power of edge devices in manufacturing sets forth a new collaborative paradigm for process parameter design: different manufacturers collaborate and borrow strength from each other to reduce the effort of trial and error. The success within this domain may help engineers develop a distributed framework that promotes collaborative design. Along this line, I plan to design a large-scale collaborative Bayesian optimization framework that allows different manufacturers to fit response-surface models, quantify uncertainties, extract knowledge across devices and make decisions on experimental design [11]. The central idea is based on FDA methods: manufacturers run local computations and collaboratively make decisions using aggregated information from a central server.

2. **Tackling statistical challenges in FDA.** I decide to work on several open questions in FDA. (I) **Federated Graph Learning:** Learning graphic networks is a problem that has received significant attention in the statistics and computer science literature. In FDA, when there is dependence amongst devices, learning a graphical model/network structure potentially improves overall performance. However, there remains the open challenge of adapting and implementing graphical algorithms that learn pairwise sufficient statistics to respect communication and differential privacy constraints. At the heart of the challenge, if our goal is to learn a network structure amongst devices, second-order statistics are required in the computation. From a privacy and communication perspective, this requires communication between all pairs of devices in order to compute these second-order sufficient statistics. Along this line, I plan to propose practical and communication-efficient graphic modeling algorithms that overcome the aforementioned challenges. (II) **Federated Uncertainty Quantification and Hypothesis Testing:** To date, very few FDA approaches are able to quantify uncertainty. Instead, they are focusing on point estimations. Yet a model should acknowledge the confidence in prediction and provide a guideline to perform hypothesis testing. Along this line, I plan to continue working on federated Bayesian methods and provide a systematic framework that provides uncertainty quantification and model testing. (III) **Heterogeneity and Personalization:** The other line of my future research is to tackle statistical heterogeneity and

personalized problems. In FDA, statistical heterogeneity is a central challenge as individual devices may have different data patterns and potentially collect different amounts and types of data. As mentioned before, personalization is one way to overcome the heterogeneity challenge by allowing clients to retain their individualized models while still borrowing strength from each other. However, there are still several open questions: (i) it is crucial to decide when a personalized model is needed and provide a trade-off between a global model and personalized models; (ii) how to cluster devices with similar features such that devices within a group can build a "group-level global model". As data sharing is not practical in FDA, clustering devices becomes extremely challenging; (iii) how to avoid adversarial attacks from malicious devices. As some devices tend to provide adversarial attacks to the central server and deteriorate aggregated information, it is important to detect those devices and take proper action.

3. **Creating real-life federated and distributed engineering data repository.** As FDA is still in its infancy phase, real-life datasets in engineering areas such as manufacturing, energy, and healthcare are pressingly needed to fully explore the disruptive potential of FDA. However, there is a limited number of existing engineering datasets. In the near future, I plan to collaborate with professors/students from manufacturing, mechanical engineering, civil engineering, transportation, and healthcare to collect real-life data and create a central directory for federated and distributed real-life engineering datasets. For example, I plan to purchase a few 3D printers and construct a small FDA environment to collect several federated 3D printing datasets. I believe this is a fruitful process that helps researchers unveil the potential challenges and opportunities faced within different domains. I will continue to build this central directory to promote collaboration and communication.

4. **Research collaboration and exploration.** Finally, I would like to mention that one of the most wonderful parts of research is the collaboration process. Research without any communication or collaboration is like an ivory tower, isolating researchers from the real world. During my past four years of studies, I have collaborated with many incredible collaborators to study parameter calibrations and physics-informed neural networks [7, 10]. The latter work [10] won the first prize in the IISE QCRE/ProcessMiner Data Challenge Competition. Though those are not my expertise area, it is amazing to explore new research directions by communicating with domain experts from that area. As such, I plan to seek research or industrial collaborations actively.

FDA is a relatively new and underdeveloped area, yet I envision that this is a very promising direction in the Engineering community, especially in the smart-and-connected IoT-enabled system.

I am extremely motivated to continue my research endeavors and strive to make the engineering community bloom.

APPENDIX A

Appendix for Chapter 2

A.1 Appendix

In Sec. A.2, we restate our main assumptions. In Sec. A.3, we provide the detailed proofs of Lemmas and Theorems in our main paper. Finally, in Sec. A.4, we present some additional empirical results.

A.2 Assumptions

We make the following assumptions.

Assumption 35.1. *F_k is L-smooth and μ-strongly convex for all $k \in [K]$.*

Assumption 35.2. *Denote by $\zeta_k^{(t)}$ the batched data from client k and $g_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)})$ the stochastic gradient calculated on this batched data. The variance of stochastic gradients are bounded. Specifically,*

$$\mathbb{E}\Big\{ \Big\| g_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) - \nabla F_k(\boldsymbol{\theta}_k^{(t)}) \Big\|^2 \Big\} \leq \sigma_k^2, \forall k \in [K].$$

It can be shown that, at local iteration t during communication round c,

$$\begin{aligned}
&\mathbb{E}\Big\{ \Big\| \nabla H_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) - \nabla H_k(\boldsymbol{\theta}_k^{(t)}) \Big\|^2 \Big\} \\
&= \mathbb{E}\Big\{ \Big\| (1 + \frac{\lambda r_k^c}{p_k |\mathcal{A}_{s_k}|}) g_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) - (1 + \frac{\lambda r_k^c}{p_k |\mathcal{A}_{s_k}|}) \nabla F_k(\boldsymbol{\theta}_k^{(t)}) \Big\|^2 \Big\} \\
&\leq (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k^c)^2 \sigma_k^2, \forall k \in [K].
\end{aligned}$$

Here, $\nabla H_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)})$ denotes the stochastic gradient of H_k evaluated on the batched data $\zeta_k^{(t)}$.

Assumption 35.3. *The expected squared norm of stochastic gradient is bounded. Specifically,*

$$\mathbb{E}\left\{ \left\| g_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) \right\|^2 \right\} \leq G^2, \forall k \in [K].$$

It can be shown that, at local iteration t during communication round c,

$$\mathbb{E}\left\{ \left\| \nabla H_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) \right\| \right\} = \mathbb{E}\left\{ \left\| (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k^c) g_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) \right\|^2 \right\}$$
$$\leq (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k^c)^2 G^2, \forall k \in [K].$$

For the non-convex setting, we replace Assumption 35.1 by the following assumption.

Assumption 35.4. *F_k is L-smooth for all $k \in [K]$.*

In our proof, for the sake of neatness, we drop the superscript of r_k^c.

We use the definition in Li et al. (2019b) to roughly quantify the degree of non-$i.i.d.$-ness. Specifically,

$$\Gamma_K = H^* - \sum_{k=1}^{K} p_k H_k^* = \sum_{k=1}^{K} p_k (H^* - H_k^*).$$

If data from all sensitive attributes are $i.i.d.$, then $\Gamma_K = 0$ as number of clients grows. Otherwise, $\Gamma_K \neq 0$ (Li et al. 2019b).

A.3 Detailed Proof

A.3.1 Proof of Lemma

Lemma 36. *For any given $\boldsymbol{\theta}$, the global objective function $H(\boldsymbol{\theta})$ defined in the main paper can be expressed as*

$$H(\boldsymbol{\theta}) = \sum_{k=1}^{K} \left(p_k + \frac{\lambda}{|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}) \right) F_k(\boldsymbol{\theta}),$$

where

$$r_k(\boldsymbol{\theta}) \triangleq \sum_{1 \leq j \neq s_k \leq d} \text{sign}(L_{s_k}(\boldsymbol{\theta}) - L_j(\boldsymbol{\theta}))$$

and $s_k \in [d]$ *is the group index of device* k*. Consequently,*

$$H(\boldsymbol{\theta}) = \sum_{k=1}^{K} p_k H_k(\boldsymbol{\theta}).$$

Proof. By definition, at communication round c,

$$\begin{aligned}
H(\boldsymbol{\theta}) &= \sum_{k=1}^{K} p_k F_k(\boldsymbol{\theta}) + \lambda \sum_{1 \le i < j \le d} |L_i(\boldsymbol{\theta}) - L_j(\boldsymbol{\theta})| \\
&= \sum_{k=1}^{K} p_k F_k(\boldsymbol{\theta}) + \lambda \sum_{1 \le i < j \le d} \left| \frac{1}{|\mathcal{A}_i|} \sum_{k \in \mathcal{A}_i} F_k(\boldsymbol{\theta}) - \frac{1}{|\mathcal{A}_j|} \sum_{k \in \mathcal{A}_j} F_k(\boldsymbol{\theta}) \right| \\
&= \sum_{k=1}^{K} p_k F_k(\boldsymbol{\theta}) + \lambda \sum_{1 \le i < j \le d} \text{sign}(L_i(\boldsymbol{\theta}) - L_j(\boldsymbol{\theta})) \left(\frac{1}{|\mathcal{A}_i|} \sum_{k \in \mathcal{A}_i} F_k(\boldsymbol{\theta}) - \frac{1}{|\mathcal{A}_j|} \sum_{k \in \mathcal{A}_j} F_k(\boldsymbol{\theta}) \right) \\
&= \sum_{k=1}^{K} p_k F_k(\boldsymbol{\theta}) + \lambda \sum_{u=1}^{d-1} \sum_{u < j \le d} \text{sign}(L_u(\boldsymbol{\theta}) - L_j(\boldsymbol{\theta})) \left(\frac{1}{|\mathcal{A}_u|} \sum_{k \in \mathcal{A}_u} F_k(\boldsymbol{\theta}) - \frac{1}{|\mathcal{A}_j|} \sum_{k \in \mathcal{A}_j} F_k(\boldsymbol{\theta}) \right) \\
&= \sum_{k=1}^{K} p_k F_k(\boldsymbol{\theta}) + \lambda \sum_{u=1}^{d} \sum_{k \in \mathcal{A}_u} \sum_{u \ne j \le d} \text{sign}(L_u(\boldsymbol{\theta}) - L_j(\boldsymbol{\theta})) \frac{F_k(\boldsymbol{\theta})}{|\mathcal{A}_u|} \\
&= \sum_{k=1}^{K} p_k F_k(\boldsymbol{\theta}) + \sum_{k=1}^{K} \frac{\lambda}{|\mathcal{A}_{s_k}|} \sum_{1 \le j \ne s_k \le d} \text{sign}(L_{s_k}(\boldsymbol{\theta}) - L_j(\boldsymbol{\theta})) F_k(\boldsymbol{\theta}) \\
&= \sum_{k=1}^{K} \left(p_k + \frac{\lambda}{|\mathcal{A}_{s_k}|} r_k^c(\boldsymbol{\theta}) \right) F_k(\boldsymbol{\theta}).
\end{aligned}$$

The fifth equality is achieved by rearranging the equation and merging items with the same group label. By definition of H_k, we thus proved

$$H(\boldsymbol{\theta}) = \sum_{k=1}^{K} p_k H_k(\boldsymbol{\theta}).$$

□

A.3.2 Learning Bound

We present a generalization bound for our learning model. Denote by $\mathcal{G}$ the family of the losses associated to a hypothesis set $\mathcal{H}$: $\mathcal{G} = \{(x, y) \mapsto \ell(h(x), y) : h \in \mathcal{H}\}$. The weighted Rademacher complexity (Mohri et al. 2019) is defined as

$$\mathfrak{R}_{\boldsymbol{m}}(\mathcal{G}, \boldsymbol{p}) \coloneqq \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{H}} \sum_{k=1}^{K} \frac{p_k}{N_k} \sum_{n=1}^{N_k} \sigma_{k,n} \ell(h(x_{k,n}), y_{k,n}) \right]$$

where $\boldsymbol{m} = (N_1, N_2, \ldots, N_k)$, $\boldsymbol{p} = (p_1, \ldots, p_K)$ and $\boldsymbol{\sigma} = (\sigma_{k,n})_{k \in [K], n \in [N_k]}$ is a collection of Rademacher variables taking values in $\{-1, +1\}$. Denote by $\mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h)$ the expected loss according to our fairness formulation. Denote by $\hat{\mathcal{L}}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h)$ the expected empirical loss (See Appendix for a detailed expression).

Theorem 37. *Assume that the loss ℓ is bounded above by $M > 0$. Fix $\epsilon_0 > 0$ and $\boldsymbol{m}$. Then, for any $\delta_0 > 0$, with probability at least $1 - \delta_0$ over samples $D_k \sim \mathcal{D}_k$, the following holds for all $h \in \mathcal{H}$:*

$$\mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) \leq \hat{\mathcal{L}}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) + \sqrt{\frac{1}{2} \sum_{k=1}^{K} (\frac{p_k}{N_k} M + \lambda \frac{d(d-1)}{2} M)^2 \log \frac{1}{\delta_0}} + 2\mathfrak{R}_{\boldsymbol{m}}(\mathcal{G}, \boldsymbol{p}) + \lambda \frac{d(d-1)}{2} M.$$

It can be seen that, given a sample of data, we can bound the generalization error $\mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) - \hat{\mathcal{L}}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h)$ with high probability. When $\lambda = 0$, the bound is same as the generalization bound in `FedAvg` (Mohri et al. 2018). When we consider the worst combination of p_k by taking the supremum of the upper bound in Theorem 37 and let $\lambda = 0$, then our generalization bound is same as the one in `AFL` (Mohri et al. 2019).

Proof. Define

$$\Phi(D_1, \ldots, D_K) = \sup_{h \in \mathcal{H}} \left(\mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) - \hat{\mathcal{L}}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) \right).$$

Let $D' = (D'_1, \ldots, D'_K)$ be a sample differing from $D = (D_1, \ldots, D_K)$ only by one point $x'_{k,n}$. Therefore, we have

$$\begin{aligned} \Phi(D') - \Phi(D) &= \sup_{h \in \mathcal{H}} \left(\mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) - \hat{\mathcal{L}}_{\mathcal{D}'^{\lambda}_{\boldsymbol{p}}}(h) \right) - \sup_{h \in \mathcal{H}} \left(\mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) - \hat{\mathcal{L}}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) \right) \\ &\leq \sup_{h \in \mathcal{H}} \left(\mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) - \hat{\mathcal{L}}_{\mathcal{D}'^{\lambda}_{\boldsymbol{p}}}(h) \right) - \left(\mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) - \hat{\mathcal{L}}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) \right) \\ &\leq \sup_{h \in \mathcal{H}} \left\{ \sup_{h \in \mathcal{H}} \mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) - \sup_{h \in \mathcal{H}} \hat{\mathcal{L}}_{\mathcal{D}'^{\lambda}_{\boldsymbol{p}}}(h) - \mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) + \hat{\mathcal{L}}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) \right\} \\ &= \sup_{h \in \mathcal{H}} \left\{ \hat{\mathcal{L}}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) - \hat{\mathcal{L}}_{\mathcal{D}'^{\lambda}_{\boldsymbol{p}}}(h) \right\} \end{aligned}$$

By definition,

$$\hat{\mathcal{L}}_{\mathcal{D}'^{\lambda}_{\boldsymbol{p}}}(h) = \sum_{k=1}^{K} \frac{p_k}{N_k} \sum_{n=1}^{N_k} \ell(h(x'_{k,n}), y'_{k,n}) + \\ \lambda \sum_{1 \le i < j \le d} | \frac{\sum_{k \in \mathcal{A}_i} \frac{1}{N_k} \sum_{n=1}^{N_k} \ell(h(x'_{k,n}), y'_{k,n})}{|\mathcal{A}_i|} - \frac{\sum_{k \in \mathcal{A}_j} \frac{1}{N_k} \sum_{n=1}^{N_k} \ell(h(x'_{k,n}), y'_{k,n})}{|\mathcal{A}_j|} |.$$

Therefore,

$$\begin{aligned} &\sup_{h \in \mathcal{H}} \left\{ \hat{\mathcal{L}}_{\mathcal{D}^{\lambda}_{\boldsymbol{p}}}(h) - \hat{\mathcal{L}}_{\mathcal{D}'^{\lambda}_{\boldsymbol{p}}}(h) \right\} \\ &\le \sup_{h \in \mathcal{H}} \left[\frac{p_k}{N_k} (\ell(h(x'_{k,n}), y'_{k,n}) - \ell(h(x_{k,n}), y_{k,n})) + \lambda \frac{d(d-1)}{2} M \right] \\ &\le \frac{p_k}{N_k} M + \lambda \frac{d(d-1)}{2} M. \end{aligned}$$

By McDiarmid's inequality, for $\delta_0 = \mathsf{exp}\left(\frac{-2\epsilon_0^2}{\sum_{k=1}^{K} (\frac{p_k}{N_k} M + \lambda \frac{d(d-1)}{2} M)^2} \right)$, the following holds with probability at least $1 - \delta_0$

$$\Phi(D) - \mathbb{E}_D[\Phi(D)] \le \epsilon_0 = \sqrt{\frac{1}{2} \sum_{k=1}^{K} (\frac{p_k}{N_k} M + \lambda \frac{d(d-1)}{2} M)^2 \log \frac{1}{\delta_0}}.$$

Our next goal is to bound $\mathbb{E}[\Phi(D)]$. We have

$$\begin{aligned}
\mathbb{E}_D[\Phi(D)] &= \mathbb{E}_D\left[\sup_{h\in\mathcal{H}}\left(\mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h) - \hat{\mathcal{L}}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h)\right)\right] \\
&= \mathbb{E}_D\left[\sup_{h\in\mathcal{H}}\mathbb{E}_{D'}\left(\hat{\mathcal{L}}_{\mathcal{D}'^{\lambda}_{\boldsymbol{p}}}(h) - \hat{\mathcal{L}}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h)\right)\right] \\
&\le \mathbb{E}_D\mathbb{E}_{D'}\sup_{h\in\mathcal{H}}\left(\hat{\mathcal{L}}_{\mathcal{D}'^{\lambda}_{\boldsymbol{p}}}(h) - \hat{\mathcal{L}}_{\mathcal{D}_{\boldsymbol{p}}^{\lambda}}(h)\right) \\
&\le \mathbb{E}_D\mathbb{E}_{D'}\sup_{h\in\mathcal{H}}\left[\sum_{k=1}^{K}\frac{p_k}{N_k}\sum_{n=1}^{N_k}\ell(h(x'_{k,n}), y'_{k,n})\right. \\
&\qquad \left. -\sum_{k=1}^{K}\frac{p_k}{N_k}\sum_{n=1}^{N_k}\ell(h(x_{k,n}), y_{k,n}) + \lambda\frac{d(d-1)}{2}M\right] \\
&\le \mathbb{E}_D\mathbb{E}_{D'}\mathbb{E}_{\boldsymbol{\sigma}}\sup_{h\in\mathcal{H}}\left[\sum_{k=1}^{K}\frac{p_k}{N_k}\sum_{n=1}^{N_k}\sigma_{k,n}\ell(h(x'_{k,n}), y'_{k,n})\right. \\
&\qquad \left. -\sum_{k=1}^{K}\frac{p_k}{N_k}\sum_{n=1}^{N_k}\sigma_{k,n}\ell(h(x_{k,n}), y_{k,n}) + \lambda\frac{d(d-1)}{2}M\right] \\
&\le 2\mathfrak{R}_{\boldsymbol{m}}(\mathcal{G}, \boldsymbol{p}) + \lambda\frac{d(d-1)}{2}M.
\end{aligned}$$

Therefore,

$$\Phi(D) \le \sqrt{\frac{1}{2}\sum_{k=1}^{K}(\frac{p_k}{N_k}M + \lambda\frac{d(d-1)}{2}M)^2\log\frac{1}{\delta_0}} + 2\mathfrak{R}_{\boldsymbol{m}}(\mathcal{G}, \boldsymbol{p}) + \lambda\frac{d(d-1)}{2}M.$$

□

A.3.3 Convergence (Strongly Convex)

Our proof is based on the convergence result of `FedAvg` (Li et al. 2019b).

Theorem 38. *Assume Assumptions in the main paper hold and $|\mathcal{S}_c| = K$. For $\gamma, \mu > 0$ and $\eta^{(t)}$ is decreasing in a rate of $\mathcal{O}(\frac{1}{t})$. If $\eta^{(t)} \le \mathcal{O}(\frac{1}{L})$, we have*

$$\mathbb{E}\left\{H(\bar{\boldsymbol{\theta}}^{(T)})\right\} - H^* \le \frac{L}{2}\frac{1}{\gamma + T}\left\{\frac{4\xi}{\epsilon^2\mu^2} + (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2\right\},$$

where $\xi = 8(E-1)^2G^2 + 4L\Gamma_K + 2\frac{\Gamma_{max}}{\eta^{(t)}} + 4\sum_{k=1}^{K}p_k^2\sigma_k^2$ and $\Gamma_{max} := \sum_{k=1}^{K}p_k|(H^* - H_k^*)| \ge |\sum_{k=1}^{K}p_k(H^* - H_k^*)| = |\Gamma_K|$.

Proof. For each device k, we introduce an intermediate model parameter $\boldsymbol{w}_k^{(t+1)} = \boldsymbol{\theta}_k^{(t)} - \eta^{(t)}\nabla H_k(\boldsymbol{\theta}_k^{(t)})$.

If iteration $t+1$ is in the communication round, then $\boldsymbol{\theta}_k^{(t+1)} = \sum_{k=1}^K p_k \boldsymbol{w}_k^{(t+1)}$ (i.e., aggregation). Otherwise, $\boldsymbol{\theta}_k^{(t+1)} = \boldsymbol{w}_k^{(t+1)}$. Define $\bar{\boldsymbol{w}}^{(t)} = \sum_{k=1}^K p_k \boldsymbol{w}_k^{(t)}$ and $\bar{\boldsymbol{\theta}}^{(t)} = \sum_{k=1}^K p_k \boldsymbol{\theta}_k^{(t)}$. Also, define $\boldsymbol{g}^{(t)} = \sum_{k=1}^K p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)})$ and $\bar{\boldsymbol{g}}^{(t)} = \mathbb{E}(\boldsymbol{g}^{(t)}) = \sum_{k=1}^K p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)})$.

Denote by $\boldsymbol{\theta}^*$ the optimal model parameter of the global objective function $H(\cdot)$. At iteration t, we have

$$\begin{aligned}
&\mathbb{E}\left\{ \left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \boldsymbol{\theta}^*\right\|^2 \right\} = \mathbb{E}\left\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \eta^{(t)}\boldsymbol{g}^{(t)} - \boldsymbol{\theta}^* - \eta^{(t)}\bar{\boldsymbol{g}}^{(t)} + \eta^{(t)}\bar{\boldsymbol{g}}^{(t)}\right\|^2 \right\} \\
&= \mathbb{E}\left\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^* - \eta^{(t)}\bar{\boldsymbol{g}}^{(t)}\right\|^2 \right\} + \mathbb{E}\left\{ 2\eta^{(t)}\langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^* - \eta^{(t)}\bar{\boldsymbol{g}}^{(t)}, \bar{\boldsymbol{g}}^{(t)} - \boldsymbol{g}^{(t)}\rangle \right\} \\
&\qquad + \mathbb{E}\left\{ \eta^{(t)2} \left\|\boldsymbol{g}^{(t)} - \bar{\boldsymbol{g}}^{(t)}\right\|^2 \right\} \\
&= \underbrace{\mathbb{E}\left\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^* - \eta^{(t)}\bar{\boldsymbol{g}}^{(t)}\right\|^2 \right\}}_{A} + \underbrace{\mathbb{E}\left\{ \eta^{(t)2} \left\|\boldsymbol{g}^{(t)} - \bar{\boldsymbol{g}}^{(t)}\right\|^2 \right\}}_{B},
\end{aligned}$$

since $\mathbb{E}\left\{ 2\eta^{(t)}\langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^* - \eta^{(t)}\bar{\boldsymbol{g}}^{(t)}, \bar{\boldsymbol{g}}^{(t)} - \boldsymbol{g}^{(t)}\rangle \right\} = 0$. Our remaining work is to bound term A and term B.

Part I: Bounding Term A We can split term A above into three parts:

$$\mathbb{E}\left\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^* - \eta^{(t)}\bar{\boldsymbol{g}}^{(t)}\right\|^2 \right\} = \mathbb{E}\left\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \right\} \underbrace{-2\eta^{(t)}\mathbb{E}\left\{ \langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*, \bar{\boldsymbol{g}}^{(t)}\rangle \right\}}_{\text{C}} + \underbrace{\eta^{(t)2}\mathbb{E}\left\{ \left\|\bar{\boldsymbol{g}}^{(t)}\right\|^2 \right\}}_{\text{D}}.$$

For part C, We have

$$\begin{aligned}
\text{C} &= -2\eta^{(t)}\mathbb{E}\left\{ \langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*, \bar{\boldsymbol{g}}^{(t)}\rangle \right\} = -2\eta^{(t)}\mathbb{E}\left\{ \sum_{k=1}^K p_k \langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*, \nabla H_k(\boldsymbol{\theta}_k^{(t)})\rangle \right\} \\
&= -2\eta^{(t)}\mathbb{E}\left\{ \sum_{k=1}^K p_k \langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}, \nabla H_k(\boldsymbol{\theta}_k^{(t)})\rangle \right\} - 2\eta^{(t)}\mathbb{E}\left\{ \sum_{k=1}^K p_k \langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, \nabla H_k(\boldsymbol{\theta}_k^{(t)})\rangle \right\}
\end{aligned}$$

To bound C, we need to use Cauchy-Schwarz inequality, inequality of arithmetic and geometric means. Specifically, the Cauchy-Schwarz inequality indicates that

$$\langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}, \nabla H_k(\boldsymbol{\theta}_k^{(t)})\rangle \geq -\left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}\right\| \left\|\nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|$$

and inequality of arithmetic and geometric means further implies

$$-\left\|\bar{\boldsymbol{\theta}}^{(t)}-\boldsymbol{\theta}_k^{(t)}\right\|\left\|\nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\| \geq -\frac{\left\|\bar{\boldsymbol{\theta}}^{(t)}-\boldsymbol{\theta}_k^{(t)}\right\|^2+\left\|\nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2}{2}.$$

Therefore, we obtain

$$\begin{aligned}
\mathrm{C} &= -2\eta^{(t)}\mathbb{E}\Big\{\langle\bar{\boldsymbol{\theta}}^{(t)}-\boldsymbol{\theta}^*,\bar{\boldsymbol{g}}^{(t)}\rangle\Big\} = -2\eta^{(t)}\mathbb{E}\Big\{\sum_{k=1}^K p_k\langle\bar{\boldsymbol{\theta}}^{(t)}-\boldsymbol{\theta}^*,\nabla H_k(\boldsymbol{\theta}_k^{(t)})\rangle\Big\}\\
&= -2\eta^{(t)}\mathbb{E}\Big\{\sum_{k=1}^K p_k\langle\bar{\boldsymbol{\theta}}^{(t)}-\boldsymbol{\theta}_k^{(t)},\nabla H_k(\boldsymbol{\theta}_k^{(t)})\rangle\Big\} - 2\eta^{(t)}\mathbb{E}\Big\{\sum_{k=1}^K p_k\langle\boldsymbol{\theta}_k^{(t)}-\boldsymbol{\theta}^*,\nabla H_k(\boldsymbol{\theta}_k^{(t)})\rangle\Big\}\\
&\leq \mathbb{E}\Big\{\eta^{(t)}\sum_{k=1}^K p_k\frac{1}{\eta^{(t)}}\left\|\bar{\boldsymbol{\theta}}^{(t)}-\boldsymbol{\theta}_k^{(t)}\right\|^2+\eta^{(t)^2}\sum_{k=1}^K p_k\left\|\nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2\\
&\qquad -2\eta^{(t)}\sum_{k=1}^K p_k(H_k(\boldsymbol{\theta}_k^{(t)})-H_k(\boldsymbol{\theta}^*)) - 2\eta^{(t)}\sum_{k=1}^K p_k\frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))\mu}{2}\left\|\boldsymbol{\theta}_k^{(t)}-\boldsymbol{\theta}^*\right\|^2\Big\},
\end{aligned}$$

where $-2\eta^{(t)}\mathbb{E}\Big\{\sum_{k=1}^K p_k\langle\boldsymbol{\theta}_k^{(t)}-\boldsymbol{\theta}^*,\nabla H_k(\boldsymbol{\theta}_k^{(t)})\rangle\Big\}$ is bounded by the property of strong convexity of H_k.

