TOWARDS PRIVACY AND COMMUNICATION EFFICIENCY IN DISTRIBUTED REPRESENTATION LEARNING

by

Sheikh Shams Azam

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

School of Electrical and Computer Engineering

West Lafayette, Indiana

August 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Christopher G. Brinton, Chair

School of Aeronautics and Astronautics

Dr. Shreyas Sundaram

School of Aeronautics and Astronautics

Dr. Qiang Qiu

School of Aeronautics and Astronautics

Approved by:

Dr. Dimitrios Peroulis

To my family and friends

ACKNOWLEDGMENTS

I would like to sincerely thank each member of my thesis committee – Dr. Christopher Brinton, Dr. Shreyas Sundaram, and Dr. Qiang Qiu – for their unwaivering support and guidance throughout my graduate studies. To Dr. Sundaram, your coursework and one-on-one discussions on various topics during my early graduate career has had a lasting impact on my research interests and philosophy. To Dr. Qiu, thank you for all the help and mentoring you provided during our research discussions. I learnt so much from you and always took away many intriguing insights from our discussions. Lastly, I would like to express my deepest gratitude to my advisor, Dr. Brinton, thank you for giving me the opportunity to work under your guidance duing my graduate career. It am sincerely thankful for your mentoring advice in general but also the countless hours you spent reading and reviewing my work. Your valuable feedback has been pivotal in shaping this work and has helped develop my research acumen. I would also like to thank you for creating a supportive environment where I got the opportunity to pursue works I was passionate about, and collaborate with other students and researchers. Finally, I would like to thank my parents for their endless support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CE	Cross-Entropy
D-EIGAN	Distributed Exclusion-Inclusion Generative Adversarial Network
D2D	Device-to-Device
DP	Differential Privacy
EIGAN	Exclusion-Inclusion Generative Adversarial Network
FCN	Fully Connected Network
FL	Federated Learning
GAN	Generative Adversarial Network
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
IoT	Internet-of-Things
KL	Kullback-Leibler
MIMIC	Medical Information Mart for Intensive Care
ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology
P2P	Peer-to-Peer
PCA	Principal Component Analysis
PRL	Private Representation Learning
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
TT-HF	Two Timescale Hybrid Federated learning

ABSTRACT

Over the past decade, distributed representation learning has emerged as a popular alternative to conventional centralized machine learning training. The increasing interest in distributed representation learning, specifically federated learning, can be attributed to its fundamental property that promotes data privacy and communication savings. While conventional ML encourages aggregating data at a central location (e.g., data centers), distributed representation learning advocates keeping data at the source and instead transmitting model parameters across the network. However, since the advent of deep learning, model sizes have become increasingly large often comprising million-billions of parameters, which leads to the problem of communication latency in the learning process. In this thesis, we propose to tackle the problem of communication latency in two different ways: (i) learning private representation of data to enable its sharing, and (ii) reducing the communication latency by minimizing the corresponding long-range communication requirements.

To tackle the former goal, we first start by studying the problem of learning representations that are private yet informative, i.e., providing information about intended “ally” targets while hiding sensitive “adversary” attributes. We propose Exclusion-Inclusion Generative Adversarial Network (EIGAN), a generalized private representation learning (PRL) architecture that accounts for multiple ally and adversary attributes, unlike existing PRL solutions. We then address the practical constraints of the distributed datasets by developing Distributed EIGAN (D-EIGAN), the first distributed PRL method that learns a private representation at each node without transmitting the source data. We theoretically analyze the behavior of adversaries under the optimal EIGAN and D-EIGAN encoders and the impact of dependencies among ally and adversary tasks on the optimization objective. Our experiments on various datasets demonstrate the advantages of EIGAN in terms of performance, robustness, and scalability. In particular, EIGAN outperforms the previous state-of-the-art by a significant accuracy margin (47% improvement), and D-EIGAN’s performance is consistently on par with EIGAN under different network settings.

We next tackle the latter objective – reducing the communication latency – and propose *two timescale hybrid federated learning* (`TT-HF`), a semi-decentralized learning architecture

that combines the conventional device-to-server communication paradigm for federated learning with device-to-device (D2D) communications for model training. In TT-HF, during each global aggregation interval, devices (i) perform multiple stochastic gradient descent iterations on their individual datasets, and (ii) aperiodically engage in consensus procedure of their model parameters through cooperative, distributed D2D communications within local clusters. With a new general definition of gradient diversity, we formally study the convergence behavior of TT-HF, resulting in new convergence bounds for distributed ML. We leverage our convergence bounds to develop an adaptive control algorithm that tunes the step size, D2D communication rounds, and global aggregation period of TT-HF over time to target a sublinear convergence rate of $\mathcal{O}(1/t)$ while minimizing network resource utilization. Our subsequent experiments demonstrate that TT-HF significantly outperforms the current art in federated learning in terms of model accuracy and/or network energy consumption in different scenarios where local device datasets exhibit statistical heterogeneity. Finally, our numerical evaluations demonstrate robustness against outages caused by fading channels, as well favorable performance with non-convex loss functions.