Since H_k is $(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L$-smooth, we know

$$\left\|\nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2 \leq 2(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L(H_k(\boldsymbol{\theta}_k^{(t)})-H_k^*)$$

and therefore

$$\begin{aligned}
\mathrm{D} &= \eta^{(t)2}\mathbb{E}\Big\{\left\|\bar{\boldsymbol{g}}^{(t)}\right\|^2\Big\} \leq \eta^{(t)2}\mathbb{E}\Big\{\sum_{k=1}^K p_k\left\|\nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2\Big\}\\
&\leq 2\eta^{(t)2}\mathbb{E}\Big\{\sum_{k=1}^K p_k(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L(H_k(\boldsymbol{\theta}_k^{(t)})-H_k^*)\Big\}
\end{aligned}$$

by convexity of norm.

Therefore, combining C and D, we have

$$\begin{aligned}
A &= \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^* - \eta^{(t)}\bar{\boldsymbol{g}}^{(t)}\right\|^2 \Big\} \\
&\leq \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} + 2\eta^{(t)2}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k(1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L(H_k(\boldsymbol{\theta}_k^{(t)}) - H_k^*)\Big\} \\
&\qquad + \eta^{(t)}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k\frac{1}{\eta^{(t)}}\left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}\right\|^2 \Big\} + {\eta^{(t)}}^2\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k\left\|\nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2 \Big\} \\
&\qquad - 2\eta^{(t)}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k(H_k(\boldsymbol{\theta}_k^{(t)}) - H_k(\boldsymbol{\theta}^*))\Big\} \\
&\qquad - 2\eta^{(t)}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k\frac{(1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))\mu}{2}\left\|\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
&\leq \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} - \eta^{(t)}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k(1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))\mu\left\|\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
&\qquad + \sum_{k=1}^{K} p_k\left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}\right\|^2 \\
&\quad \underbrace{+ 4\eta^{(t)2}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k(1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L(H_k(\boldsymbol{\theta}_k^{(t)}) - H_k^*)\Big\} - 2\eta^{(t)}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k(H_k(\boldsymbol{\theta}_k^{(t)}) - H_k(\boldsymbol{\theta}^*))\Big\}}_{\text{E}}.
\end{aligned}$$

In the last inequality, we simply rearrange other terms and use the fact that $\left\|\nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2 \leq 2(1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L(H_k(\boldsymbol{\theta}_k^{(t)}) - H_k^*)$ as aforementioned.

To bound E, we define $\gamma_k^{(t)} = 2\eta^{(t)}(1 - 2(1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L\eta^{(t)})$. Assume $\eta^{(t)} \leq \frac{1}{4(1+\frac{(d-1)}{\min\{p_k|\mathcal{A}_{s_k}|\}}\lambda)L}$, then we know $\eta^{(t)} \leq \gamma_k^{(t)} \leq 2\eta^{(t)}$.

Therefore, we have

$$\begin{aligned}
\mathrm{E} = &\, 4\eta^{(t)2}\mathbb{E}\bigg\{\sum_{k=1}^{K} p_k(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L(H_k(\boldsymbol{\theta}_k^{(t)})-H_k^*)\bigg\} \\
&\quad - 2\eta^{(t)}\mathbb{E}\bigg\{\sum_{k=1}^{K} p_k(H_k(\boldsymbol{\theta}_k^{(t)})-H_k(\boldsymbol{\theta}^*))\bigg\} \\
= &\, 4\eta^{(t)2}\mathbb{E}\bigg\{\sum_{k=1}^{K} p_k(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L(H_k(\boldsymbol{\theta}_k^{(t)})-H_k^*)\bigg\} \\
&\quad - 2\eta^{(t)}\mathbb{E}\bigg\{\sum_{k=1}^{K} p_k(H_k(\boldsymbol{\theta}_k^{(t)})-H_k^*+H_k^*-H_k(\boldsymbol{\theta}^*))\bigg\} \\
= &\, -2\eta^{(t)}\mathbb{E}\bigg\{\sum_{k=1}^{K} p_k(1-2(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L\eta^{(t)})(H_k(\boldsymbol{\theta}_k^{(t)})-H_k^*)\bigg\} \\
&\quad + 2\eta^{(t)}\mathbb{E}\bigg\{\sum_{k=1}^{K} p_k(H_k(\boldsymbol{\theta}^*)-H_k^*)\bigg\} \\
= &\, -\mathbb{E}\bigg\{\sum_{k=1}^{K} \gamma_k^{(t)}p_k(H_k(\boldsymbol{\theta}_k^{(t)})-H^*+H^*-H_k^*)\bigg\} + 2\eta^{(t)}\mathbb{E}\bigg\{H^*-\sum_{k=1}^{K}p_kH_k^*\bigg\} \\
= &\, \underbrace{-\mathbb{E}\bigg\{\sum_{k=1}^{K} \gamma_k^{(t)}p_k(H_k(\boldsymbol{\theta}_k^{(t)})-H^*)\bigg\}}_{\mathrm{F}} + \underbrace{\mathbb{E}\bigg\{\sum_{k=1}^{K}(2\eta^{(t)}-\gamma_k^{(t)})p_k(H^*-H_k^*)\bigg\}}_{\mathrm{G}}.
\end{aligned}$$

If $H^* - H_k^* \geq 0$ for some k, then $(2\eta^{(t)} - \gamma_k^{(t)})p_k(H^* - H_k^*) \leq 2\eta^{(t)}p_k(H^* - H_k^*)$. If $H^* - H_k^* < 0$ otherwise, then $(2\eta^{(t)} - \gamma_k^{(t)})p_k(H^* - H_k^*)$ is negative and $(2\eta^{(t)} - \gamma_k^{(t)})p_k(H^* - H_k^*) \leq -2\eta^{(t)}p_k(H^* - H_k^*)$. Therefore, by definition of Γ_{max},

$$\mathrm{G} \leq 2\eta^{(t)}\mathbb{E}\bigg\{\sum_{k=1}^{K} p_k|H^*-H_k^*|\bigg\} = 2\eta^{(t)}\Gamma_{max}.$$

The remaining goal of Part I is to bound term F. Note that

$$\begin{aligned}
\mathrm{F} &= -\mathbb{E}\Big\{\sum_{k=1}^{K}\gamma_k^{(t)}p_k(H_k(\boldsymbol{\theta}_k^{(t)})-H^*)\Big\}\\
&= -\mathbb{E}\Big\{\Big(\sum_{k=1}^{K}p_k\gamma_k^{(t)}(H_k(\boldsymbol{\theta}_k^{(t)})-H_k(\bar{\boldsymbol{\theta}}^{(t)}))+\sum_{k=1}^{K}p_k\gamma_k^{(t)}(H_k(\bar{\boldsymbol{\theta}}^{(t)})-H^*)\Big)\Big\}\\
&\leq -\mathbb{E}\Big\{\Big(\sum_{k=1}^{K}p_k\gamma_k^{(t)}\langle\nabla H_k(\bar{\boldsymbol{\theta}}^{(t)}),\boldsymbol{\theta}_k^{(t)}-\bar{\boldsymbol{\theta}}^{(t)}\rangle+\sum_{k=1}^{K}p_k\gamma_k^{(t)}(H_k(\bar{\boldsymbol{\theta}}^{(t)})-H^*)\Big)\Big\}\\
&\leq \mathbb{E}\Big\{\sum_{k=1}^{K}\frac{1}{2}\gamma_k^{(t)}p_k\Big[\eta^{(t)}\left\|\nabla H_k(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2+\frac{1}{\eta^{(t)}}\left\|\boldsymbol{\theta}_k^{(t)}-\bar{\boldsymbol{\theta}}^{(t)}\right\|^2\Big]-\sum_{k=1}^{K}p_k\gamma_k^{(t)}(H_k(\bar{\boldsymbol{\theta}}^{(t)})-H^*)\Big\}\\
&\leq \mathbb{E}\Big\{\sum_{k=1}^{K}\gamma_k^{(t)}p_k\Big[\eta^{(t)}(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L(H_k(\bar{\boldsymbol{\theta}}^{(t)})-H_k^*)+\frac{1}{2\eta^{(t)}}\left\|\boldsymbol{\theta}_k^{(t)}-\bar{\boldsymbol{\theta}}^{(t)}\right\|^2\Big]\\
&\qquad -\sum_{k=1}^{K}p_k\gamma_k^{(t)}(H_k(\bar{\boldsymbol{\theta}}^{(t)})-H^*)\Big\}.
\end{aligned}$$

In the second inequality, we again use the Cauchy–Schwarz inequality and Inequality of arithmetic and geometric means. In the last inequality, we use the fact that $\left\|\nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2 \leq 2(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L(H_k(\boldsymbol{\theta}_k^{(t)})-H_k^*)$.

Since $\eta^{(t)} \leq \gamma_k^{(t)} \leq 2\eta^{(t)}$, we can bound E as

$$\begin{aligned}
\mathrm{E} &\leq \mathrm{F} + \mathbb{E}\bigg\{2\eta^{(t)}\Gamma_{max}\bigg\} \\
&= (\eta^{(t)}(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L-1)\mathbb{E}\bigg\{\sum_{k=1}^{K}\gamma_k^{(t)}p_k\Big[(H_k(\bar{\boldsymbol{\theta}}^{(t)})-H^*)\Big]\bigg\} \\
&\qquad + \mathbb{E}\bigg\{\sum_{k=1}^{K}\eta^{(t)}\gamma_k^{(t)}p_k(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L(H^*-H_k^*)\bigg\} \\
&\qquad + \frac{1}{2\eta^{(t)}}\sum_{k=1}^{K}\gamma_k^{(t)}p_k\bigg\{\Big\|\boldsymbol{\theta}_k^{(t)}-\bar{\boldsymbol{\theta}}^{(t)}\Big\|^2\bigg\} + 2\eta^{(t)}\Gamma_{max} \\
&\leq \mathbb{E}\bigg\{\sum_{k=1}^{K}\eta^{(t)}\gamma_k^{(t)}p_k(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L(H^*-H_k^*)\bigg\} \\
&\qquad + \frac{1}{2\eta^{(t)}}\sum_{k=1}^{K}\gamma_k^{(t)}p_k\mathbb{E}\bigg\{\Big\|\boldsymbol{\theta}_k^{(t)}-\bar{\boldsymbol{\theta}}^{(t)}\Big\|^2\bigg\} + 2\eta^{(t)}\Gamma_{max} \\
&\leq \sum_{k=1}^{K}p_k\mathbb{E}\bigg\{\Big\|\boldsymbol{\theta}_k^{(t)}-\bar{\boldsymbol{\theta}}^{(t)}\Big\|^2\bigg\} + \mathbb{E}\bigg\{\sum_{k=1}^{K}\eta^{(t)}\gamma_k^{(t)}p_k(1+\frac{d-1}{p_k|\mathcal{A}_{s_k}|}\lambda)L(H^*-H_k^*)\bigg\} + 2\eta^{(t)}\Gamma_{max} \\
&\leq \sum_{k=1}^{K}p_k\mathbb{E}\bigg\{\Big\|\boldsymbol{\theta}_k^{(t)}-\bar{\boldsymbol{\theta}}^{(t)}\Big\|^2\bigg\} + 4\eta^{(t)2}L\mathbb{E}\bigg\{\sum_{k=1}^{K}p_k(H^*-H_k^*)\bigg\} + 2\eta^{(t)}\Gamma_{max} \\
&= \sum_{k=1}^{K}p_k\mathbb{E}\bigg\{\Big\|\boldsymbol{\theta}_k^{(t)}-\bar{\boldsymbol{\theta}}^{(t)}\Big\|^2\bigg\} + 4\eta^{(t)2}L\Gamma_K + 2\eta^{(t)}\Gamma_{max}
\end{aligned}$$

The second inequality holds because $(\eta^{(t)}(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L-1) \leq 0$ and the fourth inequality uses the fact that $1+\frac{d-1}{p_k|\mathcal{A}_{s_k}|}\lambda \leq 2$ based on the constraint of λ.

Therefore,

$$
\begin{aligned}
A \leq\ & \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} - \eta^{(t)}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k(1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))\mu \left\|\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
& + \sum_{k=1}^{K} p_k \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}\right\|^2 + \mathrm{E} \\
\leq\ & 2\sum_{k=1}^{K} p_k \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}\right\|^2 \Big\} + 4\eta^{(t)2} L\Gamma_K + 2\eta^{(t)}\Gamma_{max} + \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
& - \eta^{(t)}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k(1 - \frac{d-1}{p_k|\mathcal{A}_{s_k}|}\lambda)\mu \left\|\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
\leq\ & 2\sum_{k=1}^{K} p_k \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}\right\|^2 \Big\} + 4\eta^{(t)2} L\Gamma_K + 2\eta^{(t)}\Gamma_{max} + \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
& - \eta^{(t)}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k^2(1 - \frac{d-1}{p_k|\mathcal{A}_{s_k}|}\lambda)\mu \left\|\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
\leq\ & 2\sum_{k=1}^{K} p_k \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}\right\|^2 \Big\} + 4\eta^{(t)2} L\Gamma_K + 2\eta^{(t)}\Gamma_{max} + \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
& - \eta^{(t)}\mathbb{E}\Big\{ (1 - \frac{d-1}{\min\{p_k|\mathcal{A}_{s_k}|\}}\lambda)\mu \frac{1}{K} \left\|\sum_{k=1}^{K} p_k\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
=\ & 2\sum_{k=1}^{K} p_k \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}\right\|^2 \Big\} + 4\eta^{(t)2} L\Gamma_K + 2\eta^{(t)}\Gamma_{max} \\
& + (1 - \eta^{(t)}(1 - \frac{d-1}{\min\{p_k|\mathcal{A}_{s_k}|\}}\lambda)\frac{\mu}{K})\mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\}
\end{aligned}
$$

The third inequality uses the fact that $0 \leq p_k \leq 1$ and $-p_k^2 \geq -p_k$. The last inequality uses the fact that $\left\|\sum_{k=1}^{K} p_k\boldsymbol{\theta}_k\right\|^2 \leq K\sum_{k=1}^{K} \|p_k\boldsymbol{\theta}_k\|^2 = K\sum_{k=1}^{K} p_k^2 \|\boldsymbol{\theta}_k\|^2$ and $1 - \frac{d-1}{p_k|\mathcal{A}_{s_k}|}\lambda \geq 1 - \frac{d-1}{\min\{p_k|\mathcal{A}_{s_k}|\}}\lambda$.

Part II: Bounding Term $\sum_{k=1}^{K} p_k\mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}\right\|^2 \Big\}$ in Term A For any iteration $t \geq 0$, denote by $t_0 \leq t$ the index of previous communication iteration before t. Since the FL algorithm requires one communication each E steps, we know $t - t_0 \leq E - 1$ and $\boldsymbol{\theta}_k^{(t_0)} = \bar{\boldsymbol{\theta}}^{(t_0)}$. Assume $\eta^{(t)} \leq 2\eta^{(t+E)}$.

Since $\eta^{(t)}$ is decreasing, we have

$$
\begin{aligned}
\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k \left\| \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)} \right\|^2 \Big\} &= \mathbb{E}\Big\{ \sum_{k=1}^{K} p_k \left\| (\boldsymbol{\theta}_k^{(t)} - \bar{\boldsymbol{\theta}}^{(t_0)}) - (\bar{\boldsymbol{\theta}}^{(t)} - \bar{\boldsymbol{\theta}}^{(t_0)}) \right\|^2 \Big\} \\
&\leq \mathbb{E}\Big\{ \sum_{k=1}^{K} p_k \left\| \boldsymbol{\theta}_k^{(t)} - \bar{\boldsymbol{\theta}}^{(t_0)} \right\|^2 \Big\} \\
&= \mathbb{E}\Big\{ \sum_{k=1}^{K} p_k \left\| \sum_{t=0}^{t-1} \eta^{(t)} g_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) \right\|^2 \Big\} \\
&\leq \mathbb{E}\Big\{ \sum_{k=1}^{K} p_k (t - t_0) \sum_{t=0}^{t-1} \eta^{(t)2} \left\| g_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) \right\|^2 \Big\} \\
&\leq \sum_{k=1}^{K} p_k \sum_{t=t_0}^{t-1} (E-1) \eta^{(t)2} G^2 \leq \sum_{k=1}^{K} p_k \sum_{t=t_0}^{t-1} (E-1) \eta^{(t_0)2} G^2 \\
&\leq \sum_{k=1}^{K} p_k (E-1)^2 \eta^{(t_0)2} G^2 \leq 4\eta^{(t)2} (E-1)^2 G^2.
\end{aligned}
$$

Part III: Bounding Term B By assumption, it is easy to show

$$
\mathbb{E}\Big\{ \eta^{(t)2} \left\| \boldsymbol{g}^{(t)} - \bar{\boldsymbol{g}}^{(t)} \right\|^2 \Big\} \leq \eta^{(t)2} \sum_{k=1}^{K} p_k^2 (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2 \sigma_k^2.
$$

Part IV: Proving Convergence So far, we have shown that

$$
\begin{aligned}
&\mathbb{E}\Big\{ \left\| \bar{\boldsymbol{\theta}}^{(t+1)} - \boldsymbol{\theta}^* \right\|^2 \Big\} \leq \mathrm{A} + \mathrm{B} \\
&\leq 8\eta^{(t)2}(E-1)^2 G^2 + 4\eta^{(t)2} L \Gamma_K + 2\eta^{(t)} \Gamma_{max} + (1 - \eta^{(t)}(1 - \frac{d-1}{p_k |\mathcal{A}_{s_k}|}\lambda)\mu) \mathbb{E}\Big\{ \left\| \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^* \right\|^2 \Big\} \\
&\qquad + \eta^{(t)2} \sum_{k=1}^{K} p_k^2 (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2 \sigma_k^2 \\
&= (1 - \eta^{(t)}(1 - \frac{d-1}{\min\{p_k |\mathcal{A}_{s_k}|\}}\lambda)\frac{\mu}{K}) \mathbb{E}\Big\{ \left\| \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^* \right\|^2 \Big\} + \eta^{(t)^2} \xi
\end{aligned}
$$

where $\xi = 8(E-1)^2 G^2 + 4L\Gamma_K + 2\frac{\Gamma_{max}}{\eta^{(t)}} + \sum_{k=1}^{K} p_k^2 (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2 \sigma_k^2$.

Let $\eta^{(t)} = \frac{\beta}{t+\gamma}$ with $\beta > \frac{1}{(1 - \frac{d-1}{min\{p_k |\mathcal{A}_{s_k}|\}}\lambda)\frac{\mu}{K}}$ and $\gamma > 0$. Define $\epsilon := (1 - \frac{d-1}{min\{p_k |\mathcal{A}_{s_k}|\}}\lambda)$. Let $v = \max\{\frac{\beta^2 \xi}{\beta \epsilon \mu - 1}, (\gamma + 1) \left\| \bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^* \right\|^2\}$. We will show that $\left\| \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^* \right\|^2 \leq \frac{v}{\gamma + t}$ by induction. For $t = 0$, we have $\left\| \bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^* \right\|^2 \leq (\gamma + 1) \left\| \bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^* \right\|^2 \leq \frac{v}{\gamma + 1}$. Now assume this is true for some t, then

$$\begin{aligned}
\mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \boldsymbol{\theta}^*\right\|^2 \Big\} &\leq (1-\eta^{(t)}\epsilon\mu)\mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} + {\eta^{(t)}}^2\xi \\
&\leq (1-\frac{\beta\epsilon\mu}{t+\gamma})\frac{v}{t+\gamma} + \frac{\beta^2\xi}{(t+\gamma)^2} \\
&= \frac{t+\gamma-1}{(t+\gamma)^2}v + \frac{\beta^2\xi}{(t+\gamma)^2} - \frac{\beta\epsilon\mu-1}{(t+\gamma)^2}v.
\end{aligned}$$

It is easy to show $\frac{t+\gamma-1}{(t+\gamma)^2}v + \frac{\beta^2\xi}{(t+\gamma)^2} - \frac{\beta\epsilon\mu-1}{(t+\gamma)^2}v \leq \frac{v}{t+\gamma+1}$ by definition of v. Therefore, we proved $\left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \leq \frac{v}{\gamma+t}$.

By definition, we know H is $\sum_{k=1}^K p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))}{2}L$-smooth. Therefore,

$$\begin{aligned}
\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) \Big\} - H^* &\leq \frac{\sum_{k=1}^K p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))}{2}L}{2}\mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
&\leq \frac{\sum_{k=1}^K p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))}{2}L}{2}\frac{v}{\gamma+t}.
\end{aligned}$$

By choosing $\beta = \frac{2}{\epsilon\frac{\mu}{K}}$ We have

$$\begin{aligned}
&v = \max\{\frac{\beta^2\xi}{\beta\epsilon\mu-1}, (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2\} \leq \frac{\beta^2\xi}{\beta\epsilon\mu-1} + (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2 \\
&\leq \frac{4\xi}{\epsilon^2\mu^2} + (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(T)}) \Big\} - H^* &\leq \frac{\sum_{k=1}^K p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))}{2}L}{2}\frac{1}{\gamma+T}\Big\{ \frac{4\xi}{\epsilon^2\mu^2} + (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
&\leq \frac{\sum_{k=1}^K p_k \frac{(1+\frac{\lambda(d-1)}{p_k|\mathcal{A}_{s_k}|})}{2}L}{2}\frac{1}{\gamma+T}\Big\{ \frac{4\xi}{\epsilon^2\mu^2} + (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
&\leq \frac{L}{2}\frac{1}{\gamma+T}\Big\{ \frac{4\xi}{\epsilon^2\mu^2} + (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2 \Big\}.
\end{aligned}$$

We thus proved our convergence result. □

Theorem 39. *Assume at each communication round, central server sampled a fraction α of devices and those local devices are sampled according to the sampling probability p_k. Additionally, assume*

Assumptions in the main paper hold. For $\gamma, \mu, \epsilon > 0$, we have

$$\mathbb{E}\left\{H(\bar{\boldsymbol{\theta}}^{(T)})\right\} - H^* \leq \frac{L}{2}\frac{1}{\gamma + T}\left\{\frac{4(\xi+\tau)}{\epsilon^2\mu^2} + (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2\right\},$$

$\tau = \frac{E^2}{\lceil \alpha K \rceil}\sum_{k=1}^{K} p_k(1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))^2 G^2.$

Proof.

$$\begin{aligned}
&\mathbb{E}\left\{\left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \boldsymbol{\theta}^*\right\|^2\right\} = \mathbb{E}\left\{\left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{w}}^{(t+1)} + \bar{\boldsymbol{w}}^{(t+1)} - \boldsymbol{\theta}^*\right\|^2\right\} \\
&= \mathbb{E}\left\{\left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{w}}^{(t+1)}\right\|^2 + \left\|\bar{\boldsymbol{w}}^{(t+1)} - \boldsymbol{\theta}^*\right\|^2 + 2\langle\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{w}}^{(t+1)}, \bar{\boldsymbol{w}}^{(t+1)} - \boldsymbol{\theta}^*\rangle\right\} \\
&= \mathbb{E}\left\{\left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{w}}^{(t+1)}\right\|^2 + \left\|\bar{\boldsymbol{w}}^{(t+1)} - \boldsymbol{\theta}^*\right\|^2\right\}.
\end{aligned}$$

Note that the expectation is taken over subset $\mathcal{S}_c$.