1. INTRODUCTION

Machine learning (ML) techniques have exhibited widespread success in applications ranging from computer vision [1], [2] to natural language processing [3]–[6]. Traditionally, ML model training has been conducted in a centralized manner, e.g., at data centers, where the computational infrastructure and dataset required for training coexist. In many applications (e.g., healthcare, Internet-of-Things (IoT)), however, the data required for model training is generated at devices which are distributed across the edge of communications networks. As the amount of data on each device increases, uplink transmission of local datasets to a main server becomes bandwidth-intensive and time consuming, which is prohibitive in latency-sensitive applications [7]. Latency problems could also arise because of the increasing size of neural networks used for these applications [8]–[11]. Common examples include object detection for autonomous vehicles [12] and keyboard next-word prediction on smartphones [13], each requiring rapid analysis of data generated from embedded sensors often using very large neural networks. Also, in many applications, end users may not be willing to share their datasets with a server due to privacy concerns [14], [15] (discussed further in Chapter 1.2).

1.1 Federated Learning

Federated learning (FL) [16] has emerged as a popular distributed ML technique for addressing these bandwidth [17] and privacy challenges [16], [18], [19]. A schematic of its conventional architecture is given in Fig. 1.1: in each iteration, each device trains a local model based on its own dataset, often using (stochastic) gradient descent. The devices then upload their local models to the server, which aggregates them into a global model, often using a weighted average, and synchronizes the devices with this new model to initiate the next round of local training.

Although widespread deployment of federated learning is desired [20], [21], its conventional architecture in Fig. 1.1 poses challenges for the wireless edge: the devices comprising the Internet of Things (IoT) may exhibit significant heterogeneity in their computational resources (e.g., a high-powered drone compared to a low-powered smartphone) [22]; additionally, the devices may exhibit varying proximity to the server (e.g., varying distances from

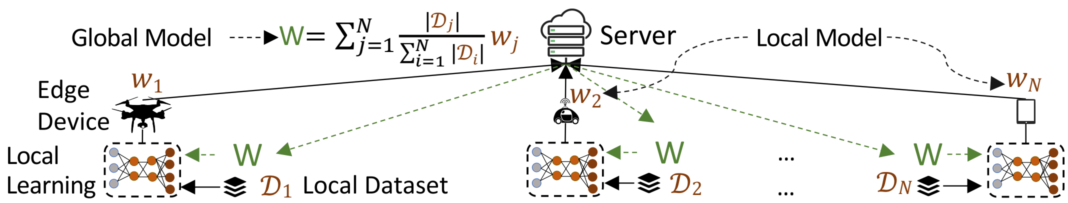

Global Model
$W = \sum_{j=1}^{N} \frac{|\mathcal{D}_j|}{\sum_{i=1}^{N} |\mathcal{D}_i|} w_j$
Server
Local Model
Edge Device
w_1
w_2
w_N
Local Learning
W
W
W
$\mathcal{D}_1$ Local Dataset
$\mathcal{D}_2$
$\mathcal{D}_N$
...
...

Figure 1.1. Conventional federated learning. In each training round, devices perform local model updates based on local datasets, followed by an aggregation at the main server to compute the global model, which is broadcast to the devices for the next round of local updates.

smartphones to the base station in a cell), which may cause significant energy consumption for upstream data transmission [23].

To mitigate the cost of uplink and downlink transmissions, local model training coupled with periodic but infrequent global aggregations has been proposed [22], [24], [25]. Yet, the local datasets may exhibit significant heterogeneity in their statistical distributions [26], resulting in learned models that may be biased towards local datasets, hence degrading the global model accuracy [24].