Part I: Bounding Term $\mathbb{E}\left\{\left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{w}}^{(t+1)}\right\|^2\right\}$ Assume $\lceil \alpha K \rceil$ number of local devices are sampled according to sampling probability p_k. During the communication round, we have $\bar{\boldsymbol{\theta}}^{t+1} = \frac{1}{\lceil \alpha K \rceil}\sum_{l=1}^{\lceil \alpha K \rceil}\boldsymbol{w}_l^{(t+1)}$. Therefore,

$$\begin{aligned}
\mathbb{E}\left\{\left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{w}}^{(t+1)}\right\|^2\right\} &= \mathbb{E}\left\{\frac{1}{\lceil \alpha K \rceil^2}\left\|\sum_{l=1}^{\lceil \alpha K \rceil}\boldsymbol{w}_l^{(t+1)} - \bar{\boldsymbol{w}}^{(t+1)}\right\|^2\right\} \\
&= \mathbb{E}\left\{\frac{1}{\lceil \alpha K \rceil^2}\sum_{l=1}^{\lceil \alpha K \rceil}\left\|\boldsymbol{w}_l^{(t+1)} - \bar{\boldsymbol{w}}^{(t+1)}\right\|^2\right\} \\
&= \frac{1}{\lceil \alpha K \rceil}\sum_{k=1}^{K} p_k\left\|\boldsymbol{w}_k^{(t+1)} - \bar{\boldsymbol{w}}^{(t+1)}\right\|^2.
\end{aligned}$$

We know

$$\begin{aligned}
&\sum_{k=1}^{K} p_k\left\|\boldsymbol{w}_k^{(t+1)} - \bar{\boldsymbol{w}}^{(t+1)}\right\|^2 = \sum_{k=1}^{K} p_k\left\|(\boldsymbol{w}_k^{(t+1)} - \bar{\boldsymbol{\theta}}^{(t_0)}) - (\bar{\boldsymbol{w}}^{(t+1)} - \bar{\boldsymbol{\theta}}^{(t_0)})\right\|^2 \\
&\leq \sum_{k=1}^{K} p_k\left\|(\boldsymbol{w}_k^{(t+1)} - \bar{\boldsymbol{\theta}}^{(t_0)})\right\|^2,
\end{aligned}$$

where $t_0 = t - E + 1$. Similarly,

$$\begin{aligned}
\mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{w}}^{(t+1)}\right\|^2 \Big\} &\leq \frac{1}{\lceil \alpha K \rceil}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k \left\|(\boldsymbol{w}_k^{(t+1)} - \bar{\boldsymbol{\theta}}^{(t_0)})\right\|^2 \Big\} \\
&\leq \frac{1}{\lceil \alpha K \rceil}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k \left\|(\boldsymbol{w}_k^{(t+1)} - \boldsymbol{\theta}_k^{(t_0)})\right\|^2 \Big\} \\
&\leq \frac{1}{\lceil \alpha K \rceil}\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k E \sum_{m=t_o}^{t} \left\|\eta^{(m)} \nabla H_k(\boldsymbol{\theta}_k^{(m)}; \zeta_k^{(t)})\right\|^2 \Big\} \\
&\leq \frac{E^2 \eta^{(t_0)2}}{\lceil \alpha K \rceil} \sum_{k=1}^{K} p_k (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2 G^2 \\
&\leq \frac{E^2 \eta^{(t)2}}{\lceil \alpha K \rceil} \sum_{k=1}^{K} p_k (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2 G^2
\end{aligned}$$

using the fact that $\eta^{(t)}$ is non-increasing in t.

Part II: Convergence Result As aforementioned,

$$\begin{aligned}
&\mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
&= \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{w}}^{(t+1)}\right\|^2 + \left\|\bar{\boldsymbol{w}}^{(t+1)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \\
&\leq \frac{E^2 \eta^{(t)2}}{\lceil \alpha K \rceil} \sum_{k=1}^{K} p_k (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2 G^2 + (1 - \eta^{(t)} \epsilon \frac{\mu}{K}) \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} + \eta^{(t)^2} \xi \\
&= (1 - \eta^{(t)} \epsilon \frac{\mu}{K}) \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} + \eta^{(t)^2} \Big(\xi + \frac{E^2}{\lceil \alpha K \rceil} \sum_{k=1}^{K} p_k (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2 G^2 \Big).
\end{aligned}$$

Let $\tau = \frac{E^2}{\lceil \alpha K \rceil} \sum_{k=1}^{K} p_k (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2 G^2$. Let $\eta^{(t)} = \frac{\beta}{t+\gamma}$ with $\beta > \frac{1}{\epsilon \frac{\mu}{K}}$ and $\gamma > 0$. Let $v = \max\{ \frac{\beta^2(\xi+\tau)}{\beta \epsilon \mu - 1}, (\gamma + 1) \left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2 \}$. Similar to the full device participation scenario, we can show that $\mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\} \leq \frac{v}{\gamma + t}$ by induction.

By definition, we know H is $\sum_{k=1}^K p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))}{2} L$-smooth. Therefore,

$$\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) \Big\} - H^* \leq \frac{\sum_{k=1}^K p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))}{2} L}{2} \mathbb{E}\Big\{ \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|^2 \Big\}$$
$$\leq \frac{\sum_{k=1}^K p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))}{2} L}{2} \frac{v}{\gamma + t}.$$

By choosing $\beta = \frac{2}{\epsilon \frac{\mu}{K}}$ We have

$$v = \max\{\frac{\beta^2 \xi}{\beta\epsilon\mu - 1}, (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2\} \leq \frac{\beta^2\xi}{\beta\epsilon\mu - 1} + (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2$$
$$\leq \frac{4\xi}{\epsilon^2\mu^2} + (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2.$$

Therefore,

$$\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(T)}) \Big\} - H^* \leq \frac{\sum_{k=1}^K p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))}{2} L}{2} \frac{1}{\gamma + T} \Big\{ \frac{4(\xi+\tau)}{\epsilon^2\mu^2} + (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2 \Big\}$$
$$\leq \frac{L}{2}\frac{1}{\gamma + T} \Big\{ \frac{4(\xi+\tau)}{\epsilon^2\mu^2} + (\gamma+1)\left\|\bar{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\right\|^2 \Big\}$$

□

A.3.4 Convergence (Non-convex)

Lemma 40. *If $\eta^{(t)} \leq \frac{2}{L}$, then $\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) \Big\} \leq \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(0)}) \Big\}$.*

Proof.

$$\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t+1)}) \Big\} = \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)} - \eta^{(t)} \sum_{k=1}^K p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)})) \Big\}$$
$$= \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)} - \eta^{(t)} \sum_{k=1}^K p_k \nabla H_k(\bar{\boldsymbol{\theta}}^{(t)}; \zeta_k^{(t)})) \Big\}$$
$$= \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)} - \eta^{(t)} g^{(t)}(\bar{\boldsymbol{\theta}}^{(t)})) \Big\}$$

Here we used the fact that $\bar{\boldsymbol{\theta}}^{(t)} = \boldsymbol{\theta}_k^{(t)}$ since the aggregated model parameter has been distributed to local devices. By Taylor's theorem, there exists a $\boldsymbol{w}^{(t)}$ such that

$$\begin{aligned}
&\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t+1)}) \Big\} \\
&= \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) - \eta^{(t)} g^{(t)}(\bar{\boldsymbol{\theta}}^{(t)})^T g^{(t)}(\bar{\boldsymbol{\theta}}^{(t)}) + \frac{1}{2}(\eta^{(t)} g^{(t)}(\bar{\boldsymbol{\theta}}^{(t)}))^T g^{(t)}(\boldsymbol{w}^{(t)})(\eta^{(t)} g^{(t)}(\bar{\boldsymbol{\theta}}^{(t)})) \Big\} \\
&\leq \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) - \eta^{(t)} g^{(t)}(\bar{\boldsymbol{\theta}}^{(t)})^T g^{(t)}(\bar{\boldsymbol{\theta}}^{(t)}) + \eta^{(t)2} \frac{\sum_{k=1}^K p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))}{2} L}{2} \left\| g^{(t)}(\bar{\boldsymbol{\theta}}^{(t)}) \right\|^2 \Big\} \\
&\leq \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) \Big\} - \eta^{(t)} \left\| g^{(t)}(\bar{\boldsymbol{\theta}}^{(t)}) \right\|^2 + \eta^{(t)2} \frac{L}{2} \left\| g^{(t)}(\bar{\boldsymbol{\theta}}^{(t)}) \right\|^2
\end{aligned}$$

since H is $\sum_{k=1}^K p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))}{2} L$-smooth. It can be shown that if $\eta^{(t)} \leq \frac{2}{L}$, we have

$$-\eta^{(t)} \left\| g^{(t)}(\bar{\boldsymbol{\theta}}^{(t)}) \right\|^2 + \eta^{(t)2} \frac{L}{2} \left\| g^{(t)}(\bar{\boldsymbol{\theta}}^{(t)}) \right\|^2 \leq 0.$$

Therefore, By choosing $\eta^{(t)} \leq \frac{2}{L}$, we proved $\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) \Big\} \leq \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(0)}) \Big\}$. □

Theorem 41. *Assume Assumptions in the main paper hold and $|\mathcal{S}_c| = K$. If $\eta^{(t)} = \mathcal{O}(\frac{1}{\sqrt{t}})$ and $\eta^{(t)} \leq \mathcal{O}(\frac{1}{L})$, then for > 0*

$$\min_{t=1,\dots,T} \mathbb{E}\Big\{ \left\| \nabla H(\bar{\boldsymbol{\theta}}^{(t)}) \right\|^2 \Big\} \leq \frac{1}{\sqrt{T}} \Big\{ 2(1 + 2KL^2 \sum_{t=1}^T \eta^{(t)2}) \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(0)}) - H^* \Big\} + 2\sum_{t=1}^T \xi^{(t)} \Big\},$$

where $\xi^{(t)} = 2KL^2\eta^{(t)2}\Gamma_K + (8\eta^{(t)3}KL^2(E-1) + 8KL\eta^{(t)2} + 4(2+4L)KL\eta^{(t)4}(E-1))G^2 + (2L\eta^{(t)2} + 8KL\eta^{(t)2})\sum_{k=1}^K p_k\sigma_k^2$

Proof. Since H is $\sum_{k=1}^K p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))}{2} L$-smooth, we have

$$\begin{aligned}
\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t+1)}) \Big\} \leq \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) \Big\} + \underbrace{\mathbb{E}\Big\{ \langle \nabla H(\bar{\boldsymbol{\theta}}^{(t)}), \bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{\theta}}^{(t)} \rangle \Big\}}_{\text{A}} + \\
\frac{\sum_{k=1}^K p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))}{2} L}{2} \underbrace{\mathbb{E}\Big\{ \left\| \bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{\theta}}^{(t)} \right\|^2 \Big\}}_{\text{B}}.
\end{aligned}$$

Part I: Bounding Term A We have

$$\begin{aligned}
\mathrm{A} &= -\eta^{(t)}\mathbb{E}\Big\{\langle \nabla H(\bar{\boldsymbol{\theta}}^{(t)}), \sum_{k=1}^{K} p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)})\rangle\Big\} = -\eta^{(t)}\mathbb{E}\Big\{\langle \nabla H(\bar{\boldsymbol{\theta}}^{(t)}), \sum_{k=1}^{K} p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)})\rangle\Big\} \\
&= -\frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\Big\} - \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\sum_{k=1}^{K} p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2\Big\} \\
&\quad + \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)}) - \sum_{k=1}^{K} p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2\Big\} \\
&= -\frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\Big\} - \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\sum_{k=1}^{K} p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2\Big\} \\
&\quad + \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\sum_{k=1}^{K} p_k \nabla H_k(\bar{\boldsymbol{\theta}}^{(t)}) - \sum_{k=1}^{K} p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2\Big\} \\
&\le -\frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\Big\} - \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\sum_{k=1}^{K} p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2\Big\} \\
&\quad + \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{K\sum_{k=1}^{K} p_k \left\|\nabla H_k(\bar{\boldsymbol{\theta}}^{(t)}) - \nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2\Big\} \\
&\le -\frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\Big\} - \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\sum_{k=1}^{K} p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2\Big\} + \\
&\quad \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{K\sum_{k=1}^{K} p_k((1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L)^2 \underbrace{\left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}\right\|^2}_{\mathrm{C}}\Big\}.
\end{aligned}$$

In the convex setting, we proved that

$$\mathrm{C} \le 4\eta^{(t)2}(E-1)G^2.$$

This is also true for the non-convex setting since we do not use any property of convex functions.

Part II: Bounding Term B We have

$$\begin{aligned}
\mathrm{B} &= \mathbb{E}\Big\{ \left\|\eta^{(t)} g^{(t)}\right\|^2 \Big\} = \mathbb{E}\Big\{ \Big\|\eta^{(t)} \sum_{k=1}^{K} p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)})\Big\|^2 \Big\} \\
&= \mathbb{E}\Big\{ \Big\|\eta^{(t)} \sum_{k=1}^{K} p_k (\nabla H_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) - \nabla H_k(\boldsymbol{\theta}_k^{(t)}))\Big\|^2 \Big\} + \mathbb{E}\Big\{ \Big\|\eta^{(t)} \sum_{k=1}^{K} p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)})\Big\|^2 \Big\} \\
&= \eta^{(t)2} \sum_{k=1}^{K} p_k^2 \mathbb{E}\Big\{ \left\|\nabla H_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) - \nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2 \Big\} + \mathbb{E}\Big\{ \Big\|\eta^{(t)} \sum_{k=1}^{K} p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)})\Big\|^2 \Big\} \\
&\leq \eta^{(t)2} \sum_{k=1}^{K} p_k^2 (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2 \sigma_k^2 + \eta^{(t)2} \mathbb{E}\Big\{ K \sum_{k=1}^{K} p_k^2 \left\|\nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2 \Big\}.
\end{aligned}$$

Since H_k is $(1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))L$-smooth, we know

$$\left\|\nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2 \leq 2(1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))L(H_k(\boldsymbol{\theta}_k^{(t)}) - H_k^*).$$

Therefore,

$$\begin{aligned}
\mathrm{B} \leq{} & \eta^{(t)2} \sum_{k=1}^{K} p_k^2 (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2 \sigma_k^2 + \\
& \eta^{(t)2} \mathbb{E}\Big\{ K \sum_{k=1}^{K} 2 p_k^2 (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta})) L (H_k(\boldsymbol{\theta}_k^{(t)}) - H_k^*) \Big\} \\
={} & \eta^{(t)2} \sum_{k=1}^{K} p_k^2 (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2 \sigma_k^2 + \\
& \eta^{(t)2} \mathbb{E}\Big\{ K \sum_{k=1}^{K} 2 p_k^2 (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta})) L (H_k(\boldsymbol{\theta}_k^{(t)}) - H^* + H^* - H_k^*) \Big\} \\
\leq{} & \eta^{(t)2} \sum_{k=1}^{K} p_k^2 (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2 \sigma_k^2 + \\
& \eta^{(t)2} \mathbb{E}\Big\{ K \sum_{k=1}^{K} 2 p_k (1 + \frac{\lambda}{p_k |\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta})) L (H_k(\boldsymbol{\theta}_k^{(t)}) - H^* + H^* - H_k^*) \Big\}
\end{aligned}$$

since $0 \leq p_k \leq 1$ and $p_k^2 \leq p_k$.

Therefore,

$$
\begin{aligned}
&\mathbb{E}\Big\{H(\bar{\boldsymbol{\theta}}^{(t+1)})\Big\}\\
&\leq \mathbb{E}\Big\{H(\bar{\boldsymbol{\theta}}^{(t)})\Big\} - \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\Big\} \underbrace{-\frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\sum_{k=1}^{K} p_k \nabla H_k(\boldsymbol{\theta}_k^{(t)})\right\|^2\Big\}}_{\text{D<0}} +\\
&\quad \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{K\sum_{k=1}^{K} p_k((1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L)^2 4\eta^{(t)2}(E-1)G^2\Big\}\\
&\quad + \frac{\sum_{k=1}^{K} p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))}{2}}{2}L\Big[\eta^{(t)2}\sum_{k=1}^{K} p_k^2(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))^2\sigma_k^2\\
&\quad + \eta^{(t)2}\mathbb{E}\Big\{K\sum_{k=1}^{K} 2p_k(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L(H_k(\boldsymbol{\theta}_k^{(t)}) - H^* + H^* - H_k^*)\Big\}\Big]\\
&\leq \mathbb{E}\Big\{H(\bar{\boldsymbol{\theta}}^{(t)})\Big\} - \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\Big\}\\
&\quad \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{K\sum_{k=1}^{K} p_k((1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))L)^2 4\eta^{(t)2}(E-1)G^2\Big\}\\
&\quad + \frac{\sum_{k=1}^{K} p_k \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))}{2}}{2}L\Big[\eta^{(t)2}\sum_{k=1}^{K} p_k^2(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|}r_k(\boldsymbol{\theta}))^2\sigma_k^2+\\
&\quad \underbrace{4KL\eta^{(t)2}\mathbb{E}\Big\{\sum_{k=1}^{K} p_k(H_k(\boldsymbol{\theta}_k^{(t)}) - H^*) + \sum_{k=1}^{K} p_k(H^* - H_k^*)\Big\}}_{\text{E}}\Big]\\
&\leq \mathbb{E}\Big\{H(\bar{\boldsymbol{\theta}}^{(t)})\Big\} - \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\Big\}\\
&\quad + \frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{K\sum_{k=1}^{K} 4p_k L^2 4\eta^{(t)2}(E-1)G^2\Big\}\\
&+ \frac{L}{2}\Big[\eta^{(t)2}\sum_{k=1}^{K} 4p_k^2\sigma_k^2 + \underbrace{4KL\eta^{(t)2}\mathbb{E}\Big\{\sum_{k=1}^{K} p_k(H_k(\boldsymbol{\theta}_k^{(t)}) - H^*) + \sum_{k=1}^{K} p_k(H^* - H_k^*)\Big\}}_{\text{E}}\Big]
\end{aligned}
$$

Here

$$
\begin{aligned}
&\text{E} \\
&= 4KL\eta^{(t)2}\mathbb{E}\Big\{\sum_{k=1}^{K} p_k(H_k(\boldsymbol{\theta}_k^{(t)}) - H^*)\Big\} + 4KL\eta^{(t)2}\mathbb{E}\Big\{\sum_{k=1}^{K} p_k(H^* - H_k^*)\Big\} \\
&= 4KL\eta^{(t)2}\mathbb{E}\Big\{\sum_{k=1}^{K} p_k(H_k(\boldsymbol{\theta}_k^{(t)}) - H_k(\bar{\boldsymbol{\theta}}^{(t)}))\Big\} + 4KL\eta^{(t)2}\mathbb{E}\Big\{\sum_{k=1}^{K} p_k(H_k(\bar{\boldsymbol{\theta}}^{(t)}) - H^*)\Big\} \\
&\qquad + 4KL\eta^{(t)2}\Gamma_K \\
&= 4KL\eta^{(t)2}\underbrace{\mathbb{E}\Big\{\sum_{k=1}^{K} p_k(H_k(\boldsymbol{\theta}_k^{(t)}) - H_k(\bar{\boldsymbol{\theta}}^{(t)}))\Big\}}_{\text{F}} + 4KL\eta^{(t)2}\mathbb{E}\Big\{H(\bar{\boldsymbol{\theta}}^{(t)}) - H^*\Big\} + 4KL\eta^{(t)2}\Gamma_K.
\end{aligned}
$$

We can bound term F as

$$
\begin{aligned}
\text{F} &= \mathbb{E}\Big\{\sum_{k=1}^{K} p_k(H_k(\boldsymbol{\theta}_k^{(t)}) - H_k(\bar{\boldsymbol{\theta}}^{(t)}))\Big\} \\
&\leq \mathbb{E}\Big\{\sum_{k=1}^{K} p_k(\langle \nabla H_k(\bar{\boldsymbol{\theta}}^{(t)}), \boldsymbol{\theta}_k^{(t)} - \bar{\boldsymbol{\theta}}^{(t)}\rangle + \frac{(1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))L}{2} \underbrace{\left\|\boldsymbol{\theta}_k^{(t)} - \bar{\boldsymbol{\theta}}^{(t)}\right\|^2}_{\leq 4\eta^{(t)2}(E-1)G^2})\Big\}
\end{aligned}
$$

where we use the fact that H_k is $(1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))L$-smooth. To bound the inner product, we again use the inequality of arithmetic and geometric means and Cauchy–Schwarz inequality:

$$
\langle \nabla H_k(\bar{\boldsymbol{\theta}}^{(t)}), \boldsymbol{\theta}_k^{(t)} - \bar{\boldsymbol{\theta}}^{(t)}\rangle \leq \left\|\nabla H_k(\bar{\boldsymbol{\theta}}^{(t)})\right\| \left\|\boldsymbol{\theta}_k^{(t)} - \bar{\boldsymbol{\theta}}^{(t)}\right\| \leq \frac{\left\|\nabla H_k(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2 + \left\|\boldsymbol{\theta}_k^{(t)} - \bar{\boldsymbol{\theta}}^{(t)}\right\|^2}{2}.
$$

It can be shown that

$$
\begin{aligned}
\mathbb{E}\Big\{\left\|\nabla H_k(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\Big\} &= \mathbb{E}\Big\{\left\|\nabla F_k(\boldsymbol{\theta}_k^{(t)}, D_k^{(t)})\right\|\Big\}^2 + \mathbb{E}\Big\{\left\|\nabla F_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) - \nabla F_k(\boldsymbol{\theta}_k^{(t)})\right\|^2\Big\} \\
&\leq \mathbb{E}\Big\{\left\|\nabla F_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)})\right\|^2\Big\} + \mathbb{E}\Big\{\left\|\nabla F_k(\boldsymbol{\theta}_k^{(t)}; \zeta_k^{(t)}) - \nabla F_k(\boldsymbol{\theta}_k^{(t)})\right\|^2\Big\} \\
&\leq (1 + \frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))^2(G^2 + \sigma_k^2) \leq 4(G^2 + \sigma_k^2)
\end{aligned}
$$

Therefore, we can simplify F as

$$
\begin{aligned}
\mathrm{F} &\leq \mathbb{E}\Big\{ \sum_{k=1}^{K} p_k\Big(\frac{\left\|\nabla H_k(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2 + \left\|\boldsymbol{\theta}_k^{(t)} - \bar{\boldsymbol{\theta}}^{(t)}\right\|^2}{2} + \frac{(1+\frac{\lambda}{p_k|\mathcal{A}_{s_k}|} r_k(\boldsymbol{\theta}))L}{2} \underbrace{\left\|\boldsymbol{\theta}_k^{(t)} - \bar{\boldsymbol{\theta}}^{(t)}\right\|^2}_{\leq 4\eta^{(t)2}(E-1)G^2} \Big) \Big\} \\
&\leq \mathbb{E}\Big\{ \sum_{k=1}^{K} p_k\Big(\frac{4(G^2+\sigma_k^2) + 4\eta^{(t)2}(E-1)G^2}{2} + 4L\eta^{(t)2}(E-1)G^2 \Big) \Big\} \\
&= 2\mathbb{E}\Big\{ \sum_{k=1}^{K} p_k\sigma_k^2 \Big\} + 2G^2 + (2+4L)\eta^{(t)2}(E-1)G^2
\end{aligned}
$$

Combining with E, we obtain

$$
\begin{aligned}
\mathrm{E} \leq 4KL\eta^{(t)2}&\Big(2\sum_{k=1}^{K} p_k\sigma_k^2 + 2G^2 + (2+4L)\eta^{(t)2}(E-1)G^2 \Big) \\
&+ 4KL\eta^{(t)2}\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) - H^* \Big\} + 4KL\eta^{(t)2}\Gamma_K
\end{aligned}
$$

Part III: Proving Convergence Therefore,

$$
\begin{aligned}
&\frac{1}{2}\eta^{(t)}\mathbb{E}\Big\{ \left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2 \Big\} \\
&\leq \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) \Big\} - \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t+1)}) \Big\} + \frac{1}{2}\eta^{(t)3}\mathbb{E}\Big\{ K\sum_{k=1}^{K} 4p_kL^2 4(E-1)G^2 \Big\} + \\
&\frac{L}{2}\Big[\eta^{(t)2}\sum_{k=1}^{K} 4p_k^2\sigma_k^2 + 4KL\eta^{(t)2}\Big(2\sum_{k=1}^{K} p_k\sigma_k^2 + 2G^2 + (2+4L)\eta^{(t)2}(E-1)G^2 \Big) \\
&\qquad + 4KL\eta^{(t)2}\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) - H^* \Big\} \\
&\qquad + 4KL\eta^{(t)2}\Gamma_K \Big] \\
&= \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) \Big\} - \mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t+1)}) \Big\} + 2KL^2\eta^{(t)2}\mathbb{E}\Big\{ H(\bar{\boldsymbol{\theta}}^{(t)}) - H^* \Big\} + 2KL^2\eta^{(t)2}\Gamma_K \\
&\qquad + (8\eta^{(t)3}KL^2(E-1) + 8KL\eta^{(t)2} + 4(2+4L)KL\eta^{(t)4}(E-1))G^2 \\
&\qquad + (2L\eta^{(t)2} + 8KL\eta^{(t)2})\sum_{k=1}^{K} p_k\sigma_k^2.
\end{aligned}
$$

Let $\xi^{(t)} = 2KL^2\eta^{(t)2}\Gamma_K + (8\eta^{(t)3}KL^2(E-1) + 8KL\eta^{(t)2} + 4(2+4L)KL\eta^{(t)4}(E-1))G^2 + (2L\eta^{(t)2} + 8KL\eta^{(t)2})\sum_{k=1}^{K} p_k\sigma_k^2$, then

$$\begin{aligned}
&\frac{1}{2}\eta^{(t)}\mathbb{E}\bigg\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\bigg\} \\
&\le \mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(t)})\bigg\} - \mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(t+1)})\bigg\} + 2KL^2\eta^{(t)2}\mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(t)}) - H^*\bigg\} + \xi^{(t)} \\
&\le \mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(t)})\bigg\} - \mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(t+1)})\bigg\} + 2KL^2\eta^{(t)2}\mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(0)}) - H^*\bigg\} + \xi^{(t)}
\end{aligned}$$

since $\eta^{(t)} \le \frac{1}{\sqrt{2KL}}$ and $\mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(t)})\bigg\} \le \mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(0)})\bigg\}$ by Lemma 44. By taking summation on both side, we obtain

$$\begin{aligned}
&\sum_{t=1}^{T}\frac{1}{2}\eta^{(t)}\mathbb{E}\bigg\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\bigg\} \\
&\le \mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(0)})\bigg\} - \mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(t+1)})\bigg\} + 2KL^2\sum_{t=1}^{T}\eta^{(t)2}\mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(0)}) - H^*\bigg\} + \sum_{t=1}^{T}\xi^{(t)} \\
&\le \mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(0)})\bigg\} - \mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{*})\bigg\} + 2KL^2\sum_{t=1}^{T}\eta^{(t)2}\mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(0)}) - H^*\bigg\} + \sum_{t=1}^{T}\xi^{(t)} \\
&= (1 + 2KL^2\sum_{t=1}^{T}\eta^{(t)2})\mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(0)}) - H^*\bigg\} + \sum_{t=1}^{T}\xi^{(t)}.
\end{aligned}$$

This implies

$$\min_{t=1,\ldots,T}\mathbb{E}\bigg\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\bigg\}\sum_{t=1}^{T}\eta^{(t)} \le 2(1 + 2KL^2\sum_{t=1}^{T}\eta^{(t)2})\mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(0)}) - H^*\bigg\} + 2\sum_{t=1}^{T}\xi^{(t)}$$

and therefore

$$\min_{t=1,\ldots,T}\mathbb{E}\bigg\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\bigg\} \le \frac{1}{\sum_{t=1}^{T}\eta^{(t)}}\bigg\{2(1 + 2KL^2\sum_{t=1}^{T}\eta^{(t)2})\mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(0)}) - H^*\bigg\} + 2\sum_{t=1}^{T}\xi^{(t)}\bigg\}.$$

Let $\eta^{(t)} = \frac{1}{\sqrt{t}}$, then we have $\sum_{t=1}^{T}\eta^{(t)} = \mathcal{O}(\sqrt{T})$ and $\sum_{t=1}^{T}\eta^{(t)2} = \mathcal{O}(\log(T+1))$. Therefore,

$$\min_{t=1,\ldots,T}\mathbb{E}\bigg\{\left\|\nabla H(\bar{\boldsymbol{\theta}}^{(t)})\right\|^2\bigg\} \le \frac{1}{\sqrt{T}}\bigg\{2(1 + 2KL^2\sum_{t=1}^{T}\eta^{(t)2})\mathbb{E}\bigg\{H(\bar{\boldsymbol{\theta}}^{(0)}) - H^*\bigg\} + 2\sum_{t=1}^{T}\xi^{(t)}\bigg\}.$$

□

A.4 Additional Experiments

We conduct a sensitivity analysis using the FEMNIST-3-groups setting. Results are reported in Figure A.1. Similar to the observation in the main paper, it can be seen that as λ increases, the discrepancy between two groups decreases accordingly. Here kindly note that we did not plot group 3 for the sake of neatness. The line of group should stay in the middle of two lines.