In this setting, motivated by the need to mitigate divergence across the local models, we study the problem of *resource-efficient federated learning across heterogeneous local datasets at the wireless edge.* A key technology that we incorporate into our approach is device-to-device (D2D) communications among edge devices, which is a localized version of peer-to-peer (P2P) among direct physical connections. D2D communications is being envisioned in fog computing and IoT systems through 5G wireless [7], [26], [27]; indeed, it is expected that 50% of all network connections will be machine-to-machine by 2023 [26]. Through D2D, we design a consensus mechanism to mitigate model divergence via low-power communications among nearby devices. We call our approach *two timescale hybrid federated learning* (`TT-HF`), since it (i) involves a hybrid between device-to-device and device-to-server communications, and (ii) incorporates two timescales for model training: iterations of stochastic gradient descent at individual devices, and rounds of cooperative D2D communications within clusters. By inducing consensus in the local models within a cluster of devices, `TT-HF` promises resource efficiency, as we will show both theoretically and by simulation, since only one device from the cluster needs to upload the *cluster model* to the server during global aggregation,

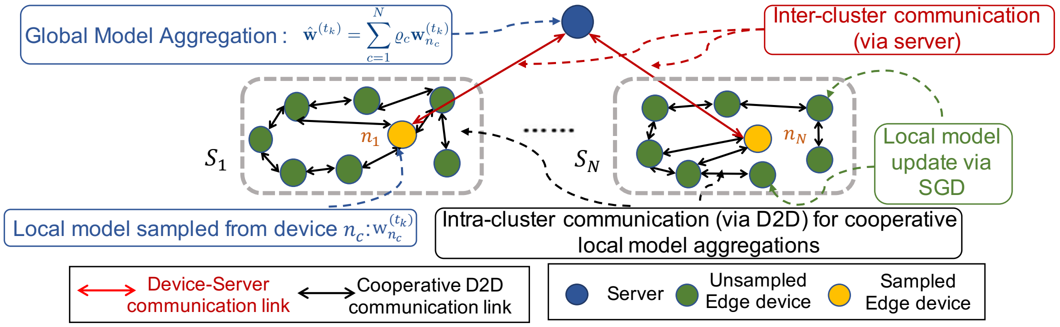

Global Model Aggregation : $\hat{\mathbf{w}}^{(t_k)} = \sum_{c=1}^{N} \varrho_c \mathbf{w}_{n_c}^{(t_k)}$
Inter-cluster communication (via server)
S_1
n_1
S_N
n_N
Local model update via SGD
Local model sampled from device n_c: $\mathrm{w}_{n_c}^{(t_k)}$
Intra-cluster communication (via D2D) for cooperative local model aggregations
Device-Server communication link
Cooperative D2D communication link
Server
Unsampled Edge device
Sampled Edge device

Figure 1.2. Network architecture of semi-decentralized federated learning. Edge devices form local cluster topologies based on their D2D communication capability. Cooperative local model aggregations among these clusters occur using D2D communications in between global aggregations conducted by the server.

as opposed to the conventional federated learning architecture, where most of the devices are required to upload their local models [18]. Specifically, during the local update interval in federated learning, devices can systematically share their model parameters with others in their neighborhood to form a distributed consensus among each cluster of edge devices. Then, at the end of each local training interval, assuming that each device's model now reflects the consensus of its cluster, the main server can randomly sample just one device from each cluster for the global aggregation. We call our approach *two timescale hybrid federated learning* (TT-HF), since it (i) involves a hybrid between device-to-device and device-to-server communications, and (ii) incorporates two timescales for model training: iterations of gradient descent at individual devices, and rounds of cooperative D2D communications within clusters.

TT-HF migrates from the "star" topology of conventional federated learning in Fig. 1.1 to a semi-decentralized learning architecture, shown in Fig. 1.2, that includes local topologies between edge devices, as advocated in the new "fog learning" paradigm [26]. In doing so, we must carefully consider the relationships between device-level stochastic gradient updates, cluster-level consensus procedure, and network-level global aggregations. We quantify these relationships in this work, and use them to tune the lengths of each local update and consensus period. As we will see, the result is a version of federated learning which optimizes the global model convergence characteristics while minimizing the uplink communication requirement in the system.

1.2 Privacy in Machine Learning

Training machine learning (ML) models often requires sharing data among multiple parties, e.g., cloud services aggregating data from multiple users to learn a global model. Such data sharing naturally raises concerns [28], [29] about exposing sensitive user attributes in datasets. It is thus imperative that both data aggregators and users engage in/propose procedures that minimize leakage of sensitive information.

A widely used technique for obfuscating sensitive attributes in data is context-agnostic noise injection (e.g. Laplace mechanism) [30], that introduces additive noise into a dataset to provide membership security [31]. However, noise injection can impact ML training and inference significantly [32]. This makes such context-agnostic techniques unsuitable in scenarios where only a few attributes need to be concealed. For example, upon sharing patient data for preventive healthcare [33], [34], both privacy (e.g., gender anonymization) and predictivity (e.g., accurate diagnosis) are desirable.

These drawbacks of context-agnostic privacy measures motivate private representation learning (PRL) [35], which exploits the knowledge of sensitive attributes in a dataset. PRL considers privacy and predictivity as joint (and possibly competing) objectives, and learns a transformation on the data that balances the goals of (i) obfuscating sensitive attributes of interest to an "adversary" while (ii) preserving predictivity on intended targets for an "ally" [36].

Conventionally, the literature on PRL assumes the existence of a single sensitive attribute and a central dataset [35], [37]–[39]. However, most real-world datasets have multiple sensitive attributes and are collected across multiple distributed nodes. Healthcare records, for example, are (i) spread across hospitals in different regions, (ii) consist of potentially multiple sensitive attributes, such as mental health, gender, ethnicity, etc., and (iii) may have varying notions of privacy that vary from one region to another, e.g., while in Europe racial/ethnic origin are considered as sensitive information (as per GDPR), in USA they are

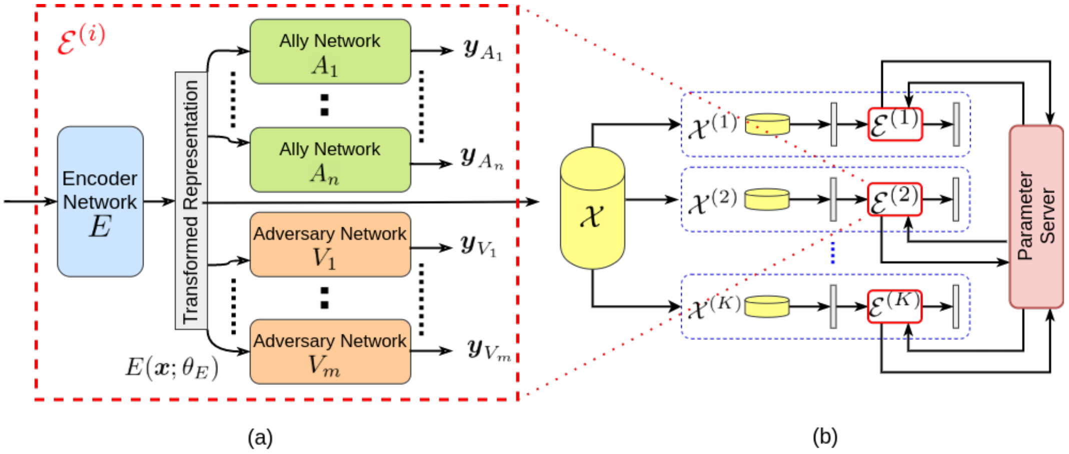


Figure 1.3. (a) Architecture of a single EIGAN node, consisting of an encoder, n ally, and m adversary networks. (b) D-EIGAN system for distributed EIGAN training, consisting of K different EIGAN nodes, each with their own subset of the full dataset. The nodes must coordinate their local encodings via a parameter server.

not (as per HIPAA). These challenges call for a *generalized and distributed* PRL methodology that takes into account multiple sensitive attributes, trains on data distributed across nodes, and learns representations that incorporate the privacy/predictivity goals of each node. Communication-efficiency is also a key objective in distributed learning, particularly when it is being deployed in network settings where nodes are restricted to communicate over limited-bandwidth links [40], [41], e.g., remote health analytics across user devices [42].

In this paper, we propose a novel PRL architecture called ***Exclusion-Inclusion Generative Adversarial Network (EIGAN)***, which addresses the aforementioned challenges. EIGAN is a generalized PRL technique designed to generate encodings "inclusive" of signals that are of utility to a set of allies, while "exclusive" of signals that can be used by adversaries to recover sensitive attributes. Further, to address the privacy vulnerabilities of pooling raw data, we develop ***D-EIGAN*** (for Distributed-EIGAN), where multiple EIGAN nodes train encoders on their local datasets and synchronize their model parameters periodically, as depicted in Fig. 1.3. D-EIGAN implements distributed training without noticeable model degradation compared to the centralized EIGAN, while accounting for realistic factors of communication constraints and non-i.i.d data distributions across nodes.

1.3 Related work.

1.3.1 Federated Learning

A multitude of works on federated learning have emerged in the past few years, addressing various aspects, such as communication and computation constraints of wireless devices [23], [43]–[45], multi-task learning [46]–[48], and personalized model training [49], [50]. We refer the reader to e.g., [51], [52] for comprehensive surveys of the federated learning literature; in Chapter 1.3.2 & 1.3.3, we further discuss the works addressing resource efficiency, statistical data heterogeneity, and cooperative learning.