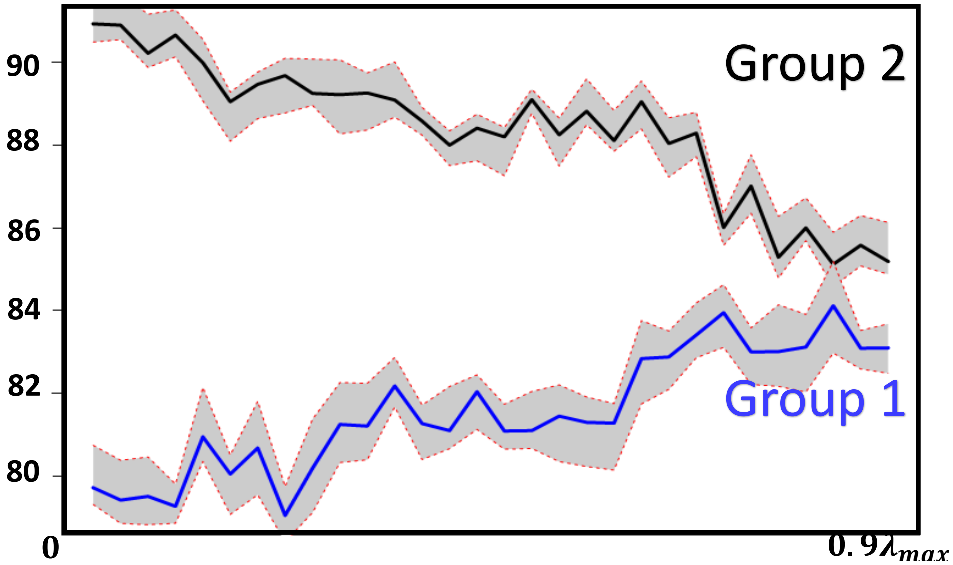

Figure A.1: Sensitivity analysis on FEMNIST

APPENDIX B

Appendix for Chapter 3

This chapter does not have an Appendix.

APPENDIX C

Appendix for Chapter 4

C.1 Convergence Plot

The convergence plots of Example 1 in the main paper are provided below. In Figure C.1(c), we observe some fluctuations due to the small sample size. In this setting, since $N = 800$, each device only has $M_k = 8$ data points on average. Recall that in the theoretical analysis, we have shown that the convergence rate is also affected by $\mathcal{O}(\sum_{k=1}^{K} p_k M_k^{\epsilon_k - 0.5})$.

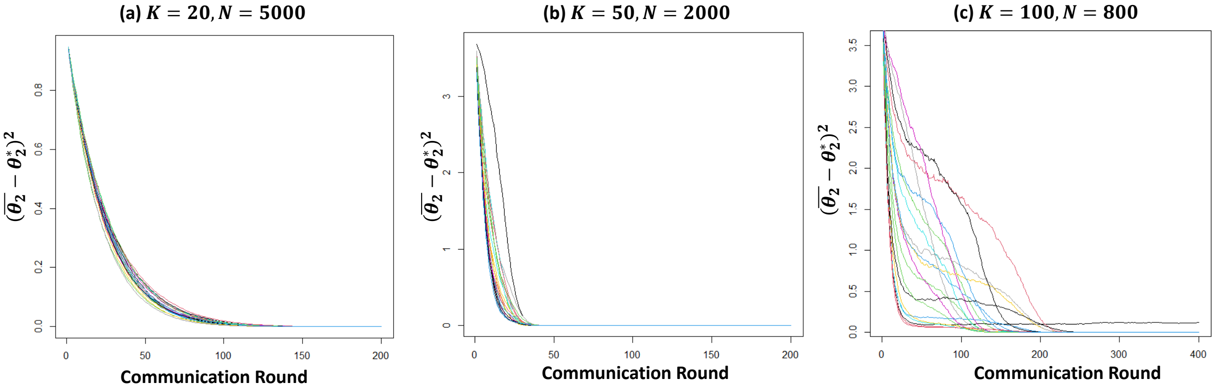

Figure C.1: (Matérn$-3/2$ kernel) Evolution of $\left\|\bar{\theta}_2 - \theta_2^*\right\|_2^2$ over training epochs. The input dimension is 1. In the plot, each color represents an independent run.

C.2 Multi-fidelity Modeling

Example 3: CURRIN The CURRIN (Currin et al. 1991, Xiong et al. 2013) is a two-dimensional function widely used for multi-fidelity computer simulation models. Given the input domain $\boldsymbol{x} \in [0, 1]^2$, the high-fidelity model is

$$y_h(\boldsymbol{x}) = \left[1 - \exp\left(-\frac{1}{2x_2}\right)\right] \frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20}$$

whereas the low-fidelity model is given by

$$
\begin{aligned}
y_l(\boldsymbol{x}) = &\frac{1}{4}[y_h(x_1+0.05, x_2+0.05) + y_h(x_1+0.05, \max(0, x_2-0.05))] \\
&+ \frac{1}{4}[y_h(x_1-0.05, x_2+0.05) + y_h(x_1-0.05, \max(0, x_2-0.05))].
\end{aligned}
$$

We collect 40 data points from the HF model and 200 data points from the LF model. The number of testing data points is 1,000.

Example 4: PARK The PARK function (Cox et al. 2001, Xiong et al. 2013) is a four-dimensional function ($\boldsymbol{x} \in (0,1]^4$) where the high-fidelity model is given as

$$
y_h(\boldsymbol{x}) = \frac{x_1}{2}\left[\sqrt{1+(x_2+x_3^2)\frac{x_4}{x_1^2}} - 1\right] + (x_1+3x_4)\mathsf{exp}[1+\sin(x_3)],
$$

while the low-fidelity model is

$$
y_l(\boldsymbol{x}) = \left[1+\frac{\sin(x_1)}{10}\right] y_h(\boldsymbol{x}) - 2x_1 + x_2^2 + x_3^2 + 0.5.
$$

Example 5: BRANIN In this example, there are three fidelity levels (Perdikaris et al. 2017, Cutajar et al. 2019):

$$
\begin{aligned}
y_h &= \left(\frac{-1.275x_1^2}{\pi^2} + \frac{5x_1}{\pi} + x_2 - 6\right)^2 + \left(10 - \frac{5}{4\pi}\right)\cos(x_1) + 10, \\
y_m &= 10\sqrt{y_h(\boldsymbol{x}-2)} + 2(x_1-0.5) - 3(3x_2-1) - 1, \\
y_l &= y_m(1.2(\boldsymbol{x}+2)) - 3x_2 + 1, \\
x &\in [-5, 10] \times [0, 15]
\end{aligned}
$$

where $y_m(\cdot)$ represents the output from the medium-fidelity (MF) model.

Example 6: Hartmann-3D Similar to Example 5, this is a 3-level multi-fidelity dataset where the input space is $[0,1]^3$. The evaluation of observations with fidelity t is defined as (Cutajar et al. 2019)

$$
y_t(\boldsymbol{x}) = \sum_{i=1}^{4} \alpha_i \mathsf{exp}\left(-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2\right)
$$

where

$$A = \begin{bmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}, P = \begin{bmatrix} 0.3689 & 0.1170 & 0.2673 \\ 0.4699 & 0.4387 & 0.7470 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.0381 & 0.5743 & 0.8828 \end{bmatrix},$$

$$\boldsymbol{\alpha} = (1.0, 1.2, 3.0, 3.2)^\intercal, \boldsymbol{\alpha}_t = \boldsymbol{\alpha} + (3-t)\boldsymbol{\delta}, \boldsymbol{\delta} = (0.01, -0.01, -0.1, 0.1)^\intercal.$$

Example 7: Borehole Model The Borehole model is an 8-dimensional physical model that simulates water flow through a borehole (Moon et al. 2012, Gramacy and Lian 2012, Xiong et al. 2013). The high-fidelity model is given as

$$y_h(\boldsymbol{x}) = \frac{2\pi x_3(x_4 - x_6)}{\ln(x_2/x_1)[1 + 2x_7x_3/(\ln(x_2/x_1)x_1^2x_8) + x_3/x_5]}$$

where $x_1 \in [0.05, 0.15], x_2 \in [100, 50000], x_3 \in [63070, 115600], x_4 \in [990, 1110], x_5 \in [63.1, 115]$, $x_6 \in [700, 820], x_7 \in [1120, 1680], x_8 \in [9855, 12045]$. The low-fidelity model is

$$y_l(\boldsymbol{x}) = \frac{5\pi x_3(x_4 - x_6)}{\ln(x_2/x_1)[1.5 + 2x_7x_3/(\ln(x_2/x_1)x_1^2x_8) + x_3/x_5]}.$$

C.3 Additional Application: NASA Aircraft Gas Turbine Engines

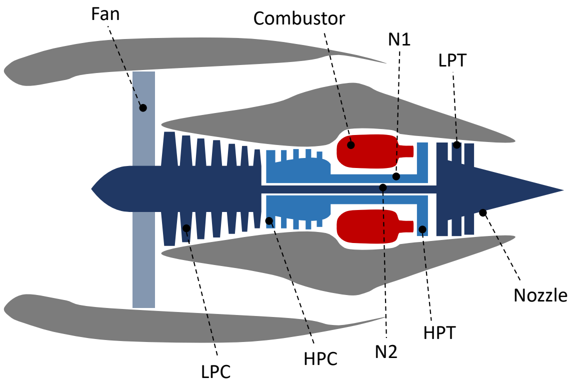

Figure C.2: The engine diagram in C-MAPSS.

In this case study, we consider degradation signals generated from aircraft gas turbine engines using the NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) tools (NASA dataset Link). The dataset consists of 100 engines and contains time-series degradation signals collected from multiple sensors installed on the engines. Figure C.2 illustrates the engine diagram in C-MAPSS. The experiment aims to predict the degradation signals for test engines in a federated paradigm. To do so, we assume that each client/device is a single engine, and all engines aim to collaboratively learn a predictive degradation model.

Table C.1: Averaged RMSE (line 1 in each cell) and standard deviation (std) of RMSE (line 2 in each cell) across all testing devices for the NASA data. Each experiment is repeated 30 times.

Averaged RMSE ×10 std of RMSE ×10	FGPR	Polynomial	Neural
Sensor 2	**5.45 (0.01)** **0.87 (0.02)**	6.79 (0.01) 0.98 (0.02)	6.47 (0.05) 1.02 (0.01)
Sensor 7	**5.76 (0.03)** **0.76 (0.01)**	6.55 (0.02) 0.89 (0.04)	6.71 (0.02) 0.85 (0.03)

We briefly describe our training procedures. We randomly divide the 100 engines into 60 training engines and 40 testing engines. For each testing unit k, we randomly split the data on each device into a 50% training dataset $D_{k,\text{train}} := (\boldsymbol{X}_{k,\text{train}}, \boldsymbol{y}_{k,\text{train}})$ and a 50% testing dataset $D_{k,\text{test}} := (\boldsymbol{X}_k^*, \boldsymbol{y}_k^*)$, where $\boldsymbol{y}_k^* = \left[y_{k,1}^*, ...y_{k,|D_{k,\text{test}}|}^*\right]^\intercal$, $\boldsymbol{X}_k^* = \left[{x_{k,1}^*}^\intercal, ..., {x_{k,|D_{k,\text{test}}|}^*}^\intercal\right]$. Recall that in the main paper, we define $|D_{k,\text{test}}|$ as the number of data points in the set $D_{k,\text{test}}$. We first train `FGPR` using the 60 training units and obtain a final aggregated global model parameter $\boldsymbol{\theta}$. The testing unit k then directly uses this global parameter $\boldsymbol{\theta}$ and $D_{k,\text{train}}$ to predict outputs $[f({x_{k,1}^*}^\intercal), \cdots, f({x_{k,|D_{k,\text{test}}|}^*}^\intercal)]$ at testing locations $\boldsymbol{X}_k^*$ without any additional training.

We benchmark `FGPR` with the following models.

1. `Polynomial`: All signal trajectories exhibit polynomial patterns, and therefore a polynomial regression is often employed to analyze this dataset (Liu et al. 2013, Yan et al. 2016, Song and Liu 2018). More specifically, we train a polynomial regression using `FedAvg`. During the training process, each device updates the coefficients of a polynomial regression in the form of $y_k(x) = \sum_{i=0}^{p} \beta_{ik} x^i + \epsilon_k(x)$, where $\{\beta_{ik}\}_{i=0}^{p}$ are model parameters. This update is done by running gradient descent to minimize the local sum squared error. The central server aggregates the parameters using `FedAvg` and broadcasts the aggregated parameter to all devices in the following communication round. Here, we conduct experiments with different $p \in \{1, \ldots, 20\}$ and select the best p with the smallest averaged testing RMSE. Our empirical study finds that $p = 10$ provides the best performance.

2. `Neural`: we train a q-layer neural network using `FedAvg` (McMahan et al. 2017). Similar to `Polynomial`, we test the performance of the neural network with different $q \in \{1, \ldots, 20\}$. The best value is 2.

The prediction performance of each model is measured by the averaged RMSE across all 40 testing devices.

The averaged RMSE and the standard deviation of RMSEs across all testing devices are reported in Table C.1. Each experiment is repeated 30 times. The outputs on each device are scaled to be a mean 0 and variance 1 sequence.

From Table C.1, we can obtain some important insights. First, FGPR consistently yields lower averaged RMSE than other benchmark models. This illustrates the good transferability of FGPR. More concretely, a shared global model can provide accurate surrogates even on untrained devices. This feature is in fact very helpful in transfer learning or online learning. For instance, the shared global model can be used as an initial parameter for fine-tuning on streaming data. Second, FGPR also provides smaller standard deviations of RMSEs across all devices. This credits to the automatic personalization feature encoded in $\mathcal{GP}$.

C.4 One Additional Illustrative Example

In this section, we provide another toy example to demonstrate why the global model may fail in deep networks. Consider a noiseless linear regression problem $y = \beta x$. Suppose there are two devices. Device 1 has the data that follows $y = 3x$ (i.e., $\beta_1^* = 3$) while device 2 has data that follows $y = x$ (i.e., $\beta_2^* = 1$). Each device has 100 training points uniformly spread in $[0, 1]$. If we use FedAvg to train neural networks, then the optimization problem is

$$\min_{\beta} \left(\|f_\beta - 3x\|_2^2 + \|f_\beta - x\|_2^2 \right),$$

where $\|\cdot\|_2^2$ is a functional on $[0, 1]$ defined as $\|f\|_2^2 = \int_0^1 f(x)^2 dx$. One can derive that the global optimal model parameter will be $\beta = 2$, and as a result, each device will fit a line $y = 2x$ that fails to predict the trend on any device.

C.5 Heterogeneous Setting

In this section, we consider the scenario where data from all devices are generated from several different processes or distributions. Equivalently, this indicates

$$\mathbb{P}\left(\left| \sum_{k=1}^{K} p_k L_k(\boldsymbol{\theta}^*; D_k) - \sum_{k=1}^{K} p_k L_k(\boldsymbol{\theta}_k^*; D_k) \right| = 0 \right) = 0.$$

Since the data are heterogeneous, the weighted average of $L_k(\boldsymbol{\theta}_k^*; D_k)$ can be very different from $L(\boldsymbol{\theta}^*)$.

Theorem 42. *(RBF kernels) Suppose Assumptions 1-3a hold. At each communication round, assume* $|\mathcal{S}| = K$. *If* $\eta^{(t)} = \mathcal{O}(\frac{1}{t})$, *then for some constants* $C_{\boldsymbol{\theta}}, c_{\boldsymbol{\theta}} > 0, \epsilon_k \in (0, \frac{1}{2})$, *when* $M_k > C_{\boldsymbol{\theta}}$, *at* *iteration T, with probability at least* $\min_k\{1 - C_{\boldsymbol{\theta}}(\log(M_k^{\epsilon_k - \frac{1}{2}}))^4 T \mathsf{exp}\{-c_{\boldsymbol{\theta}} M_k^{2\epsilon_k}\}\}$,

$$\left\|\nabla L(\bar{\boldsymbol{\theta}}^{(T)})\right\|_2^2 \leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{4\theta_{min}^4(T+1)} \\ + \mathcal{O}\left(\max_k \frac{\log M_k}{M_k} + \sum_{k=1}^K p_k M_k^{\epsilon_k - \frac{1}{2}}\right).$$

On the other hand, at each communication round, assume $|\mathcal{S}| = K_{sample}$ number of devices are sampled according to the sampling probability p_k, then we have

$$\mathbb{E}_{\mathcal{S}}\left\{\left\|\nabla L(\bar{\boldsymbol{\theta}}^{(T)})\right\|_2^2\right\} \leq \frac{2\beta_1^2\left(\frac{1}{|\mathcal{S}|}4E^2+8(E-1)^2+2\right)G^2}{4\theta_{min}^4(T+1)} \\ + \mathcal{O}\left(\max_k \frac{\log M_k}{M_k} + \sum_{k=1}^K p_k M_k^{\epsilon_k - \frac{1}{2}}\right).$$

Remark 43. *In the heterogeneous setting, we show that the* `FGPR` *algorithm will converge to a critical point of $L(\cdot)$ at a rate of $\mathcal{O}(\frac{1}{T})$ subject to a statistical error. The upper bound of $\left\|\nabla L(\bar{\boldsymbol{\theta}}^{(T)})\right\|_2^2$ has the same form as the one for $\left\|\bar{\theta}_2 - \theta_2^*\right\|_2^2$.*

For the Matérn Kernel, we have the same upper bounds on $\left\|\nabla L(\bar{\boldsymbol{\theta}}^{(T)})\right\|_2^2$ and $\mathbb{E}_{\mathcal{S}}\left\{\left\|\nabla L(\bar{\boldsymbol{\theta}}^{(T)})\right\|_2^2\right\}$ as those in the Theorem 3. This implies that the heterogeneous data distribution has little to no influence on the convergence behavior. The reason is that the heterogeneous definition is based on true parameters $\boldsymbol{\theta}^*$ and $\{\boldsymbol{\theta}_k^*\}_{k=1}^K$. The main Theorem, however, states that our algorithm will converge to a stationary point. In the non-convex scenario, the stationary point might be different from the true parameter.

C.6 Important Lemmas

In this section, we present some key lemmas used in our theoretical analysis. We defer the proofs of those Lemmas into Section C.8.

Lemma 44. *(Theorem 4 in Braun (2006)) Let Ker be a Mercer kernel on a probability space $\mathcal{X}$ with probability measure μ, satisfying Ker$(x,x) \leq 1$ for all $x \in \mathcal{X}$, with eigenvalues $\{\lambda_i^*\}_{i=1}^\infty$. Let $\boldsymbol{K}_{f,N}$ (i.e., $\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})$) be the empirical kernel matrix evaluated on data $\boldsymbol{X}$ $i.i.d.$ sampled from μ, then with probability at least $1-\delta$, the eigenvalues of $\lambda_j(\boldsymbol{K}_{f,N})$ satisfies the following bound for*

$1 \leq j \leq N$ and $1 \leq r \leq N$:

$$\left|\frac{\lambda_j(\boldsymbol{K}_{f,N})}{N} - \lambda_j^*\right| \leq \lambda_j^* C(r,N) + H(r,N),$$

where

$$C(r,N) < r\sqrt{\frac{2}{N\lambda_r^*}\log\frac{2r(r+1)}{\delta}} + \frac{4r}{3N\lambda_r^*}\log\frac{2r(r+1)}{\delta},$$
$$H(r,N) < \lambda_r^* + \sum_{i=r+1}^{\infty}\lambda_i^* + \sqrt{\frac{2\sum_{i=r+1}^{\infty}\lambda_i^*}{N}\log\frac{2}{\delta}} + \frac{2}{3N}\log\frac{2}{\delta}.$$

Alternatively, $C(r,N)$ and $H(r,N)$ can also be bounded as follows:

$$C(r,N) < r\sqrt{\frac{r(r+1)}{N\delta\lambda_r^*}},$$
$$H(R,N) < \lambda_r^* + \sum_{i=r+1}^{\infty}\lambda_i^* + \sqrt{\frac{2\sum_{i=r+1}^{\infty}\lambda_i^*}{N\delta}}.$$

This Lemma is proved in Braun (2006).

Lemma 45. *(Chen et al. 2020) Under Assumptions 1-3a, in device k, for any $0 < \epsilon_k, \alpha_k < 1$, $C_{1k}(\alpha_k, b_k) > 0$ and $N_k > C_{2k}(\epsilon_k, b_k)$, then with probability at least $1 - \frac{2}{N_k^{\alpha_k}}$, we have*

$$\frac{\epsilon_k \log N_k}{8b_k\theta_{max}^2} \leq \sum_{j=1}^{N_k}\frac{\lambda_{1j}^2}{\left(\theta_{1k}^{(t)}\lambda_{1j} + \theta_{2k}^{(t)}\lambda_{2j}\right)^2} \leq \frac{4+2\alpha_k}{b_k\theta_{min}^2}\log N_k$$
$$\frac{N_k - C_{1k}(\alpha_k, b_k)\log N_k}{4\theta_{max}^2} \leq \sum_{j=1}^{N_k}\frac{\lambda_{2j}^2}{\left(\theta_{1k}^{(t)}\lambda_{1j} + \theta_{2k}^{(t)}\lambda_{2j}\right)^2} \leq \frac{N_k}{\theta_{min}^2}$$
$$\sum_{j=1}^{N_k}\frac{\lambda_{1j}\lambda_{2j}}{\left(\theta_{1k}^{(t)}\lambda_{1j} + \theta_{2k}^{(t)}\lambda_{2j}\right)^2} \leq \frac{5+2\alpha_k}{7b_k\theta_{min}^2}\log N_k.$$

This Lemma is proved in Chen et al. (2020). Here note that we omit the subscript k in the eigenvalues λ for simplicity. The full notation should be, for example, λ_{1jk} for device k.

Lemma 46. *Under Assumption 1-2 and 3b, for any $0 < \alpha_k < \frac{8b_k^2 - 12b_k - 6}{4b_k + 3}$, with probability at least*

$1 - \frac{1}{N_k^{1+\alpha_k}}$*, the following inequalities hold:*

$$\sum_{j=1}^{N_k} \frac{\lambda_{1j}^2}{\left(\theta_{1k}^{(t)}\lambda_{1j} + \theta_{2k}^{(t)}\lambda_{2j}\right)^2} \leq N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}} \left(\frac{1}{\theta_{min}^2} + \frac{C_{mat,k}^2(4b_k+3)}{\theta_{min}^2(8b_k^2-8b_k-3)}\right),$$

$$\sum_{j=1}^{N_k} \frac{\lambda_{1j}\lambda_{2j}}{\left(\theta_{1k}^{(t)}\lambda_{1j} + \theta_{2k}^{(t)}\lambda_{2j}\right)^2} \leq N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}} \left(\frac{1}{\theta_{min}^2} + \frac{C_{mat,k}(4b_k+3)}{\theta_{min}^2(4b_k^2-6b_k-3)}\right),$$

$$\frac{N_k - C_{mat,k}N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}}}{4\theta_{max}} \leq \sum_{j=1}^{N_k} \frac{\lambda_{2j}^2}{\left(\theta_{1k}^{(t)}\lambda_{1j} + \theta_{2k}^{(t)}\lambda_{2j}\right)^2} \leq \frac{N_k}{\theta_{min}^2},$$

where $C_{mat,k}$ will be defined later.

Lemma 46 provides several bounds to constrain the eigenvalues of a Matérn kernel.

Lemma 47. *Under Assumption 1-3a, with probability at least $1-2TM_k^{-\alpha_k}$, the following inequality holds for any $k \in [K]$ and $0 \leq t < T$:*

$$\langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k^*(\boldsymbol{\theta}_k^{(t)})\rangle \geq \frac{\gamma_k}{2}\left\|\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}_k^*\right\|_2^2 - C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k},$$

where $\gamma_k = \min\left\{\frac{1}{32\tau b_k\theta_{max}^2}, \frac{1}{4\theta_{max}^2} - \frac{8\theta_{max}^2}{\tau b_k\theta_{min}^4}\right\}$ and $C_{3k}(\alpha_k, b_k) = \frac{1}{64b_k} + \frac{C_{1k}(\alpha_k,b_k)}{8} - \frac{4\theta_{max}^2}{b\theta_{min}^2}$.

Lemma 48. *Under Assumption 1-2 and 3b, with probability at least $1 - T\frac{1}{M_k^{1+\alpha_k}}$, the following inequality holds:*

$$\begin{aligned}&\left[g_k^*(\boldsymbol{\theta}_k^{(t)})\right]_2(\theta_{2k}^{(t)} - \theta_{2k}^*)\\ &\geq \frac{\gamma_k}{2}(\theta_{2k}^{(t)} - \theta_{2k}^*)^2 - (\theta_{max} - \theta_{min})^2 M_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1}\left(\frac{1}{2\theta_{min}^2} + \frac{C_{mat,k}(4b_k+3)}{2\theta_{min}^2(4b_k^2-6b_k-3)}\right),\end{aligned}$$

where $\gamma_k := \frac{1}{2M_k}\frac{M_k - C_{mat,k}M_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}}}{4\theta_{max}}$.

Lemma 49. *(Chen et al. 2020) Under Assumptions 1-2, for any $\phi > 0$, we have*

$$P\left(\sup_{\boldsymbol{\theta}} \frac{N_k}{s_i(N_k)}\left|[\nabla L_k(\boldsymbol{\theta}; D_k)]_i - [\nabla L_k^*(\boldsymbol{\theta})]_i\right| > C_{\boldsymbol{\theta}}\phi\right) \leq \delta(\phi), i = 1, 2,$$

where $\nabla L_k^*(\boldsymbol{\theta}) := \mathbb{E}\left(\nabla L_k(\boldsymbol{\theta}; D_k)|\boldsymbol{X}_k\right)$.

Furthermore, if assumption 3a holds and $s_i(N_k) = \tau \log N_k$, then for $N_k > C_{\boldsymbol{\theta}}, c_{\boldsymbol{\theta}} > 0$, we *have*

$$\delta(\phi) \leq \frac{C_{\boldsymbol{\theta}}}{N_k^{c_{\boldsymbol{\theta}}}} + C_{\boldsymbol{\theta}}(\log \phi)^4 \mathsf{exp}\{-c_{\boldsymbol{\theta}} \log N_k \min\{\phi^2, \phi\}\}.$$

If assumption 3a or 3b holds and $s_i(N_k) = N_k$, then

$$\delta(\phi) \leq C_{\boldsymbol{\theta}}(\log \phi)^4 \mathsf{exp}\{-c_{\boldsymbol{\theta}} N_k \min\{\phi^2, \phi\}\}.$$

C.7 Proof of Theorems

C.7.1 Detailed Notations

Let $\boldsymbol{\theta}_k^{(t)}$ be the model parameter maintained in the k^{th} device at the t^{th} step. Let $\mathcal{I}_E = \{cE \mid c = 1, 2, \ldots, R\}$ be the set of global aggregation steps. If $t+1 \in \mathcal{I}_E$, then the central server collects model parameters from active devices and aggregates all of those model parameters. Motivated by (Li et al. 2019b), we introduce an intermediate parameter $\boldsymbol{v}_k^{(t+1)} := \boldsymbol{\theta}_k^{(t)} - \eta^{(t)} g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)})$. It can be seen that $\boldsymbol{\theta}_k^{(t+1)} = \boldsymbol{v}_k^{(t+1)}$ if $t+1 \notin \mathcal{I}_E$ and $\boldsymbol{\theta}_k^{(t+1)} = \sum_{k=1}^K p_k \boldsymbol{v}_k^{(t+1)}$ otherwise. Let $\bar{\boldsymbol{v}}^{(t)} = \sum_{k=1}^K p_k \boldsymbol{v}_k^{(t)}$ and $\bar{\boldsymbol{\theta}}^{(t)} = \sum_{k=1}^K p_k \boldsymbol{\theta}_k^{(t)}$. The central server can only obtain $\bar{\boldsymbol{\theta}}^{(t)}$ when $t+1 \in \mathcal{I}_E$. The term $\bar{\boldsymbol{v}}^{(t)}$ is introduced for the purpose of proof and is inaccessible in practice. We further define $g^{(t)} = \sum_{k=1}^K p_k g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)})$.

C.7.2 Proof of Theorem 1

Under the scenario of full device participation, we have $\bar{\boldsymbol{\theta}}^{(t+1)} = \bar{\boldsymbol{v}}^{(t+1)}$ for all t. By definition of $\bar{\boldsymbol{v}}^{(t)}$, we have

$$\begin{aligned}\left\|\bar{\boldsymbol{v}}^{(t+1)} - \boldsymbol{\theta}^*\right\|_2^2 &= \left\|\bar{\boldsymbol{\theta}}^{(t)} - \eta^{(t)} g^{(t)} - \boldsymbol{\theta}^*\right\|_2^2 \\ &= \underbrace{\left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|_2^2}_{\text{A}} \underbrace{-2\eta^{(t)} \langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*, g^{(t)} \rangle}_{\text{B}} + \eta^{(t)2} \underbrace{\left\|g^{(t)}\right\|_2^2}_{\text{C}}.\end{aligned}$$

We can write B as

$$\begin{aligned}
\mathrm{B} &= -2\eta^{(t)}\langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*, g^{(t)}\rangle = -2\eta^{(t)}\langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*, \sum_{k=1}^{K} p_k g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)})\rangle \\
&= -2\eta^{(t)} \sum_{k=1}^{K} p_k \langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)})\rangle \\
&= -2\eta^{(t)} \sum_{k=1}^{K} p_k \langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}, g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)})\rangle - 2\eta^{(t)} \sum_{k=1}^{K} p_k \langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)})\rangle.
\end{aligned}$$

By Cauchy-Schwarz inequality and inequality of arithmetic and geometric means, we can simplify the first term in B as

$$\begin{aligned}
-2\langle \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}, g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)})\rangle &\le 2\frac{\sqrt{\eta^{(t)}}}{\sqrt{\eta^{(t)}}} \left\| \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)} \right\| \left\| g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) \right\| \\
&\le 2\frac{\frac{1}{\eta^{(t)}} \left\| \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)} \right\|^2 + \eta^{(t)} \left\| g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) \right\|^2}{2} \\
&\le \left(\frac{1}{\eta^{(t)}} \left\| \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)} \right\|_2^2 + \eta^{(t)} \left\| g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) \right\|_2^2 \right).
\end{aligned}$$

By Lemma 47, we can simplify the second term in B as

$$\begin{aligned}
&- 2\eta^{(t)}\langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)})\rangle = -2\eta^{(t)}\langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) + g_k^*(\boldsymbol{\theta}_k^{(t)}) - g_k^*(\boldsymbol{\theta}_k^{(t)})\rangle \\
&\le -2\eta^{(t)}\frac{\gamma_k}{2} \left\| \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^* \right\|_2^2 + 2\eta^{(t)} C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} - 2\eta^{(t)}\langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) - g_k^*(\boldsymbol{\theta}_k^{(t)})\rangle.
\end{aligned}$$

By Assumption 2,

$$\mathrm{C} = \left\| g^{(t)} \right\|_2^2 = \left\| \sum_{k=1}^{K} p_k g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) \right\|^2 \le \left(\sum_{k=1}^{K} \left\| p_k g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) \right\| \right)^2 \le \left(\sum_{k=1}^{K} p_k G \right)^2 = G^2.$$

Combining A, B and C together, we obtain

$$
\begin{aligned}
&\left\|\bar{\boldsymbol{v}}^{(t+1)} - \boldsymbol{\theta}^*\right\|_2^2 \\
&\leq \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|_2^2 + \eta^{(t)} \sum_{k=1}^{K} p_k \left(\frac{1}{\eta^{(t)}} \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}\right\|_2^2 + \eta^{(t)} G^2 \right) \\
&\quad - 2\eta^{(t)} \sum_{k=1}^{K} p_k \frac{\gamma_k}{2} \left\|\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*\right\|_2^2 + 2\eta^{(t)} \sum_{k=1}^{K} p_k C_{3k}(\alpha_k, b_k) \frac{\log M_k}{M_k} + \eta^{(t)2} G^2 \\
&\quad - 2\eta^{(t)} \sum_{k=1}^{K} p_k \langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) - g_k^*(\boldsymbol{\theta}_k^{(t)}) \rangle \\
&= \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|_2^2 + \underbrace{\sum_{k=1}^{K} p_k \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)}\right\|_2^2}_{\text{D}} + \eta^{(t)2} G^2 \\
&\quad - 2\eta^{(t)} \underbrace{\sum_{k=1}^{K} p_k \frac{\gamma_k}{2} \left\|\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*\right\|_2^2}_{\text{E}} + 2\eta^{(t)} \sum_{k=1}^{K} p_k C_{3k}(\alpha_k, b_k) \frac{\log M_k}{M_k} + \eta^{(t)2} G^2 \\
&\quad - 2\eta^{(t)} \sum_{k=1}^{K} p_k \langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) - g_k^*(\boldsymbol{\theta}_k^{(t)}) \rangle.
\end{aligned}
$$

Since the aggregation step happens each E steps, for any $t \geq 0$, there exists a $t_0 \leq t$ such that $t - t_0 \leq E - 1$ and $\boldsymbol{\theta}_k^{(t_0)} = \bar{\boldsymbol{\theta}}^{(t_0)}$ for all $k \in [K]$. Since $\eta^{(t)}$ is non-increasing, for all $t - t_0 \leq E - 1$, we can simplify D as

$$\begin{aligned}
\mathrm{D} = \sum_{k=1}^{K} p_k \left\| \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}_k^{(t)} \right\|_2^2 &= \sum_{k=1}^{K} p_k \left\| (\boldsymbol{\theta}_k^{(t)} - \bar{\boldsymbol{\theta}}^{(t_0)}) - (\bar{\boldsymbol{\theta}}^{(t)} - \bar{\boldsymbol{\theta}}^{(t_0)}) \right\|_2^2 \\
&\leq \sum_{k=1}^{K} p_k \left\| \boldsymbol{\theta}_k^{(t)} - \bar{\boldsymbol{\theta}}^{(t_0)} \right\|_2^2 + \underbrace{\sum_{k=1}^{K} p_k \left\| \bar{\boldsymbol{\theta}}^{(t)} - \bar{\boldsymbol{\theta}}^{(t_0)} \right\|_2^2}_{\sum p_k = 1} \\
&= \sum_{k=1}^{K} p_k \left\| \sum_{t=t_0}^{t-1} \eta^{(t)} g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) \right\|_2^2 + \left\| \sum_{k=1}^{K} p_k \sum_{t=t_0}^{t-1} \eta^{(t)} g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) \right\|_2^2 \\
&\leq \sum_{k=1}^{K} p_k (t - t_0) \sum_{t=t_0}^{t-1} \eta^{(t)2} \left\| g(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) \right\|_2^2 + \sum_{k=1}^{K} p_k \left\| \sum_{t=t_0}^{t-1} \eta^{(t)} g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) \right\|_2^2 \\
&\leq 2 \sum_{k=1}^{K} p_k (E-1) \sum_{t=t_0}^{t-1} \eta^{(t)2} \left\| g(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) \right\|_2^2 \leq 2 \sum_{k=1}^{K} p_k (E-1) \sum_{t=t_0}^{t-1} \eta^{(t_0)2} G^2 \\
&= 2 \sum_{k=1}^{K} p_k (E-1)^2 \eta^{(t_0)2} G^2.
\end{aligned}$$

Without loss of generality, assume $\eta^{(t_0)} \leq 2\eta^{(t)}$ since the learning rate is decreasing. Therefore, $\mathrm{D} \leq 8(E-1)^2 \eta^{(t)2} G^2$. To simplify E, we have

$$\begin{aligned}
\mathrm{E} = \sum_{k=1}^{K} p_k \frac{\gamma_k}{2} \left\| \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^* \right\|_2^2 &\geq \min_k \gamma_k \frac{1}{2} \sum_{k=1}^{K} p_k \left\| \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^* \right\|_2^2 \\
&\geq \min_k \gamma_k \frac{1}{2} \left\| \sum_{k=1}^{K} p_k (\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*) \right\|_2^2 = \min_k \gamma_k \frac{1}{2} \left\| \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^* \right\|_2^2,
\end{aligned}$$

using Jensen's inequality.

Therefore, we obtain

$$
\begin{aligned}
&\left\|\bar{\boldsymbol{v}}^{(t+1)}-\boldsymbol{\theta}^*\right\|_2^2 \\
&\leq \left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|_2^2 + 8(E-1)^2\eta^{(t)2}G^2 + \eta^{(t)2}G^2 \\
&\quad - 2\eta^{(t)}\min_k \gamma_k \frac{1}{2}\left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|_2^2 + 2\eta^{(t)}\sum_{k=1}^K p_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} + \eta^{(t)2}G^2 \\
&\quad - 2\eta^{(t)}\sum_{k=1}^K p_k\langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(t)};\xi_k^{(t)}) - g_k^*(\boldsymbol{\theta}_k^{(t)})\rangle \\
&= \left(1 - 2\eta^{(t)}\min_k \gamma_k \frac{1}{2}\right)\left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|_2^2 + (8(E-1)^2\eta^{(t)2} + 2\eta^{(t)2})G^2 \\
&\quad + 2\eta^{(t)}\sum_{k=1}^K p_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\
&\quad - 2\eta^{(t)}\sum_{k=1}^K p_k\langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(t)};\xi_k^{(t)}) - g_k^*(\boldsymbol{\theta}_k^{(t)})\rangle \\
&\leq \left(1 - 2\eta^{(t)}\min_k \gamma_k \frac{1}{2}\right)\left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|_2^2 + (8(E-1)^2\eta^{(t)2} + 2\eta^{(t)2})G^2 \\
&\quad + 2\eta^{(t)}\max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\
&\quad - 2\eta^{(t)}\sum_{k=1}^K p_k\langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(t)};\xi_k^{(t)}) - g_k^*(\boldsymbol{\theta}_k^{(t)})\rangle \\
&= \left(1 - 2\eta^{(t)}\min_k \gamma_k \frac{1}{2}\right)\left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|_2^2 + \left(8(E-1)^2 + 2\right)\eta^{(t)2}G^2 \\
&\quad + 2\eta^{(t)}\Bigg(\max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\
&\quad - \sum_{k=1}^K p_k\langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(t)};\xi_k^{(t)}) - g_k^*(\boldsymbol{\theta}_k^{(t)})\rangle\Bigg).
\end{aligned}
$$

Since $\frac{3}{2\min_k \gamma_k} \leq \beta_1 \leq \frac{2}{\min_k \gamma_k}$ and $\eta^{(t)} = \frac{\beta_1}{t}$ for all $t \geq 1$. Here we set $\eta^{(0)} = \beta_1$. We will show

$$
\begin{aligned}
\left\|\bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^*\right\|_2^2 &\leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+1} \\
&\quad + \sum_{u=0}^{t-1}\left(2\eta^{(u+1)}\prod_{v=u+2}^{t}(1-\eta^{(v)}\min_k \gamma_k)\right)\Bigg(\max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\
&\quad - \sum_{k=1}^K p_k\langle \boldsymbol{\theta}_k^{(u)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(u)};\xi_k^{(u)}) - g_k^*(\boldsymbol{\theta}_k^{(u)})\rangle\Bigg)
\end{aligned}
$$

by induction. When $t = 1$, we have

$$\begin{aligned}\left\|\bar{\boldsymbol{\theta}}^{(1)} - \boldsymbol{\theta}^*\right\|_2^2 \le{}& \left(8(E-1)^2+2\right)\beta_1^2 G^2 + 2\beta_1\Bigg(\max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\ &- \sum_{k=1}^{K} p_k \langle \boldsymbol{\theta}_k^{(0)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(0)}; \xi_k^{(0)}) - g_k^*(\boldsymbol{\theta}_k^{(0)})\rangle\Bigg)\end{aligned}$$

since $\left(1 - 2\eta^{(0)} \min_k \gamma_k \frac{1}{2}\right) < 0$. Assume the inequality holds for $t = l \ge 1$, then we have

$$\begin{aligned}&\left\|\bar{\boldsymbol{\theta}}^{(l+1)} - \boldsymbol{\theta}^*\right\|_2^2 \\ &\le \left(1 - 2\eta^{(l)} \min_k \gamma_k \frac{1}{2}\right)\Bigg\{\frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{l+1} \\ &\quad + \sum_{u=0}^{l-1} 2\eta^{(u+1)} \prod_{v=u+2}^{l}(1 - \eta^{(v)}\min_k \gamma_k)\Bigg(\max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\ &\quad - \sum_{k=1}^{K} p_k \langle \boldsymbol{\theta}_k^{(u)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(u)}; \xi_k^{(u)}) - g_k^*(\boldsymbol{\theta}_k^{(u)})\rangle\Bigg)\Bigg\} \\ &\quad + \left(8(E-1)^2+2\right)\eta^{(l)2}G^2 \\ &\quad + 2\eta^{(l)}\Bigg(\max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\ &\quad - \sum_{k=1}^{K} p_k \langle \boldsymbol{\theta}_k^{(l)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(l)}; \xi_k^{(l)}) - g_k^*(\boldsymbol{\theta}_k^{(l)})\rangle\Bigg) \\ &\le \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{l+2} \\ &\quad + \sum_{u=0}^{l} 2\eta^{(u+1)} \prod_{v=u+2}^{l}(1 - \eta^{(v)}\min_k \gamma_k)\Bigg(\max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\ &\quad - \sum_{k=1}^{K} p_k \langle \boldsymbol{\theta}_k^{(u)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(u)}; \xi_k^{(u)}) - g_k^*(\boldsymbol{\theta}_k^{(u)})\rangle\Bigg).\end{aligned}$$

To derive the above inequality, we can first show that

$$\begin{aligned}&\left(1 - 2\eta^{(l)} \min_k \gamma_k \frac{1}{2}\right)\frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{l+1} + \left(8(E-1)^2+2\right)\eta^{(l)2}G^2 \\ &\le \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{l+2}\end{aligned}$$

as long as $\beta_1 \ge \frac{3l+1}{2l+2}\frac{1}{\min_k \gamma_k}$. This is true since the right-hand side is always less or equal to $\frac{3}{2\min_k \gamma_k}$. The remaining part in the above inequality is apparent since $1 - 2\eta^{(l)} \min_k \gamma_k \frac{1}{2} \le 1$. Thus, the proof of the induction step is complete. Using this fact, it can be shown that

$$
\begin{aligned}
&\left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \boldsymbol{\theta}^*\right\|_2^2 \\
&\leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+2} \\
&\quad + \sum_{u=0}^{t} 2\eta^{(u+1)} \prod_{v=u+2}^{t}(1-\eta^{(v)}\min_k \gamma_k)\Bigg(\max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\
&\quad - \sum_{k=1}^{K} p_k \langle \boldsymbol{\theta}_k^{(u)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(u)};\xi_k^{(u)}) - g_k^*(\boldsymbol{\theta}_k^{(u)})\rangle \Bigg) \\
&\leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+2} \\
&\quad + \sum_{u=0}^{t} \left(2\eta^{(u+1)} \prod_{v=u+2}^{t}(1-\eta^{(v)}\min_k \gamma_k) \right) \left(\max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \right) \\
&\quad - \sum_{u=0}^{t} \left(2\eta^{(u+1)} \prod_{v=u+2}^{t}(1-\eta^{(v)}\min_k \gamma_k) \right) \left(\sum_{k=1}^{K} p_k \langle \boldsymbol{\theta}_k^{(u)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(u)};\xi_k^{(u)}) - g_k^*(\boldsymbol{\theta}_k^{(u)})\rangle \right) \\
&\leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+2} \\
&\quad + \sum_{u=0}^{t} \frac{2\beta_1}{t+1} \left(\max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \right) \\
&\quad + \sum_{u=0}^{t} \frac{2\beta_1}{t+1} \left(\sum_{k=1}^{K} p_k \left\| \boldsymbol{\theta}_k^{(u)} - \boldsymbol{\theta}^* \right\|_2 \left\| g_k(\boldsymbol{\theta}_k^{(u)};\xi_k^{(u)}) - g_k^*(\boldsymbol{\theta}_k^{(u)}) \right\|_2 \right) \\
&\leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+2} \\
&\quad + \left(\max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \right) \\
&\quad + \sum_{u=0}^{t} \frac{2\beta_1}{t+1} \left(\sum_{k=1}^{K} p_k \left\| \boldsymbol{\theta}_k^{(u)} - \boldsymbol{\theta}^* \right\|_2 \left\| g_k(\boldsymbol{\theta}_k^{(u)};\xi_k^{(u)}) - g_k^*(\boldsymbol{\theta}_k^{(u)}) \right\|_2 \right) \\
&\leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+2} \\
&\quad + \max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\
&\quad + \frac{2\beta_1}{t+1} \sum_{u=0}^{t} \left(\sum_{k=1}^{K} \sqrt{2} p_k (\theta_{max} - \theta_{min}) \left\| g_k(\boldsymbol{\theta}_k^{(u)};\xi_k^{(u)}) - g_k^*(\boldsymbol{\theta}_k^{(u)}) \right\|_2 \right) \\
&\leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+2} \\
&\quad + 2\beta_1 \max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\
&\quad + 2\beta_1 \max_{0 \leq u \leq t} \left(\sum_{k=1}^{K} \sqrt{2} p_k (\theta_{max} - \theta_{min}) \left\| g_k(\boldsymbol{\theta}_k^{(u)};\xi_k^{(u)}) - g_k^*(\boldsymbol{\theta}_k^{(u)}) \right\|_2 \right)
\end{aligned}
$$

In the third inequality, we use the Cauchy–Schwarz inequality and the fact that $2\eta^{(u+1)} \prod_{v=u+2}^{t}(1 - \eta^{(v)} \min_k \gamma_k) \leq 2\frac{\beta_1}{u+1} \prod_{v=u+2}^{t}(1 - \frac{3}{2v}) \leq \frac{2\beta_1}{t+1}$.

Let $\phi_k = (\log M_k)^{\epsilon_k - \frac{1}{2}}$. By Lemma 49 and using a union bound over u, with probability at least $1 - C_{\boldsymbol{\theta}}(T+1)\mathsf{exp}(-c_{\boldsymbol{\theta}}(\log M_k)^{2\epsilon_k})$, we have

$$\max_{0\leq u\leq t} \left\| g_k(\boldsymbol{\theta}_k^{(u)}; \xi_k^{(u)}) - g_k^*(\boldsymbol{\theta}_k^{(u)}) \right\|_2 \leq C_{\boldsymbol{\theta}}(\log M_k)^{\epsilon_k - \frac{1}{2}}.$$

Therefore,

$$\begin{aligned}
&\left\| \bar{\boldsymbol{\theta}}^{(t+1)} - \boldsymbol{\theta}^* \right\|_2^2 \\
&\leq \frac{2\beta_1^2 \left(8(E-1)^2 + 2\right) G^2}{t+2} + 2\beta_1 \max_k C_{3k}(\alpha_k, b_k) \frac{\log M_k}{M_k} \\
&\quad + 2\beta_1 \max_{0\leq u\leq t} \sum_{k=1}^{K} \sqrt{2} p_k (\theta_{max} - \theta_{min}) C_{\boldsymbol{\theta}} (\log M_k)^{\epsilon_k - \frac{1}{2}} \\
&= \frac{2\beta_1^2 \left(8(E-1)^2 + 2\right) G^2}{t+2} + 2\beta_1 \max_k C_{3k}(\alpha_k, b_k) \frac{\log M_k}{M_k} \\
&\quad + 2\sqrt{2}\beta_1 (\theta_{max} - \theta_{min}) C_{\boldsymbol{\theta}} \sum_{k=1}^{K} p_k (\log M_k)^{\epsilon_k - \frac{1}{2}}.
\end{aligned}$$

Using the same proof technique, we can also derive the same bound on $\left\| \bar{\theta}_2^{(t+1)} - \theta_2^* \right\|_2^2$. Let $\phi_k = M_k^{\epsilon_k - \frac{1}{2}}$. By Lemma 49 and using a union bound over u, with probability at least $1 - C_{\boldsymbol{\theta}}(t+1)(\log(M_k^{\epsilon_k - \frac{1}{2}}))^4 \mathsf{exp}\{-c_{\boldsymbol{\theta}} M_k^{2\epsilon_k}\}$,

$$\max_{0\leq u\leq t} \left\| g_k(\boldsymbol{\theta}_k^{(u)}; \xi_k^{(u)}) - g_k^*(\boldsymbol{\theta}_k^{(u)}) \right\|_2 \leq C_{\boldsymbol{\theta}} M_k^{\epsilon_k - \frac{1}{2}}.$$

Therefore,

$$\begin{aligned}
&\left\| \bar{\theta}_2^{(t+1)} - \theta_2^* \right\|_2^2 \\
&\leq \frac{2\beta_1^2 \left(8(E-1)^2 + 2\right) G^2}{t+2} + 2\beta_1 \max_k C_{3k}(\alpha_k, b_k) \frac{\log M_k}{M_k} \\
&\quad + 2\beta_1 \max_{0\leq u\leq t} \sum_{k=1}^{K} \sqrt{2} p_k (\theta_{max} - \theta_{min}) C_{\boldsymbol{\theta}} M_k^{\epsilon_k - \frac{1}{2}} \\
&= \frac{2\beta_1^2 \left(8(E-1)^2 + 2\right) G^2}{t+2} + 2\beta_1 \max_k C_{3k}(\alpha_k, b_k) \frac{\log M_k}{M_k} \\
&\quad + 2\sqrt{2}\beta_1 (\theta_{max} - \theta_{min}) C_{\boldsymbol{\theta}} \sum_{k=1}^{K} p_k M_k^{\epsilon_k - \frac{1}{2}}.
\end{aligned}$$

C.7.3 Proof of Theorem 2

We slightly modify the definition of $\boldsymbol{\theta}_k^{(t+1)}$ such that $\boldsymbol{\theta}_k^{(t+1)} = \frac{1}{|\mathcal{S}|}\sum_{k\in\mathcal{S}} \boldsymbol{v}_k^{(t+1)}$ if $t+1 \in \mathcal{I}_E$. Under the scenario of asynchronous update, it can be seen that $\bar{\boldsymbol{\theta}}^{(t+1)} \neq \bar{\boldsymbol{v}}^{(t+1)}$. Therefore, we want to establish a bound on the difference $\left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{v}}^{(t+1)}\right\|_2^2$. We have

$$\begin{aligned}\left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \boldsymbol{\theta}^*\right\|_2^2 &= \left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{v}}^{(t+1)} + \bar{\boldsymbol{v}}^{(t+1)} - \boldsymbol{\theta}^*\right\|_2^2 \\ &= \underbrace{\left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{v}}^{(t+1)}\right\|_2^2}_{\text{A}} + \underbrace{\left\|\bar{\boldsymbol{v}}^{(t+1)} - \boldsymbol{\theta}^*\right\|_2^2}_{\text{B}} + \underbrace{2\langle\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{v}}^{(t+1)}, \bar{\boldsymbol{v}}^{(t+1)} - \boldsymbol{\theta}^*\rangle}_{\text{C}}.\end{aligned}$$

We can show

$$\mathbb{E}_{\mathcal{S}}\left\{\bar{\boldsymbol{\theta}}^{(t+1)}\right\} = \mathbb{E}_{\mathcal{S}}\left\{\frac{1}{|\mathcal{S}|}\sum_{k\in\mathcal{S}} \boldsymbol{v}_k^{(t+1)}\right\} = \frac{1}{|\mathcal{S}|}\sum_{k\in\mathcal{S}} \mathbb{E}_{\mathcal{S}}\left\{\boldsymbol{v}_k^{(t+1)}\right\} = \mathbb{E}_{\mathcal{S}}\left\{\boldsymbol{v}_1^{(t+1)}\right\} = \sum_{k=1}^{K} p_k \boldsymbol{v}_k^{(t+1)} = \bar{\boldsymbol{v}}^{(t+1)}$$

since the sampling distribution is identical. Therefore, $\mathbb{E}_{\mathcal{S}}[\text{C}] = 0$.

For part A, we have

$$\begin{aligned}&\mathbb{E}_{\mathcal{S}}\left\{\left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{v}}^{(t+1)}\right\|_2^2\right\} \\ &= \mathbb{E}_{\mathcal{S}}\left\{\frac{1}{|\mathcal{S}|^2}\sum_{k\in\mathcal{S}} \left\|\boldsymbol{v}_k^{(t+1)} - \bar{\boldsymbol{v}}^{(t+1)}\right\|_2^2\right\} = \frac{1}{|\mathcal{S}|}\sum_{k=1}^{K} p_k \left\|\boldsymbol{v}_k^{(t+1)} - \bar{\boldsymbol{v}}^{(t+1)}\right\|_2^2.\end{aligned}$$

The first equality uses the fact that $\boldsymbol{v}_k^{(t+1)}$ is independent of each other and is an unbiased estimator of $\bar{\boldsymbol{v}}^{(t+1)}$. Therefore, we have

$$\begin{aligned}
\sum_{k=1}^{K} p_k \left\| \boldsymbol{v}_k^{(t+1)} - \bar{\boldsymbol{v}}^{(t+1)} \right\|_2^2 &= \sum_{k=1}^{K} p_k \left\| \boldsymbol{v}_k^{(t+1)} - \bar{\boldsymbol{\theta}}^{(t_0)} - (\bar{\boldsymbol{\theta}}^{(t_0)} - \bar{\boldsymbol{v}}^{(t+1)}) \right\|_2^2 \\
&\leq \sum_{k=1}^{K} p_k \left\| \boldsymbol{v}_k^{(t+1)} - \bar{\boldsymbol{\theta}}^{(t_0)} \right\|_2^2 \\
&\leq \sum_{k=1}^{K} p_k \left\| \boldsymbol{v}_k^{(t+1)} - \boldsymbol{\theta}_k^{(t_0)} \right\|_2^2 \\
&= \sum_{k=1}^{K} p_k \left\| \sum_{i=t_0}^{t} \eta^{(i)} g_k(\boldsymbol{\theta}_k^{(i)}; \xi_k^{(i)}) \right\|_2^2 \\
&\leq \sum_{k=1}^{K} p_k \sum_{i=t_0}^{t} E \left\| \eta^{(i)} g_k(\boldsymbol{\theta}_k^{(i)}; \xi_k^{(i)}) \right\|_2^2 \\
&\leq E^2 \eta^{(t_0)2} G^2 \leq 4E^2 \eta^{(t)2} G^2
\end{aligned}$$

where $t_0 = t - E + 1$ is the iteration where communication happens. Therefore,

$$\mathbb{E}_{\mathcal{S}} \left\{ \left\| \bar{\boldsymbol{\theta}}^{(t+1)} - \bar{\boldsymbol{v}}^{(t+1)} \right\|_2^2 \right\} \leq \frac{4E^2 \eta^{(t)2} G^2}{|\mathcal{S}|}.$$

For part B, we can follow the exact proof in Theorem 1 to get an upper bound after taking expectation with respect to $\mathcal{S}$. In a nutshell, we can obtain

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S}} \left\{ \left\| \bar{\boldsymbol{\theta}}^{(t+1)} - \boldsymbol{\theta}^* \right\|_2^2 \right\} \\
&\leq \frac{4E^2 \eta^{(t)2} G^2}{|\mathcal{S}|} + \left(1 - 2\eta^{(t)} \min_k \gamma_k \frac{1}{2}\right) \left\| \bar{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^* \right\|_2^2 + \left(8(E-1)^2 + 2\right) \eta^{(t)2} G^2 \\
&\quad + 2\eta^{(t)} \Bigg(\max_k C_{3k}(\alpha_k, b_k) \frac{\log M_k}{M_k} \\
&\quad - \sum_{k=1}^{K} p_k \langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k(\boldsymbol{\theta}_k^{(t)}; \xi_k^{(t)}) - g_k^*(\boldsymbol{\theta}_k^{(t)}) \rangle \Bigg).
\end{aligned}$$

Following the induction proof in Theorem 1, we have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S}}\left\{\left\|\bar{\boldsymbol{\theta}}^{(t+1)} - \boldsymbol{\theta}^*\right\|_2^2\right\} \\
&\leq \frac{2\beta_1^2\left(\frac{1}{|\mathcal{S}|}4E^2 + 8(E-1)^2 + 2\right)G^2}{t+2} + 2\beta_1 \max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\
&\quad + 2\sqrt{2}\beta_1(\theta_{max} - \theta_{min})C_{\boldsymbol{\theta}}\sum_{k=1}^{K} p_k M_k^{\epsilon_k - \frac{1}{2}}.
\end{aligned}$$

C.7.4 Proof of Theorem 3

Convergence of Parameter Iterates

Define $C_{4k} := (\theta_{max} - \theta_{min})^2\left(\frac{1}{2\theta_{min}^2} + \frac{C_{mat,k}(4b_k+3)}{2\theta_{min}^2(4b_k^2-6b_k-3)}\right)$. Following the same proof strategy in Theorem 1 and using Lemma 46 and 48, we can show that

$$\begin{aligned}
&\left\|\bar{\theta}_2^{(t+1)} - \theta_2^*\right\|_2^2 \\
&\leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+1} + \sum_{u=0}^{t} 2\eta^{(u+1)}\prod_{v=u+2}^{t}(1-\eta^{(v)}\min_k \gamma_k)\Bigg(\max_k C_{4k} M_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1} - \\
&\quad \sum_{k=1}^{K} p_k \langle \theta_{2k}^{(u)} - \theta_2^*, [g_k(\boldsymbol{\theta}_k^{(u)}; \xi_k^{(u)})]_2 - [g_k^*(\boldsymbol{\theta}_k^{(u)})]_2\rangle\Bigg) \\
&\leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+1} + 2\beta_1\max_k C_{4k} M_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1} \\
&\quad + 2\beta_1 \max_{0\leq u\leq t}\left(\sum_{k=1}^{K}\sqrt{2}p_k(\theta_{max} - \theta_{min})\left\|[g_k(\boldsymbol{\theta}_k^{(u)}; \xi_k^{(u)})]_2 - [g_k^*(\boldsymbol{\theta}_k^{(u)})]_2\right\|_2\right).
\end{aligned}$$

Let $\phi_k = M_k^{\epsilon_k - \frac{1}{2}}$. By Lemma 49, for any $0 < \alpha_k < \frac{8b_k^2 - 12b_k - 6}{4b_k+3}$, $\epsilon_k < \frac{1}{2}$, with probability at least $1 - C_{\boldsymbol{\theta}}(t+1)(\log(M_k^{\epsilon_k - \frac{1}{2}}))^4 \mathsf{exp}\{-c_{\boldsymbol{\theta}} M_k^{2\epsilon_k}\}$, we have

$$\max_{0\leq u\leq t}\left\|[g_k(\boldsymbol{\theta}_k^{(u)}; \xi_k^{(u)})]_2 - [g_k^*(\boldsymbol{\theta}_k^{(u)})]_2\right\|_2 \leq C_{\boldsymbol{\theta}} M_k^{\epsilon_k - \frac{1}{2}}.$$

Therefore,

$$\begin{aligned}
&\left\|\bar{\theta}_2^{(t+1)} - \theta_2^*\right\|_2^2 \\
&\leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+1} + 2\beta_1 \max_k C_{4k} M_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1} \\
&\quad + 2\beta_1\left(\sum_{k=1}^{K}\sqrt{2}p_k(\theta_{max}-\theta_{min})C_{\boldsymbol{\theta}}M_k^{\epsilon_k-\frac{1}{2}}\right) \\
&= \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+1} + \mathcal{O}\left(\max_k M_k^{-\frac{8b_k^2-12b_k-6-3\alpha_k-4\alpha_k b_k}{8b_k^2-4b_k}}\right) + \mathcal{O}\left(\sum_{k=1}^{K}p_k M_k^{\epsilon_k-\frac{1}{2}}\right).
\end{aligned}$$

The partial device participation proof is similar to Theorem 2. Again, using Lemma 46 and 48, we can show that

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S}}\left\{\left\|\bar{\theta}_2^{(t+1)} - \theta_2^*\right\|_2^2\right\} \\
&\leq \frac{2\beta_1^2\left(\frac{4E^2}{|\mathcal{S}|}+8(E-1)^2+2\right)G^2}{t+1} \\
&\quad + 2\beta_1 \max_k C_{4k} M_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1} + 2\beta_1\left(\sum_{k=1}^{K}\sqrt{2}p_k(\theta_{max}-\theta_{min})C_{\boldsymbol{\theta}}M_k^{\epsilon_k-\frac{1}{2}}\right) \\
&= \frac{2\beta_1^2\left(\frac{4E^2}{|\mathcal{S}|}+8(E-1)^2+2\right)G^2}{t+1} + \mathcal{O}\left(\max_k M_k^{-\frac{8b_k^2-12b_k-6-3\alpha_k-4\alpha_k b_k}{8b_k^2-4b_k}}\right) + \mathcal{O}\left(\sum_{k=1}^{K}p_k M_k^{\epsilon_k-\frac{1}{2}}\right).
\end{aligned}$$

Convergence of Full Gradient

We follow the same proof strategy in Theorem 4. We defer this proof to the subsection after it.

C.7.5 Proof of Theorem 4

Proof. Our final goal is to bound the squared norm of full gradient

$$\left\|\nabla L(\bar{\boldsymbol{\theta}})\right\|_2^2 = \left\|\sum_{k=1}^{K} p_k \nabla L_k(\bar{\boldsymbol{\theta}}; D_k)\right\|_2^2.$$

We define a conditional expectation of $\nabla L_k(\bar{\boldsymbol{\theta}}^{(t)}; D_k)$ as

$$\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)}) := \mathbb{E}\left(\nabla L_k(\bar{\boldsymbol{\theta}}^{(t)}; D_k)|\boldsymbol{X}_k\right).$$

By the definition of $\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})$, for $i \in \{1,2\}$, we have

$$\begin{aligned}
&\left[\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_i \\
&= \frac{1}{2N_k} \operatorname{Tr}\left[\boldsymbol{K}_{N_k}(\bar{\boldsymbol{\theta}}^{(t)})^{-1}\left(\boldsymbol{I}_{N_k} - \boldsymbol{K}_{N_k}(\boldsymbol{\theta}_k^*)\boldsymbol{K}_{N_k}(\bar{\boldsymbol{\theta}}^{(t)})^{-1}\right)\frac{\partial \boldsymbol{K}_{N_k}(\bar{\boldsymbol{\theta}}^{(t)})}{\partial \bar{\theta}_i^{(t)}}\right] \\
&= \frac{1}{2N_k} \operatorname{Tr}\left[\boldsymbol{K}_{N_k}(\bar{\boldsymbol{\theta}}^{(t)})^{-1}\left(\boldsymbol{K}_{N_k}(\bar{\boldsymbol{\theta}}^{(t)}) - \boldsymbol{K}_{N_k}(\boldsymbol{\theta}_k^*)\right)\boldsymbol{K}_{N_k}(\bar{\boldsymbol{\theta}}^{(t)})^{-1}\frac{\partial \boldsymbol{K}_{N_k}(\bar{\boldsymbol{\theta}}^{(t)})}{\partial \bar{\theta}_i^{(t)}}\right].
\end{aligned}$$

where $\boldsymbol{\theta}_k^* := (\theta_{1k}^*, \theta_{2k}^*)$ is the set of optimal model parameters for device k and $\boldsymbol{K}_{N_k}(\boldsymbol{\theta}_k) = \theta_{1k}\boldsymbol{K}_{f,N_k} + \theta_{2k}\boldsymbol{I}_{N_k}$, where θ_{1k}, θ_{2k} are device-specific model parameters and $\boldsymbol{I}_{N_k}$ is an identity matrix of size N_k. Therefore, we obtain

$$\begin{aligned}
&\left[\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_i \\
&= \frac{1}{2N_k} \operatorname{Tr}\left[\boldsymbol{K}_{N_k}(\bar{\boldsymbol{\theta}}^{(t)})^{-1}\left((\bar{\theta}_1^{(t)} - \theta_{1k}^*)\boldsymbol{K}_{f,N_k} + (\bar{\theta}_2^{(t)} - \theta_{2k}^*)\boldsymbol{I}_{N_k}\right)\boldsymbol{K}_{N_k}(\bar{\boldsymbol{\theta}}^{(t)})^{-1}\frac{\partial \boldsymbol{K}_{N_k}(\bar{\boldsymbol{\theta}}^{(t)})}{\partial \bar{\theta}_i^{(t)}}\right],
\end{aligned}$$

where $\bar{\theta}_i^{(t)} = \sum_{k=1}^K p_k \theta_{ik}^{(t)}, i = 1, 2$.

By Eigendecomposition, we can write $\boldsymbol{K}_{f,N_k} = \boldsymbol{Q}_{N_k}\boldsymbol{\Lambda}_{N_k}\boldsymbol{Q}_{N_k}^{-1}$ where $\boldsymbol{Q}_{N_k}$ contains eigenvectors of $\boldsymbol{K}_{f,N_k}$, $\boldsymbol{\Lambda}_{N_k} := \operatorname{diag}(\lambda_{11}, \lambda_{12}, \ldots, \lambda_{1N_k})$ is a diagonal matrix with eigenvalues of $\boldsymbol{K}_{f,N_k}$ and λ_{1j} is the j^{th} largest eigenvalue of $\boldsymbol{K}_{f,N_k}$. Here note that the values of $\lambda_{..}$ are different for each device k. For simplicity, we drop the notation k in the eigenvalues unless there is an ambiguity. When $i = 1$, we can simplify $\left[\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_i$ as

$$\begin{aligned}
&\left[\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_1 \\
&= \frac{1}{2N_k} \operatorname{Tr}\left[\boldsymbol{K}_{N_k}(\bar{\boldsymbol{\theta}}^{(t)})^{-1}\left((\bar{\theta}_1^{(t)} - \theta_{1k}^*)\boldsymbol{K}_{f,N_k} + (\bar{\theta}_1^{(t)} - \theta_{2k}^*)\boldsymbol{I}_{N_k}\right)\boldsymbol{K}_{N_k}(\bar{\boldsymbol{\theta}}^{(t)})^{-1}\boldsymbol{K}_{f,N_k}\right] \\
&= \frac{1}{2N_k}(\bar{\theta}_1^{(t)} - \theta_{1k}^*)\sum_{j=1}^{N_k}\frac{\lambda_{1j}^2}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} + \frac{1}{2N_k}(\bar{\theta}_2^{(t)} - \theta_{2k}^*)\sum_{j=1}^{N_k}\frac{\lambda_{2j}\lambda_{1j}}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2}.
\end{aligned}$$

where $\lambda_{2j} = 1$ is the j^{th} largest eigenvalue of $\boldsymbol{I}_{N_k}$. Similarly, it can be shown that, when $i = 2$,

$$\begin{aligned}
&\left[\nabla L^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_2 \\
&= \frac{1}{2N_k}(\bar{\theta}_1^{(t)} - \theta_{1k}^*)\sum_{j=1}^{N_k}\frac{\lambda_{1j}\lambda_{2j}}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} + \frac{1}{2N_k}(\bar{\theta}_2^{(t)} - \theta_{2k}^*)\sum_{j=1}^{N_k}\frac{\lambda_{2j}^2}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2}.
\end{aligned}$$

Our first goal is to bound eigenvalues of $\boldsymbol{K}_{f,N_k}$ using Lemma 44 and 45.

Part I: Bounding eigenvalues By Lemma 44 and 45, for any $0 < \epsilon_k, \alpha_k < 1$, $C_{1k}(\alpha, b) > 0$ and $N_k > C_{2k}(\epsilon_k, b_k)$, with probability at least $1 - \frac{3}{N_k^{\alpha_k}}$,

$$\frac{\epsilon_k \log N_k}{8b_k\theta_{max}^2} \leq \sum_{j=1}^{N_k} \frac{\lambda_{1j}^2}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} \leq \frac{4+2\alpha_k}{b_k\theta_{min}^2}\log N_k$$

$$\frac{N_k - C_{1k}(\alpha_k, b_k)\log N_k}{4\theta_{max}^2} \leq \sum_{j=1}^{N_k} \frac{\lambda_{2j}^2}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} \leq \frac{N_k}{\theta_{min}^2}$$

$$0 < \sum_{j=1}^{N_k} \frac{\lambda_{1j}\lambda_{2j}}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} \leq \frac{5+2\alpha_k}{7b_k\theta_{min}^2}\log N_k.$$

Therefore, we can show that, with probability at least $1 - \frac{3}{N_k^{\alpha_k}}$,

$$\begin{aligned}\left[\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_1 &= \frac{1}{2N_k}(\bar{\theta}_1^{(t)} - \theta_{1k}^*)\sum_{j=1}^{N_k} \frac{\lambda_{1j}^2}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} \\ &\quad + \frac{1}{2N_k}(\bar{\theta}_2^{(t)} - \theta_{2k}^*)\sum_{j=1}^{N_k} \frac{\lambda_{2j}\lambda_{1j}}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} \\ &\leq \frac{1}{2N_k}(\bar{\theta}_1^{(t)} - \theta_{1k}^*)\frac{4+2\alpha_k}{b_k\theta_{min}^2}\log N_k + \frac{1}{2N_k}\frac{5+2\alpha_k}{7b_k\theta_{min}^2}\log N_k \\ &\leq \frac{(\theta_{max} - \theta_{min})(33+16\alpha_k)}{14N_k b_k\theta_{min}^2}\log N_k = \frac{(\theta_{max} - \theta_{min})(33+16\alpha_k)}{14b_k\theta_{min}^2}\frac{\log N_k}{N_k},\end{aligned}$$

and

$$\begin{aligned}\left[\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_1 &= \frac{1}{2N_k}(\bar{\theta}_1^{(t)} - \theta_{1k}^*)\sum_{j=1}^{N_k} \frac{\lambda_{1j}^2}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} \\ &\quad + \frac{1}{2N_k}(\bar{\theta}_2^{(t)} - \theta_{2k}^*)\sum_{j=1}^{N} \frac{\lambda_{2j}\lambda_{1j}}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} \\ &\geq \frac{1}{2N_k}\frac{\epsilon_k \log N_k}{8b_k\theta_{max}^2} = \frac{\epsilon_k}{16b_k\theta_{max}^2}\frac{\log N_k}{N_k} > 0.\end{aligned}$$

Similarly, it can be shown that

$$
\begin{aligned}
&\left[\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_2 \\
&= \frac{1}{2N_k}(\bar{\theta}_1^{(t)} - \theta_{1k}^*) \sum_{j=1}^{N_k} \frac{\lambda_{1j}\lambda_{2j}}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} + \frac{1}{2N_k}(\bar{\theta}_2^{(t)} - \theta_{2k}^*) \sum_{j=1}^{N_k} \frac{\lambda_{2j}^2}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} \\
&\leq \frac{1}{2N_k}(\bar{\theta}_1^{(t)} - \theta_{1k}^*)\frac{5+2\alpha_k}{7b_k\theta_{min}^2}\log N_k + \frac{1}{2N_k}(\bar{\theta}_2^{(t)} - \theta_{2k}^*)\frac{N_k}{\theta_{min}^2} \\
&\leq \frac{(\theta_{max} - \theta_{min})(5+2\alpha_k)}{14b_k\theta_{min}^2}\frac{\log N_k}{N_k} + (\bar{\theta}_2^{(t)} - \theta_{2k}^*)\frac{1}{2\theta_{min}^2},
\end{aligned}
$$

and

$$
\begin{aligned}
&\left[\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_2 \\
&= \frac{1}{2N_k}(\bar{\theta}_1^{(t)} - \theta_{1k}^*) \sum_{j=1}^{N_k} \frac{\lambda_{1j}\lambda_{2j}}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} + \frac{1}{2N_k}(\bar{\theta}_2^{(t)} - \theta_{2k}^*) \sum_{j=1}^{N_k} \frac{\lambda_{2j}^2}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} \\
&\geq \frac{1}{2N_k}(\bar{\theta}_2^{(t)} - \theta_{2k}^*)\frac{N_k - C_{1k}(\alpha_k, b_k)\log N_k}{4\theta_{max}^2} \\
&\geq (\bar{\theta}_2^{(t)} - \theta_{2k}^*)\frac{1}{8\theta_{max}^2} - \frac{(\theta_{max} - \theta_{min})C_{1k}(\alpha_k, b_k)}{8\theta_{max}^2}\frac{\log N_k}{N_k}.
\end{aligned}
$$

By combining above inequalities, we obtain

$$
\begin{aligned}
&\left\|\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right\|_2^2 \\
&= \left(\left[\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_1^2 + \left[\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_2^2\right) \\
&\leq \Bigg\{ \left(\frac{(\theta_{max} - \theta_{min})(33+16\alpha_k)}{14b_k\theta_{min}^2}\frac{\log N_k}{N_k}\right)^2 \\
&\qquad + \left(\frac{(\theta_{max} - \theta_{min})(5+2\alpha_k)}{14b_k\theta_{min}^2}\frac{\log N_k}{N_k} + (\bar{\theta}_2^{(t)} - \theta_{2k}^*)\frac{1}{2\theta_{min}^2}\right)^2 \Bigg\}.
\end{aligned}
$$

Our next goal is therefore to study the behavior of $\bar{\theta}_2^{(t)} - \theta_{2k}^*$ during iteration and provide a bound on this parameter iterate.

Part II: Bounding parameter iterates We consider the full device participation scenario and the partial device participation scenario separately.

Under the full device participation scenario, following the same procedure in the proof of

Theorem 1, we can show that

$$\begin{aligned}
&\left\|\bar{\theta}_2^{(t+1)} - \bar{\theta}_{2k}^*\right\|_2^2 \leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+2} \\
&\qquad + 2\beta_1 \max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\
&\qquad + 2\beta_1 \max_{0\leq u\leq t}\left(\sum_{k=1}^{K}\sqrt{2}p_k(\theta_{max}-\theta_{min})\left\|g_k(\boldsymbol{\theta}_k^{(u)};\xi_k^{(u)}) - g_k^*(\boldsymbol{\theta}_k^{(u)})\right\|_2\right) \\
&\leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+2} + 2\beta_1 \max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} + 2\sqrt{2}\beta_1(\theta_{max}-\theta_{min})C_{\boldsymbol{\theta}}M_k^{\epsilon_k-\frac{1}{2}},
\end{aligned}$$

with probability at least $1 - C_{\boldsymbol{\theta}}(t+1)(\log(M_k^{\epsilon-\frac{1}{2}}))^4\mathsf{exp}\{-c_{\boldsymbol{\theta}}M_k^{2\epsilon_k}\}$.

Under the partial device participation scenario, following the same procedure in the proof of Theorem 2, we can show

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S}}\left\{\left\|\bar{\theta}^{(t+1)} - \bar{\theta}_{2k}^*\right\|_2^2\right\} \\
&\leq \frac{2\beta_1^2\left(\frac{1}{|\mathcal{S}|}4E^2 + 8(E-1)^2+2\right)G^2}{t+2} + 2\beta_1 \max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\
&\qquad + 2\sqrt{2}\beta_1(\theta_{max}-\theta_{min})C_{\boldsymbol{\theta}}M_k^{\epsilon_k-\frac{1}{2}}.
\end{aligned}$$

Part III: Bounding $\left[\nabla L(\bar{\boldsymbol{\theta}}^{(t)})\right]_i$ for $i = 1, 2$ and Proving convergence Finally, equipped with all aforementioned results, we are going to prove our convergence result.

From Part I, we know

$$\begin{aligned}
&\left\|\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right\|_2^2 \\
&\leq \Bigg\{\left(\frac{(\theta_{max}-\theta_{min})(33+16\alpha_k)}{14b_k\theta_{min}^2}\frac{\log N_k}{N_k}\right)^2 \\
&\qquad + \left(\frac{(\theta_{max}-\theta_{min})(5+2\alpha_k)}{14b_k\theta_{min}^2}\frac{\log N_k}{N_k} + (\bar{\theta}_2^{(t)} - \theta_{2k}^*)\frac{1}{2\theta_{min}^2}\right)^2\Bigg\} \\
&\leq w_{1k}^2\left(\frac{\log N_k}{N_k}\right)^2 + w_{2k}^2\left(\frac{\log N_k}{N_k}\right)^2 + 2w_{2k}\frac{\log N_k}{N_k}(\bar{\theta}_2^{(t)} - \theta_{2k}^*)\frac{1}{2\theta_{min}^2} + \left\|\bar{\theta}_2^{(t)} - \bar{\theta}_{2k}^*\right\|_2^2\frac{1}{4\theta_{min}^4} \\
&\leq (w_{1k}^2 + w_{2k}^2)\left(\frac{\log N_k}{N_k}\right)^2 + 2w_{2k}\frac{\log N_k}{N_k}(\theta_{max}-\theta_{min})\frac{1}{2\theta_{min}^2} \\
&+ \Bigg(\frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+1} + 2\beta_1 \max_k C_{3k}(\alpha_k, b_k)\frac{\log M_k}{M_k} \\
&\qquad + 2\sqrt{2}\beta_1(\theta_{max}-\theta_{min})C_{\boldsymbol{\theta}}M_k^{\epsilon_k-\frac{1}{2}}\Bigg)\frac{1}{4\theta_{min}^4},
\end{aligned}$$

where

$$\begin{aligned} w_{1k} &= \frac{(\theta_{max} - \theta_{min})(33 + 16\alpha_k)}{14 b_k \theta_{min}^2}, \\ w_{2k} &= \frac{(\theta_{max} - \theta_{min})(5 + 2\alpha_k)}{14 b_k \theta_{min}^2}. \end{aligned}$$

By Lemma 49, with probability at least $1 - C_{\boldsymbol{\theta}}(t+1)(\log(M_k^{\epsilon - \frac{1}{2}}))^4 \mathsf{exp}\{-c_{\boldsymbol{\theta}} M_k^{2\epsilon_k}\}$,

$$\begin{aligned} &\left\|\nabla L_k(\bar{\boldsymbol{\theta}}^{(t)})\right\|_2^2 \leq \left(C_{\boldsymbol{\theta}} M_k^{\epsilon_k - \frac{1}{2}}\right)^2 + \left\|\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right\|_2^2 + 2\left\|\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right\|_2 \left(C_{\boldsymbol{\theta}} M_k^{\epsilon_k - \frac{1}{2}}\right) \\ &\leq C_{\theta,k}^2 M_k^{2\epsilon_k - 1} + (w_{1k}^2 + w_{2k}^2)\left(\frac{\log N_k}{N_k}\right)^2 + 2 w_{2k} \frac{\log N_k}{N_k}(\theta_{max} - \theta_{min}) \frac{1}{2\theta_{min}^2} \\ &+ \left(\frac{2\beta_1^2\left(8(E-1)^2 + 2\right) G^2}{t+1} + 2\beta_1 \max_k C_{3k}(\alpha_k, b_k) \frac{\log M_k}{M_k}\right. \\ &\qquad \left. + 2\sqrt{2}\beta_1(\theta_{max} - \theta_{min}) C_{\boldsymbol{\theta}} M_k^{\epsilon_k - \frac{1}{2}}\right) \frac{1}{4\theta_{min}^4} \\ &+ 2\left\|\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right\|_2 \left(C_{\boldsymbol{\theta}} M_k^{\epsilon_k - \frac{1}{2}}\right). \end{aligned}$$

Therefore,

$$\begin{aligned} &\left\|\nabla L(\bar{\boldsymbol{\theta}})\right\|_2^2 = \left\|\sum_{k=1}^K p_k \nabla L_k(\bar{\boldsymbol{\theta}}; D_k)\right\|_2^2 \leq \sum_{k=1}^K p_k \left\|\nabla L_k(\bar{\boldsymbol{\theta}}; D_k)\right\|_2^2 \leq \max_k \left\|\nabla L_k(\bar{\boldsymbol{\theta}}; D_k)\right\|_2^2 \\ &\leq \max_k \left(\frac{2\beta_1^2\left(8(E-1)^2 + 2\right) G^2}{4\theta_{min}^4(t+1)} + \mathcal{O}\left(\frac{\log M_k}{M_k} + M_k^{\epsilon_k - \frac{1}{2}} + \frac{\log N_k}{N_k}\right)\right). \end{aligned}$$

Under the partial device participation scenario, we have

$$\left\|\nabla L(\bar{\boldsymbol{\theta}})\right\|_2^2 \leq \max_k \left(\frac{2\beta_1^2\left(\frac{1}{|\mathcal{S}|}4E^2 + 8(E-1)^2 + 2\right) G^2}{4\theta_{min}^4(t+1)} + \mathcal{O}\left(\frac{\log M_k}{M_k} + M_k^{\epsilon_k - \frac{1}{2}} + \frac{\log N_k}{N_k}\right)\right).$$

□

C.7.6 Missing Proof in Theorem 3

Following the same strategy in Theorem 4, we can show that

$$
\begin{aligned}
&\left\|\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right\|_2^2 \\
&= \left(\left[\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_1^2 + \left[\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right]_2^2\right) \\
&= \left(\frac{1}{2N_k}(\bar{\theta}_1^{(t)} - \theta_1^*)\sum_{j=1}^{N_k}\frac{\lambda_{1j}^2}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} + \frac{1}{2N_k}(\bar{\theta}_2^{(t)} - \theta_2^*)\sum_{j=1}^{N_k}\frac{\lambda_{2j}\lambda_{1j}}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2}\right)^2 \\
&+ \left(\frac{1}{2N_k}(\bar{\theta}_1^{(t)} - \theta_1^*)\sum_{j=1}^{N_k}\frac{\lambda_{1j}\lambda_{2j}}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2} + \frac{1}{2N_k}(\bar{\theta}_2^{(t)} - \theta_2^*)\sum_{j=1}^{N_k}\frac{\lambda_{2j}^2}{\left(\bar{\theta}_1^{(t)}\lambda_{1j} + \bar{\theta}_2^{(t)}\lambda_{2j}\right)^2}\right)^2.
\end{aligned}
$$

By Lemma 46, we have

$$\begin{aligned}
&\left\|\nabla L_k^*(\bar{\boldsymbol{\theta}}^{(t)})\right\|_2^2 \\
&\leq \Bigg(\frac{1}{2N_k}(\bar{\theta}_1^{(t)}-\theta_1^*)N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}}\left(\frac{1}{\theta_{min}^2}+\frac{C_{mat,k}^2(4b_k+3)}{\theta_{min}^2(8b_k^2-8b_k-3)}\right) \\
&\quad+\frac{1}{2N_k}(\bar{\theta}_2^{(t)}-\theta_2^*)N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}}\left(\frac{1}{\theta_{min}^2}+\frac{C_{mat,k}(4b_k+3)}{\theta_{min}^2(4b_k^2-6b_k-3)}\right)\Bigg)^2 \\
&\quad+\Bigg(\frac{1}{2N_k}(\bar{\theta}_1^{(t)}-\theta_1^*)N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}}\left(\frac{1}{\theta_{min}^2}+\frac{C_{mat,k}(4b_k+3)}{\theta_{min}^2(4b_k^2-6b_k-3)}\right) \\
&\qquad+\frac{1}{2N_k}(\bar{\theta}_2^{(t)}-\theta_2^*)\frac{N_k}{\theta_{min}^2}\Bigg)^2 \\
&\leq \Bigg(\frac{1}{2}(\theta_{max}-\theta_{min})N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1}\left(\frac{1}{\theta_{min}^2}+\frac{C_{mat,k}^2(4b_k+3)}{\theta_{min}^2(8b_k^2-8b_k-3)}\right) \\
&\quad+\frac{1}{2}(\theta_{max}-\theta_{min})N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1}\left(\frac{1}{\theta_{min}^2}+\frac{C_{mat,k}(4b_k+3)}{\theta_{min}^2(4b_k^2-6b_k-3)}\right)\Bigg)^2 \\
&\quad+\left(\frac{1}{2}(\theta_{max}-\theta_{min})N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1}\left(\frac{1}{\theta_{min}^2}+\frac{C_{mat,k}(4b_k+3)}{\theta_{min}^2(4b_k^2-6b_k-3)}\right)+\frac{(\bar{\theta}_2^{(t)}-\theta_2^*)}{2\theta_{min}^2}\right)^2 \\
&\leq a_{mat,1}N_k^{\frac{2(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-2}+\left(a_{mat,2}N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1}+\frac{(\bar{\theta}_2^{(t)}-\theta_2^*)}{2\theta_{min}^2}\right)^2 \\
&\leq a_{mat,1}N_k^{\frac{2(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-2}+a_{mat,2}^2N_k^{\frac{2(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-2}+2a_{mat,2}N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1}\frac{(\bar{\theta}_2^{(t)}-\theta_2^*)}{2\theta_{min}^2} \\
&\qquad+\frac{(\bar{\theta}_2^{(t)}-\theta_2^*)^2}{4\theta_{min}^4} \\
&\leq a_{mat,1}N_k^{\frac{2(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-2}+a_{mat,2}^2N_k^{\frac{2(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-2}+2a_{mat,2}N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1}\frac{(\bar{\theta}_2^{(t)}-\theta_2^*)}{2\theta_{min}^2} \\
&\quad+\frac{1}{4\theta_{min}^4}\Bigg(\frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{t+1} \\
&\qquad+2\beta_1\max_k C_{4k}M_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1}+2\beta_1\Bigg(\sum_{k=1}^K\sqrt{2}p_k(\theta_{max}-\theta_{min})C_{\boldsymbol{\theta}}M_k^{\epsilon_k-\frac{1}{2}}\Bigg)\Bigg)
\end{aligned}$$

where

$$\begin{aligned}
a_{mat,1}=\Bigg(&\frac{1}{2}(\theta_{max}-\theta_{min})\left(\frac{1}{\theta_{min}^2}+\frac{C_{mat,k}^2(4b_k+3)}{\theta_{min}^2(8b_k^2-8b_k-3)}\right) \\
&+\frac{1}{2}(\theta_{max}-\theta_{min})\left(\frac{1}{\theta_{min}^2}+\frac{C_{mat,k}(4b_k+3)}{\theta_{min}^2(4b_k^2-6b_k-3)}\right)\Bigg)^2
\end{aligned}$$

and $a_{mat,2} = \frac{1}{2}(\theta_{max} - \theta_{min})\left(\frac{1}{\theta_{min}^2} + \frac{C_{mat,k}(4b_k+3)}{\theta_{min}^2(4b_k^2-6b_k-3)}\right)$.

By Lemma 49 and Lemma 48, with probability at least

$$1 - \max_k\{C_{\boldsymbol{\theta}}(t+1)(\log(M_k^{\epsilon_k - \frac{1}{2}}))^4 \mathsf{exp}\{-c_{\boldsymbol{\theta}} M_k^{2\epsilon_k}\}\}$$

$$\begin{aligned}
&\left\|\nabla L_k(\bar{\boldsymbol{\theta}}^{(t)})\right\|_2^2 \\
&\leq \frac{2\beta_1^2\left(8(E-1)^2+2\right)G^2}{4\theta_{min}^4(t+1)} + \mathcal{O}\left(M_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1} + \sum_{k=1}^K p_k M_k^{\epsilon_k - \frac{1}{2}} + N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}-1}\right).
\end{aligned}$$

For the partial device participation scenario, the proof is similar.

C.8 Proof of Lemmas

C.8.1 Proof of Lemma 46

Remember that the eigenvalues in each device k are different. For the sake of neatness, we omit the subscript k in the eigenvalues. Let $r_k = j^{\frac{4b_k}{4b_k+3}}$ and $\delta_k = \frac{1}{N_k^{\alpha_k+1}}$, where $0 < \alpha_k < \frac{8b_k^2-12b_k-6}{4b_k+3}$, then, by Lemma 44, with probability at least $1-\delta_k$, we have

$$\begin{aligned}
C(r_k, N_k) &< r_k\sqrt{\frac{r_k(r_k+1)}{N_k\delta_k\lambda_{r_k}^*}} = j^{\frac{4b_k}{4b_k+3}}\sqrt{\frac{j^{\frac{4b_k}{4b_k+3}}(j^{\frac{4b_k}{4b_k+3}}+1)}{C_k j^{\frac{-8b_k^2}{4b_k+3}}}N_k^{\frac{\alpha_k}{2}}} \\
&= N_k^{\frac{\alpha}{2}} j^{\frac{4b_k^2+6b_k}{4b_k+3}}\sqrt{\frac{j^{\frac{4b_k}{4b_k+3}}(j^{-\frac{4b_k}{4b_k+3}}+1)}{C_k}} \leq N_k^{\frac{\alpha}{2}} j^{\frac{4b_k^2+8b_k}{4b_k+3}}\sqrt{\frac{2}{C_k}}
\end{aligned}$$

and

$$\begin{aligned}
H(r_k, N_k) &< \frac{C_k}{2b_k-1} r^{-(2b_k-1)} + \sqrt{\frac{2C_k}{2b_k-1}} r^{-(b_k-1/2)} N_k^{\alpha/2} \\
&\leq \left(\frac{C_k}{2b_k-1} + \sqrt{\frac{2C_k}{2b_k-1}}\right) j^{-\frac{2b_k(2b_k-1)}{4b_k+3}} N_k^{\alpha_k/2}.
\end{aligned}$$

Therefore, by Lemma 44, we obtain

$$
\begin{aligned}
\frac{\lambda_j(\boldsymbol{K}_{f,N_k})}{N_k} &\leq \lambda_j^* + \lambda_j^* N_k^{\frac{\alpha}{2}} j^{\frac{4b_k^2+8b_k}{4b_k+3}} \sqrt{\frac{2}{C_k}} + \left(\frac{C_k}{2b_k-1} + \sqrt{\frac{2C_k}{2b_k-1}} \right) j^{-\frac{2b_k(2b_k-1)}{4b_k+3}} N_k^{\alpha_k/2} \\
&= C_k j^{-2b_k} + C_k j^{-2b_k} N_k^{\frac{\alpha_k}{2}} j^{\frac{4b_k^2+8b_k}{4b_k+3}} \sqrt{\frac{2}{C_k}} + \left(\frac{C_k}{2b_k-1} + \sqrt{\frac{2C_k}{2b_k-1}} \right) j^{-\frac{2b_k(2b_k-1)}{4b_k+3}} N_k^{\alpha_k/2}.
\end{aligned}
$$

This implies

$$
\begin{aligned}
\lambda_j(\boldsymbol{K}_{f,N_k}) &\leq C_k j^{-2b_k} \left(N_k + N_k^{1+\frac{\alpha}{2}} j^{\frac{4b_k^2+8b_k}{4b_k+3}} \sqrt{\frac{2}{C_k}} \right) \\
&\quad + \left(\frac{C_k}{2b_k-1} + \sqrt{\frac{2C_k}{2b_k-1}} \right) j^{-\frac{2b_k(2b_k-1)}{4b_k+3}} N_k^{1+\alpha_k/2} \\
&\leq \left(2\sqrt{2C_k} + \frac{C_k}{2b_k-1} + \sqrt{\frac{2C_k}{2b_k-1}} \right) j^{-\frac{2b_k(2b_k-1)}{4b_k+3}} N_k^{1+\alpha_k/2},
\end{aligned}
$$

where probability at least $1 - \frac{1}{N_k^{\alpha_k+1}}$. Let $C_{mat,k} = \left(2\sqrt{2C_k} + \frac{C_k}{2b_k-1} + \sqrt{\frac{2C_k}{2b_k-1}} \right)$.

Therefore, we have

$$
\sum_{j=1}^{N_k} \frac{\lambda_{1j}^2}{\left(\theta_{1k}^{(t)} \lambda_{1j} + \theta_{2k}^{(t)} \lambda_{2j} \right)^2} \leq \frac{L_{mat,k}}{\theta_{min}^2} + \frac{C_{mat,k}^2}{\theta_{min}^2} \sum_{j=L_{mat,k}}^{\infty} j^{-\frac{4b_k(2b_k-1)}{4b_k+3}} N_k^{2+\alpha_k}
$$

for any $0 < L_{mat,k} \leq N_k$. Let $L_{mat,k} = N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}}$, then we obtain

$$
\sum_{j=1}^{N_k} \frac{\lambda_{1j}^2}{\left(\theta_{1k}^{(t)} \lambda_{1j} + \theta_{2k}^{(t)} \lambda_{2j} \right)^2} \leq N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}} \left(\frac{1}{\theta_{min}^2} + \frac{C_{mat,k}^2(4b_k+3)}{\theta_{min}^2(8b_k^2-8b_k-3)} \right).
$$

Similarly, we have

$$
\sum_{j=1}^{N_k} \frac{\lambda_{1j}\lambda_{2j}}{\left(\theta_{1k}^{(t)} \lambda_{1j} + \theta_{2k}^{(t)} \lambda_{2j} \right)^2} \leq N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}} \left(\frac{1}{\theta_{min}^2} + \frac{C_{mat,k}(4b_k+3)}{\theta_{min}^2(4b_k^2-6b_k-3)} \right).
$$

Additionally, we can show that

$$
\sum_{j=1}^{N_k} \frac{\lambda_{2j}^2}{\left(\theta_{1k}^{(t)} \lambda_{1j} + \theta_{2k}^{(t)} \lambda_{2j} \right)^2} \geq \frac{|\{j : \theta_{1k}^{(t)} \lambda_{1j} + \theta_{2k}^{(t)} \lambda_{2j} \leq 2\theta_{max}\}|}{4\theta_{max}^2}.
$$

The fact that $j : \theta_{1k}^{(t)}\lambda_{1j} + \theta_{2k}^{(t)}\lambda_{2j} \leq 2\theta_{max}$ implies

$$\begin{aligned} & C_{mat,k}\theta_{max} j^{-\frac{2b_k(2b_k-1)}{4b_k+3}} N_k^{1+\alpha_k/2} \leq \theta_{max} \\ & \Rightarrow j \geq C_{mat,k}^{\frac{4b_k+3}{2b_k(2b_k-1)}} N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}} \geq C_{mat,k} N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}} \end{aligned}$$

since $b_k \geq \frac{\sqrt{21}+3}{4}$. Therefore,

$$\frac{N_k - C_{mat,k} N_k^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}}}{4\theta_{max}} \leq \sum_{j=1}^{N_k} \frac{\lambda_{2j}^2}{\left(\theta_{1k}^{(t)}\lambda_{1j} + \theta_{2k}^{(t)}\lambda_{2j}\right)^2} \leq \frac{N_k}{\theta_{min}^2}$$

where the upper bound is trivially true.

C.8.2 Proof of Lemma 47

Proof. For compactness, we drop the subscript k in M_k. For device k, denote by $\boldsymbol{\theta}_k^{(t)} = (\theta_{1k}^{(t)}, \theta_{2k}^{(t)})$ the model parameter at iteration t. Let $\lambda_{1jk}^{(t)}$ be the j^{th} largest eigenvalue of $\boldsymbol{K}_{f,\xi_k^{(t)}}$ and $\lambda_{2jk}^{(t)} = 1$ be the j^{th} largest eigenvalue of $\boldsymbol{I}_M$. By definition,

$$\begin{aligned} & \left[g_k^*(\boldsymbol{\theta}_k^{(t)})\right]_1 \\ & = \frac{1}{2s_1(M)} \operatorname{Tr}\left[\boldsymbol{K}_{\xi_k^{(t)}}(\boldsymbol{\theta}_k^{(t)})^{-1}\left(\boldsymbol{I}_M - \boldsymbol{K}_{\xi_k^{(t)}}(\boldsymbol{\theta}_k^{*})\boldsymbol{K}_{\xi_k^{(t)}}(\boldsymbol{\theta}_k^{(t)})^{-1}\right)\frac{\partial \boldsymbol{K}_{\xi_k^{(t)}}(\boldsymbol{\theta}_k^{(t)})}{\partial \theta_{1k}^{(t)}}\right] \\ & = \frac{1}{2s_1(M)}(\theta_{1k}^{(t)} - \theta_{1k}^{*}) \sum_{j=1}^{M} \frac{\lambda_{1jk}^{(t)2}}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2} \\ & \quad + \frac{1}{2s_1(M)}(\theta_{2k}^{(t)} - \theta_{2k}^{*}) \sum_{j=1}^{M} \frac{\lambda_{2jk}^{(t)}\lambda_{1jk}^{(t)}}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2} \end{aligned}$$

and

$$\begin{aligned}
&\left[g_k^*(\boldsymbol{\theta}_k^{(t)})\right]_2 \\
&= \frac{1}{2M} \operatorname{Tr}\left[\boldsymbol{K}_{\xi_k^{(t)}}(\boldsymbol{\theta}_k^{(t)})^{-1}\left(\boldsymbol{I}_M - \boldsymbol{K}_{\xi_k^{(t)}}(\boldsymbol{\theta}_k^*)\boldsymbol{K}_{\xi_k^{(t)}}(\boldsymbol{\theta}_k^{(t)})^{-1}\right)\frac{\partial \boldsymbol{K}_{\xi_k^{(t)}}(\boldsymbol{\theta}_k^{(t)})}{\partial \theta_{2k}^{(t)}}\right] \\
&= \frac{1}{2M}(\theta_{1k}^{(t)} - \theta_{1k}^*)\sum_{j=1}^M \frac{\lambda_{1jk}^{(t)}}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2} \\
&\qquad + \frac{1}{2s_1(M)}(\theta_{2k}^{(t)} - \theta_{2k}^*)\sum_{j=1}^M \frac{\lambda_{2jk}^{(t)}}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2}.
\end{aligned}$$

Based on those two expressions, we can obtain

$$\begin{aligned}
&\langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k^*(\boldsymbol{\theta}_k^{(t)})\rangle \\
&= (\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*)^\intercal \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} (\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*)
\end{aligned}$$

where $A_{11}, A_{12}, A_{21}, A_{22}$ will be clarified shortly. Let $\epsilon_k = \frac{1}{2}$, by Lemma 45, with probability at least $1 - \frac{2}{M^{\alpha_k}}$,

$$\begin{aligned}
A_{11} &:= \frac{1}{2\tau \log M}\sum_{j=1}^M \frac{\lambda_{1jk}^{(t)2}}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2} \geq \frac{1}{2\tau \log M}\frac{\epsilon_k \log M}{8b_k\theta_{max}^2} = \frac{1}{32\tau b_k \theta_{max}^2}, \\
A_{12} &:= \frac{1}{2\tau \log M}\sum_{j=1}^M \frac{\lambda_{1jk}^{(t)}}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2} \leq \frac{1}{2\tau \log M}\frac{5 + 2\alpha_k}{7b_k\theta_{min}^2}\log M \leq \frac{1}{2\tau b_k \theta_{min}^2}, \\
A_{21} &:= \frac{1}{2M}\sum_{j=1}^M \frac{\lambda_{1jk}^{(t)}}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2} \leq \frac{1}{2M}\frac{5 + 2\alpha_k}{7b_k\theta_{min}^2}\log M = \frac{5 + 2\alpha_k}{14b_k\theta_{min}^2}\frac{\log M}{M} \\
&\leq \frac{1}{2b_k\theta_{min}^2}\frac{\log M}{M}, \\
A_{12} &:= \frac{1}{2M}\sum_{j=1}^M \frac{1}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2} \\
&\geq \frac{1}{2M}\frac{M - C_{1k}(\alpha_k, b_k)\log M}{4\theta_{max}^2}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\langle \boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}^*, g_k^*(\boldsymbol{\theta}_k^{(t)})\rangle \\
&\geq \left(\frac{1}{64\tau b_k \theta_{max}^2} - \frac{\log M}{64\theta_{max}^2 b_k M}\right)(\theta_{1k}^{(t)} - \theta_1^*)^2 \\
&\quad + \left(\frac{1}{8\theta_{max}^2} - \frac{4\theta_{max}^2}{\tau b_k \theta_{min}^4} - \frac{C_{1k}(\alpha_k, b_k)\log M}{8\theta_{max}^2 M} + \frac{4\theta_{max}^2 \log M}{b\theta_{min}^4 M}\right)(\theta_{2k}^{(t)} - \theta_1^*)^2 \\
&= \frac{1}{64\tau b_k \theta_{max}^2}(\theta_{1k}^{(t)} - \theta_1^*)^2 + \left(\frac{1}{8\theta_{max}^2} - \frac{4\theta_{max}^2}{\tau b_k \theta_{min}^4}\right)(\theta_{2k}^{(t)} - \theta_1^*)^2 \\
&\quad - \frac{\log M}{64\theta_{max}^2 b_k M}\theta_{max}^2 - \frac{C_{1k}(\alpha_k, b_k)\log M}{8\theta_{max}^2 M}\theta_{max}^2 + \frac{4\theta_{max}^2 \log M}{b_k \theta_{min}^4 M}\theta_{min}^2 \\
&\geq \frac{\gamma_k}{2}\left\|\boldsymbol{\theta}_k^{(t)} - \boldsymbol{\theta}_k^*\right\|_2^2 - C_{3k}(\alpha_k, b_k)\frac{\log M}{M},
\end{aligned}$$

where

$$\gamma_k = \min\left\{\frac{1}{32\tau b_k \theta_{max}^2}, \frac{1}{4\theta_{max}^2} - \frac{8\theta_{max}^2}{\tau b_k \theta_{min}^4}\right\} > 0$$

and

$$C_{3k}(\alpha_k, b_k) = \frac{1}{64 b_k} + \frac{C_{1k}(\alpha_k, b_k)}{8} - \frac{4\theta_{max}^2}{b\theta_{min}^2}.$$

□

C.8.3 Proof of Lemma 48

For compactness, we drop the subscript k in M_k. By definition, we can show that

$$\begin{aligned}
&\left[g_k^*(\boldsymbol{\theta}_k^{(t)})\right]_2 \\
&= \frac{1}{2M}\operatorname{Tr}\left[\boldsymbol{K}_{\xi_k^{(t)}}(\boldsymbol{\theta}_k^{(t)})^{-1}\left(\boldsymbol{I}_M - \boldsymbol{K}_{\xi_k^{(t)}}(\boldsymbol{\theta}_k^*)\boldsymbol{K}_{\xi_k^{(t)}}(\boldsymbol{\theta}_k^{(t)})^{-1}\right)\frac{\partial \boldsymbol{K}_{\xi_k^{(t)}}(\boldsymbol{\theta}_k^{(t)})}{\partial \theta_{2k}^{(t)}}\right] \\
&= \frac{1}{2M}(\theta_{1k}^{(t)} - \theta_{1k}^*)\sum_{j=1}^M \frac{\lambda_{1jk}^{(t)}}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2} \\
&\quad + \frac{1}{2M}(\theta_{2k}^{(t)} - \theta_{2k}^*)\sum_{j=1}^M \frac{\lambda_{2jk}^{(t)}}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\left[g_k^*(\boldsymbol{\theta}_k^{(t)})\right]_2 (\theta_{2k}^{(t)} - \theta_{2k}^*) \\
&= \frac{1}{2M}(\theta_{1k}^{(t)} - \theta_{1k}^*)(\theta_{2k}^{(t)} - \theta_{2k}^*) \sum_{j=1}^{M} \frac{\lambda_{1jk}^{(t)}}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2} \\
&\quad + \frac{1}{2M}(\theta_{2k}^{(t)} - \theta_{2k}^*)^2 \sum_{j=1}^{M} \frac{\lambda_{2jk}^{(t)}}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2} \\
&\geq \frac{1}{2M}(\theta_{2k}^{(t)} - \theta_{2k}^*)^2 \sum_{j=1}^{M} \frac{1}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2} \\
&\quad - \frac{1}{2M}(\theta_{max} - \theta_{min})^2 \sum_{j=1}^{M} \frac{\lambda_{1jk}^{(t)}}{(\theta_{1k}^{(t)}\lambda_{1jk}^{(t)} + \theta_{2k}^{(t)}\lambda_{2jk}^{(t)})^2} \\
&\geq \frac{1}{2M}(\theta_{2k}^{(t)} - \theta_{2k}^*)^2 \frac{M - C_{mat,k} M^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}}}{4\theta_{max}} \\
&\quad - \frac{1}{2M}(\theta_{max} - \theta_{min})^2 M^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}} \left(\frac{1}{\theta_{min}^2} + \frac{C_{mat,k}(4b_k+3)}{\theta_{min}^2(4b_k^2 - 6b_k - 3)}\right)
\end{aligned}$$

with probability at least $1 - \frac{1}{M^{1+\alpha_k}}$. Therefore,

$$\begin{aligned}
&\left[g_k^*(\boldsymbol{\theta}_k^{(t)})\right]_2 (\theta_{2k}^{(t)} - \theta_{2k}^*) \\
&\quad \geq \frac{\gamma_k}{2}(\theta_{2k}^{(t)} - \theta_{2k}^*)^2 - (\theta_{max} - \theta_{min})^2 M^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)} - 1} \left(\frac{1}{2\theta_{min}^2} + \frac{C_{mat,k}(4b_k+3)}{2\theta_{min}^2(4b_k^2 - 6b_k - 3)}\right),
\end{aligned}$$

where we slightly abuse the notation and define $\gamma_k := \frac{1}{2M}\frac{M - C_{mat,k} M^{\frac{(2+\alpha_k)(4b_k+3)}{4b_k(2b_k-1)}}}{4\theta_{max}}$. Here note that this γ_k is different from the γ_k in the Lemmas/Theorems involved with RBF kernels.

BIBLIOGRAPHY

José Ángel Morell and Enrique Alba. Dynamic and adaptive fault-tolerant asynchronous federated learning using volunteer edge devices. *Future Generation Computer Systems*, 133:53–67, 2022.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

Philippa Lawson, Brenda McPhail, and Eric Lawton. The connected car: Who is in the driver's seat? a study on privacy and onboard vehicle telematics technology. 2015.

Fuxun Yu, Weishan Zhang, Zhuwei Qin, Zirui Xu, Di Wang, Chenchen Liu, Zhi Tian, and Xiang Chen. Heterogeneous federated learning. *arXiv preprint arXiv:2008.06767*, 2020a.

Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021a.

Samsung for Business Samsung. Your phone is now more powerful than your pc. `https://insights.samsung.com/2021/08/19/your-phone-is-now-more-powerful-than-your-pc-3/`, 2019. Accessed: 2019-02-19.

CleanTechnica. Tesla fsd hardware has 150 million times more computer power than apollo 11 computer, 2021. Accessed: 2021-05-24.

AR Al-Ali, Ragini Gupta, and Ahmad Al Nabulsi. Cyber physical systems role in manufacturing technologies. In *AIP Conference Proceedings*, volume 1957, page 050007. AIP Publishing LLC, 2018.

Raed Kontar, Naichen Shi, Xubo Yue, Seokhyun Chung, Eunshin Byon, Mosharaf Chowdhury, Jionghua Jin, Wissam Kontar, Neda Masoud, Maher Nouiehed, et al. The internet of federated things (ioft). *IEEE Access*, 9:156071–156113, 2021.

Jingyan Jiang, Liang Hu, Chenghao Hu, Jiate Liu, and Zhi Wang. Bacombo—bandwidth-aware decentralized federated learning. *Electronics*, 9(3):440, 2020a.

Kai Yang, Tao Jiang, Yuanming Shi, and Zhi Ding. Federated learning via over-the-air computation. *IEEE Transactions on Wireless Communications*, 19(3):2022–2035, 2020.

Xiongtao Zhang, Xiaomin Zhu, Ji Wang, Hui Yan, Huangke Chen, and Weidong Bao. Federated learning with adaptive communication compression under dynamic bandwidth and unreliable networks. *Information Sciences*, 540:242–262, 2020a.

Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020a.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329*, 2019.

Zhongxiang Dai, Kian Hsiang Low, and Patrick Jaillet. Federated bayesian optimization via thompson sampling. *arXiv preprint arXiv:2010.10154*, 2020.

Mikhail Khodak, Renbo Tu, Tian Li, Liam Li, Maria-Florina Balcan, Virginia Smith, and Ameet Talwalkar. Federated hyperparameter tuning: Challenges, baselines, and connections to weight-sharing. *arXiv preprint arXiv:2106.04502*, 2021.

Xubo Yue and Raed Al Kontar. Federated gaussian process: Convergence, automatic personalization and multi-fidelity modeling. *arXiv preprint arXiv:2111.14008*, 2021.

Binxuan Hu, Yujia Gao, Liang Liu, and Huadong Ma. Federated region-learning: An edge computing based framework for urban environment sensing. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2018.

Ji Chu Jiang, Burak Kantarci, Sema Oktug, and Tolga Soyata. Federated learning in smart city sensing: Challenges and opportunities. *Sensors*, 20(21):6230, 2020b.

Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1):1–19, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

Honglin Yuan and Tengyu Ma. Federated accelerated stochastic gradient descent. *Conference on Neural Information Processing Systems*, 2020.

Hung T Nguyen, Vikash Sehwag, Seyyedali Hosseinalipour, Christopher G Brinton, Mung Chiang, and H Vincent Poor. Fast-convergent federated learning. *IEEE Journal on Selected Areas in Communications*, 39(1):201–218, 2020.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018a.

Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

Naichen Shi, Raed Al Kontar, and Salar Fattahi. Heterogeneous matrix factorization: When features differ by datasets. *arXiv preprint arXiv:2305.17744*, 2023a.

Naichen Shi, Fan Lai, Raed Al Kontar, and Mosharaf Chowdhury. Ensemble models in federated learning for improved generalization and uncertainty quantification. *IEEE Transactions on Automation Science and Engineering*, 2023b.

Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*, 2019.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *International Conference on Learning Representations*, 2020a.

Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *Advances in Neural Information Processing Systems 33*, 2020.

Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643. PMLR, 2019.

Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Conference on Neural Information Processing Systems*, 2020b.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019.

Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *International Conference on Learning Representations*, 2019a.

Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. Fairness-aware agnostic federated learning. *arXiv preprint arXiv:2010.05057*, 2020.

Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu. Fedmgda+: Federated learning meets multi-objective optimization. *arXiv preprint arXiv:2006.11489*, 2020.

Wei Huang, Tianrui Li, Dexian Wang, Shengdong Du, and Junbo Zhang. Fairness and accuracy in federated learning. *arXiv preprint arXiv:2012.10069*, 2020.

Daniel Yue Zhang, Ziyi Kou, and Dong Wang. Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1051–1060. IEEE, 2020b.

Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019b.

Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. Tilted empirical risk minimization. *arXiv preprint arXiv:2007.01162*, 2020b.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. *Conference on Neural Information Processing Systems*, 2016.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180, 2017.

Jingfeng Zhang, Cheng Li, Antonio Robles-Kelly, and Mohan Kankanhalli. Hierarchically fair federated learning. *arXiv preprint arXiv:2004.10386*, 2020c.

Xinyi Xu and Lingjuan Lyu. Towards building a robust and fair federated learning system. *arXiv preprint arXiv:2011.10464*, 2020.

Lingjuan Lyu, Xinyi Xu, Qian Wang, and Han Yu. Collaborative fairness in federated learning. In *Federated Learning*, pages 189–204. Springer, 2020.

Naichen Shi and Raed Al Kontar. Personalized federated learning via domain adaptation with an application to distributed 3d printing. *Technometrics*, (just-accepted):1–22, 2022a.

Geyu Liang, Naichen Shi, Raed Al Kontar, and Salar Fattahi. Personalized dictionary learning for heterogeneous datasets. *arXiv preprint arXiv:2305.15311*, 2023.

Naichen Shi and Raed Al Kontar. Personalized pca: Decoupling shared and unique features. *arXiv preprint arXiv:2207.08041*, 2022b.

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020b.

Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. *arXiv preprint arXiv:2006.08848*, 2020.

Xiang Li, Wenhao Yang, Shusen Wang, and Zhihua Zhang. Communication-efficient local decentralized sgd methods. *arXiv preprint arXiv:1910.09126*, 2019c.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

Michael N Jones and Douglas JK Mewhort. Case-sensitive letter and bigram frequency counts from large-scale english corpora. *Behavior research methods, instruments, & computers*, 36(3):388–396, 2004.

Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Najmul Hassan, Saira Gillani, Ejaz Ahmed, Ibrar Yaqoob, and Muhammad Imran. The role of edge computing in internet of things. *IEEE communications magazine*, 56(11):110–115, 2018.

Beatriz Blanco-Filgueira, Daniel Garcia-Lesta, Mauro Fernández-Sanjurjo, Víctor Manuel Brea, and Paula López. Deep learning-based multiple object visual tracking on embedded system for iot and mobile edge computing applications. *IEEE Internet of Things Journal*, 6(3):5423–5431, 2019.

Md Abdur Rahman and M Shamim Hossain. An internet-of-medical-things-enabled edge computing framework for tackling covid-19. *IEEE Internet of Things Journal*, 8(21):15847–15854, 2021.

Sha Zhu, Kaoru Ota, and Mianxiong Dong. Green ai for iiot: Energy efficient intelligent edge computing for industrial internet of things. *IEEE Transactions on Green Communications and Networking*, 2021b.

Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

Shaoshan Liu, Liangkai Liu, Jie Tang, Bo Yu, Yifan Wang, and Weisong Shi. Edge computing for autonomous driving: Opportunities and challenges. *Proceedings of the IEEE*, 107(8):1697–1716, 2019.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020c.

Xubo Yue, Maher Nouiehed, and Raed Al Kontar. Salr: Sharpness-aware learning rates for improved generalization. *arXiv preprint arXiv:2011.05348*, 2020.

Michael I Jordan, Jason D Lee, and Yun Yang. Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, 2018.

Jianqing Fan, Yongyi Guo, and Kaizheng Wang. Communication-efficient accurate statistical estimation. *Journal of the American Statistical Association*, (just-accepted):1–29, 2021.

Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

Xubo Yue, Raed Al Kontar, and Ana María Estrada Gómez. Federated data analytics: A study on linear models. *IISE Transactions*, (just-accepted):1–25, 2022a.

Kaibo Liu, Nagi Z Gebraeel, and Jianjun Shi. A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. *IEEE Transactions on Automation Science and Engineering*, 10(3):652–664, 2013.

Wujun Si, Qingyu Yang, and Xin Wu. A distribution-based functional linear model for reliability analysis of advanced high-strength dual-phase steels by utilizing material microstructure images. *IISE Transactions*, 49(9):863–873, 2017.

Jian Li, Jiakun Xu, and Qiang Zhou. Monitoring serially dependent categorical processes with ordinal information. *IISE Transactions*, 50(7):596–605, 2018b.

Marc-Andre Schulz, BT Yeo, Joshua T Vogelstein, Janaina Mourao-Miranada, Jakob N Kather, Konrad Kording, Blake Richards, and Danilo Bzdok. Different scaling of linear models and deep learning in ukbiobank brain images versus machine-learning datasets. *Nature communications*, 11(1):1–15, 2020.

Mohammad Arashi, Mahdi Roozbeh, Nor Aishah Hamzah, and M Gasparini. Ridge regression and its applications in genetic studies. *Plos one*, 16(4):e0245376, 2021.

Durmuş Özkan Şahın, Sedat Akleylek, and Erdal Kiliç. Linregdroid: Detection of android malware using multiple linear regression models-based classifiers. *IEEE Access*, 10:14246–14259, 2022.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Wei Liu, Li Chen, Yunfei Chen, and Wenyi Zhang. Accelerating federated learning via momentum gradient descent. *IEEE Transactions on Parallel and Distributed Systems*, 31(8):1754–1766, 2020.

Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. A fairness-aware incentive scheme for federated learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 393–399, 2020c.

Xubo Yue, Maher Nouiehed, and Raed Al Kontar. Gifair-fl: An approach for group and individual fairness in federated learning. *arXiv preprint arXiv:2108.02741*, 2021.

Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. Fairness-aware agnostic federated learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 181–189. SIAM, 2021.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020d.

Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of local-update sgd algorithms. *Journal of Machine Learning Research*, 22(213):1–50, 2021.

Naichen Shi, Fan Lai, Raed Al Kontar, and Mosharaf Chowdhury. Fed-ensemble: Improving generalization through model ensembling in federated learning. *arXiv preprint arXiv:2107.10663*, 2021.

Honglin Yuan, Manzil Zaheer, and Sashank Reddi. Federated composite optimization. In *International Conference on Machine Learning*, pages 12253–12266. PMLR, 2021.

Qianqian Tong, Guannan Liang, Tan Zhu, and Jinbo Bi. Federated nonconvex sparse learning. *arXiv preprint arXiv:2101.00052*, 2020.

Yu Zhang and Dit-Yan Yeung. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*, 2012.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Joaquin Vanschoren. Meta-learning. In *Automated Machine Learning*, pages 35–61. Springer, Cham, 2019.

Jim Albert and Jingchen Hu. *Probability and Bayesian modeling*. CRC Press, 2019.

Thomas Peter Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

Thomas P Minka. Expectation propagation for approximate bayesian inference. *arXiv preprint arXiv:1301.2294*, 2013.

Aki Vehtari, Andrew Gelman, Tuomas Sivula, Pasi Jylänki, Dustin Tran, Swupnil Sahai, Paul Blomstedt, John P Cunningham, David Schiminovich, and Christian P Robert. Expectation propagation as a way of life: A framework for bayesian inference on partitioned data. *J. Mach. Learn. Res.*, 21(17):1–53, 2020.

Simon Barthelme. Simon barthelme: The expectation-propagation algorithm: a tutorial - part 1, 2016. URL `https://www.carmin.tv/en/speakers/simon-barthelme`.

Christopher K Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Sara Van Erp, Daniel L Oberski, and Joris Mulder. Shrinkage priors for bayesian penalized regression. *Journal of Mathematical Psychology*, 89:31–50, 2019.

Paulo Cortez and Alice Maria Gonçalves Silva. Using data mining to predict secondary school student performance. 2008.

Changyue Song and Kaibo Liu. Statistical degradation modeling and prognostics of multiple sensor signals via data fusion: A composite health index approach. *IISE Transactions*, 50(10):853–867, 2018.

Olvi L Mangasarian and Mikhail V Solodov. Backpropagation convergence via deterministic nonmonotone perturbed minimization. *Advances in Neural Information Processing Systems*, pages 383–383, 1994.

Virginia Smith, Simone Forte, Ma Chenxin, Martin Takáč, Michael I Jordan, and Martin Jaggi. Cocoa: A general framework for communication-efficient distributed optimization. *Journal of Machine Learning Research*, 18:230, 2018.

Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018.

Nguyen H Tran, Wei Bao, Albert Zomaya, Minh NH Nguyen, and Choong Seon Hong. Federated learning over wireless networks: Optimization model design and analysis. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 1387–1395. IEEE, 2019.

Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Blockchained on-device federated learning. *IEEE Communications Letters*, 24(6):1279–1283, 2019.

Xubo Yue and Raed Kontar. The renyi gaussian process: Towards improved generalization. *arXiv preprint arXiv:1910.06990*, 2019.

Xubo Yue, Maher Nouiehed, and Raed Al Kontar. Gifair-fl: A framework for group and individual fairness in federated learning. *INFORMS Journal on Data Science*, 2022b.

Yuchen Zeng, Hongxu Chen, and Kangwook Lee. Improving fairness via federated learning. *arXiv preprint arXiv:2110.15545*, 2021.

Santu Rana, Cheng Li, Sunil Gupta, Vu Nguyen, and Svetha Venkatesh. High dimensional bayesian optimization with elastic gaussian process. In *International conference on machine learning*, pages 2883–2891. PMLR, 2017.

Xubo Yue and Raed AL Kontar. Why non-myopic bayesian optimization is promising and how far should we look-ahead? a study via rollout. In *International Conference on Artificial Intelligence and Statistics*, pages 2808–2818. PMLR, 2020a.

Shali Jiang, Henry Chai, Javier Gonzalez, and Roman Garnett. Binoculars for efficient, nonmyopic sequential experimental design. In *International Conference on Machine Learning*, pages 4794–4803. PMLR, 2020c.

Arvind Krishna, V Roshan Joseph, Shan Ba, William A Brenneman, and William R Myers. Robust experimental designs for model calibration. *Journal of Quality Technology*, pages 1–12, 2021.

Grace Tapia, Alaa H Elwany, and Huiyan Sang. Prediction of porosity in metal-based additive manufacturing using spatial gaussian process models. *Additive Manufacturing*, 12:282–290, 2016.

Weiwen Peng, Yan-Feng Li, Yuan-Jian Yang, Jinhua Mi, and Hong-Zhong Huang. Bayesian degradation analysis with inverse gaussian process models under time-varying degradation rates. *IEEE Transactions on Reliability*, 66(1):84–96, 2017.

Xubo Yue and Raed Al Kontar. Joint models for event prediction from time series and survival data. *Technometrics*, pages 1–10, 2020b.

Seokhyun Chung and Raed Kontar. Functional principal component analysis for extrapolating multistream longitudinal data. *IEEE Transactions on Reliability*, 70(4):1321–1331, 2020.

Seokhyun Chung, Cheng-Hao Chou, Xiaozhu Fang, Raed Al Kontar, and Chinedum Okwudire. A multi-stage approach for knowledge-guided predictions with application to additive manufacturing. *IEEE Transactions on Automation Science and Engineering*, 19(3):1675–1687, 2022a.

Xubo Yue and Raed Al Kontar. Optimize to generalize in gaussian processes: An alternative objective based on the rényi divergence. *IISE Transactions*, (just-accepted):1–21, 2023.

Farhad Imani, Changqing Cheng, Ruimin Chen, and Hui Yang. Nested gaussian process modeling for high-dimensional data imputation in healthcare systems. In *IISE 2018 Conference & Expo, Orlando, FL, May*, pages 19–22, 2018.

Shwet Ketu and Pramod Kumar Mishra. Enhanced gaussian process regression-based forecasting model for covid-19 outbreak and significance of iot for its detection. *Applied Intelligence*, 51(3):1492–1512, 2021.

Seokhyun Chung, Raed Al Kontar, and Zhenke Wu. Weakly supervised multi-output regression via correlated gaussian processes. *INFORMS Journal on Data Science*, 1(2):115–137, 2022b.

Sepideh Afkhami Goli, Behrouz H Far, and Abraham O Fapojuwo. Vehicle trajectory prediction with gaussian process regression in connected vehicle environment. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 550–555. IEEE, 2018.

Bingjie Liu, Xubo Yue, Eunshin Byon, and Raed Al Kontar. Parameter calibration in wake effect simulation model with stochastic gradient descent and stratified sampling. *The Annals of Applied Statistics*, 16(3): 1795–1821, 2022.

Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, 37(2): 408–423, 2013.

Dohyun Jang, Jaehyun Yoo, Clark Youngdong Son, Dabin Kim, and H Jin Kim. Multi-robot active sensing and environmental model learning with distributed gaussian process. *IEEE Robotics and Automation Letters*, 5(4):5905–5912, 2020.

Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical science*, 4(4):409–423, 1989.

Carla Currin, Toby Mitchell, Max Morris, and Don Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416):953–963, 1991.

Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data. *arXiv preprint arXiv:2005.11418*, 2020d.

Reese Pathak and Martin J Wainwright. Fedsplit: An algorithmic framework for fast federated optimization. *arXiv preprint arXiv:2005.05238*, 2020.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526, 2017.

Yarin Gal, Mark Van Der Wilk, and Carl Edward Rasmussen. Distributed variational inference in sparse gaussian process regression and latent variable models. *Advances in neural information processing systems*, 27, 2014.

Marc Deisenroth and Jun Wei Ng. Distributed gaussian processes. In *International Conference on Machine Learning*, pages 1481–1490. PMLR, 2015.

Pablo Moreno-Muñoz, Antonio Artés, and Mauricio Álvarez. Modular gaussian processes for transfer learning. *Advances in Neural Information Processing Systems*, 34:24730–24740, 2021.

Yuchen Zhang, John C Duchi, and Martin J Wainwright. Communication-efficient algorithms for statistical optimization. *The Journal of Machine Learning Research*, 14(1):3321–3363, 2013.

Xiangyu Wang and David B Dunson. Parallelizing mcmc via weierstrass sampler. *arXiv preprint arXiv:1312.4605*, 2013.

Jason D Lee, Qiang Liu, Yuekai Sun, and Jonathan E Taylor. Communication-efficient sparse regression. *The Journal of Machine Learning Research*, 18(1):115–144, 2017.

Shao-Bo Lin, Xin Guo, and Ding-Xuan Zhou. Distributed learning with regularized least squares. *The Journal of Machine Learning Research*, 18(1):3202–3232, 2017.

Xi Chen, Weidong Liu, and Yichen Zhang. Quantile regression under memory constraint. *The Annals of Statistics*, 47(6):3244–3273, 2019.

Xi Chen, Jason D Lee, He Li, and Yun Yang. Distributed estimation for principal component analysis: An enlarged eigenspace analysis. *Journal of the American Statistical Association*, pages 1–12, 2021a.

Xi Chen, Weidong Liu, and Yichen Zhang. First-order newton-type estimator for distributed estimation and inference. *Journal of the American Statistical Association*, pages 1–17, 2021b.

Sihui Zheng, Cong Shen, and Xiang Chen. Design and analysis of uplink and downlink communications for federated learning. *IEEE Journal on Selected Areas in Communications*, 39(7):2150–2167, 2020.

Yanshuai Cao and David J Fleet. Generalized product of experts for automatic and principled fusion of gaussian process predictions. *arXiv preprint arXiv:1410.7827*, 2014.

Jun Wei Ng and Marc Peter Deisenroth. Hierarchical mixture-of-experts model for large-scale gaussian process regression. *arXiv preprint arXiv:1412.3078*, 2014.

Mostafa Tavassolipour, Seyed Abolfazl Motahari, and Mohammad Taghi Manzuri Shalmani. Learning of gaussian processes in distributed and communication limited systems. *IEEE transactions on pattern analysis and machine intelligence*, 42(8):1928–1941, 2019.

Kai Chen, Qinglei Kong, Yijue Dai, Yue Xu, Feng Yin, Lexi Xu, and Shuguang Cui. Recent advances in data-driven wireless communication using gaussian processes: A comprehensive survey. *China Communications*, 19(1):218–237, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Raghav Gnanasambandam, Bo Shen, Jihoon Chung, Xubo Yue, et al. Self-scalable tanh (stan): Faster convergence and better generalization in physics-informed neural networks. *arXiv preprint arXiv:2204.12589*, 2022.

Lei Wu, Chao Ma, et al. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. *Advances in Neural Information Processing Systems*, 31:8279–8288, 2018.

Hao Chen, Lili Zheng, Raed Al Kontar, and Garvesh Raskutti. Stochastic gradient descent in correlated settings: A study on gaussian processes. *Advances in Neural Information Processing Systems*, 33, 2020.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

Michael L Stein. Limitations on low rank approximations for covariance matrices of spatial data. *Spatial Statistics*, 8:1–19, 2014.

David Burt, Carl Edward Rasmussen, and Mark Van Der Wilk. Rates of convergence for sparse variational gaussian process regression. In *International Conference on Machine Learning*, pages 862–871. PMLR, 2019.

Joëlle Bailly and Didier Bailly. Multifidelity aerodynamic optimization of a helicopter rotor blade. *AIAA Journal*, 57(8):3132–3144, 2019.

Kurt Cutajar, Mark Pullin, Andreas Damianou, Neil Lawrence, and Javier González. Deep gaussian processes for multi-fidelity modeling. *arXiv preprint arXiv:1903.07320*, 2019.

Loïc Brevault, Mathieu Balesdent, and Ali Hebbal. Overview of gaussian process based multi-fidelity techniques with variable relationship between fidelities, application to aerospace systems. *Aerospace Science and Technology*, 107:106339, 2020.

M Giselle Fernández-Godino, Chanyoung Park, Nam-Ho Kim, and Raphael T Haftka. Review of multi-fidelity models. *arXiv preprint arXiv:1609.07196*, 2016.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *Siam Review*, 60(3):550–591, 2018.

Marc C Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

Julien Laurenceau and P Sagaut. Building efficient response surfaces of aerodynamic functions with kriging and cokriging. *AIAA journal*, 46(2):498–507, 2008.

Andreas Damianou and Neil D Lawrence. Deep gaussian processes. In *Artificial intelligence and statistics*, pages 207–215. PMLR, 2013.

Shifeng Xiong, Peter ZG Qian, and CF Jeff Wu. Sequential design and analysis of high-accuracy and low-accuracy computer codes. *Technometrics*, 55(1):37–46, 2013.

Dennis D Cox, Jeong-Soo Park, and Clifford E Singer. A statistical method for tuning a computer code to a data base. *Computational statistics & data analysis*, 37(1):77–92, 2001.

Paris Perdikaris, Maziar Raissi, Andreas Damianou, Neil D Lawrence, and George Em Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2198):20160751, 2017.

Hyejung Moon, Angela M Dean, and Thomas J Santner. Two-stage sensitivity-based group screening in computer experiments. *Technometrics*, 54(4):376–387, 2012.

Robert B Gramacy and Heng Lian. Gaussian process single-index models as emulators for computer experiments. *Technometrics*, 54(1):30–41, 2012.

Duy Nguyen-Tuong, Matthias Seeger, and Jan Peters. Computed torque control with nonparametric regression models. In *2008 American Control Conference*, pages 212–217. IEEE, 2008.

Duy Nguyen-Tuong and Jan Peters. Model learning for robot control: a survey. *Cognitive processing*, 12(4): 319–340, 2011.

Thang D Bui, Cuong V Nguyen, Siddharth Swaroop, and Richard E Turner. Partitioned variational inference: A unified framework encompassing federated and continual learning. *arXiv preprint arXiv:1811.11206*, 2018.

Abhinav Saxena and Kai Goebel. Turbofan engine degradation simulation data set. *NASA Ames Prognostics Data Repository*, pages 878–887, 2008.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Hao Yan, Kaibo Liu, Xi Zhang, and Jianjun Shi. Multiple sensor data fusion for degradation modeling and prognostics under multiple operational conditions. *IEEE Transactions on Reliability*, 65(3):1416–1426, 2016.

Mikio L Braun. Accurate error bounds for the eigenvalues of the kernel matrix. *The Journal of Machine Learning Research*, 7:2303–2328, 2006.