1.3.2 Communication Efficiency

In terms of wireless communication efficiency, several works have investigated the impact of performing multiple rounds of local gradient updates in-between consecutive global aggregations [24], [53], including optimizing the aggregation period according to a total resource budget [24]. To further reduce the demand for global aggregations, [54] proposed a hierarchical system model for federated learning where edge servers are utilized for partial global aggregations.

Model quantization [55] and sparsification [56] techniques have also been proposed. As compared to above works, we propose a semi-decentralized architecture, where D2D communications are used to exchange model parameters among the nodes in conjunction with global aggregations. We show that our framework can reduce the frequency of global aggregations and result in network resource savings.

1.3.3 FL under Heterogeneous Data Distribution

Other works have considered improving model training in the presence of heterogeneous data among the devices via raw data sharing [22], [57]–[59]. In [57], the authors propose uploading portions of the local datasets to the server, which is then used to augment global model training. The works [22], [58], [59] mitigate local data heterogeneity by enabling the server to share a portion of its aggregated data among the devices [59], or by optimizing

D2D data offloading [22], [58]. However, raw data sharing may suffer from privacy concerns or bandwidth limitations. In our TT-HL framework, we exploit D2D communications to exchange model parameters among the devices, which alleviates such concerns.

Different from the above works discussed in Chapter 1.3.1, 1.3.2, & 1.3.3, we propose TT-HF that addresses the communication efficiency and data heterogeneity challenges simultaneously. To do this, we introduce distributed cooperative learning among devices into the local update process – as advocated recently [26] – resulting in a novel system architecture with D2D-augmented learning. In this regard, the most relevant work is [40], which also studies cluster-based consensus procedure between global aggregations in federated learning. Different from [40], we consider the case where (i) devices may conduct multiple (stochastic) gradient iterations between global aggregations, (ii) the global aggregations are aperiodic, and (iii) consensus procedure among the devices may occur aperiodically during each global aggregation. We further introduce a new metric of gradient diversity that extends the previous existing definition in literature. Doing so leads to a more complex system model, which we analyze to provide improvements to resource efficiency and model convergence. Consequently, the techniques used in the convergence analysis and the bound obtained differ significantly from [40]. There is also an emerging set of works on fully decentralized (server-less) federated learning [60]–[63]. However, these architectures require a well-connected communication graph among all the devices in the network, which may not be scalable to large-scale systems where devices from various regions/countries are involved in ML model training. Our work can be seen as intermediate between the star topology assumed in conventional federated learning and fully decentralized architectures, and constitutes a novel semi-decentralized learning architecture that mitigates the cost of resource intensive uplink communications of conventional server-based methods over star topologies, achieved via local low-power D2D communications, while improving scalability over fully decentralized server-less architectures.

Finally, note that there is a well-developed literature on consensus-based optimization, e.g., [64]–[67]. Our work employs the distributed average consensus technique and exploits that in a new semi-decentralized machine learning architecture and contributes new results on distributed ML to this literature.

1.3.4 Private Representation Learning

Most of the recent works in PRL [35], [37], [38], [68], [69] have only proposed a centralized architectures that jointly maximize the loss in predicting sensitive attributes while minimizing the loss of target task prediction. Specifically, [35] proposed a three-network encoder-ally-adversary architecture and showed that the achievable tradeoff between the two objectives is better than that provided by DP. In [37], the problem was formulated as a non-zero-sum game between the three networks to minimize information leakage in encoded image representations. [38] experimentally outperform [37], [70]–[72] using a minimax optimization among three networks, and derive its closed-form solution when the networks are linear maps. We demonstrate that EIGAN converges to the optimal performance obtained by these closed form solutions. However, unlike the closed form solution in [38], EIGAN can be extended to account for multiple ally and adversary attributes. Furthermore, EIGAN has computational advantage over [38] as it does not depend on matrix inversions, and thus can work with higher dimensional data.

Other PRL works take an information-theoretic approach. [39] view PRL as minimization of the utility lost in the learned representation, subject to an upper bound on mutual information between the output representation and the sensitive attribute. Similarly, [73] formulate the minimax problem in terms of KL-divergence. EIGAN, on the other hand, considers a cross-entropy PRL formulation, which promotes interpretability and training stability over multiple objectives (discussed in Chapter 2.2). Furthermore, our experiments show that EIGAN significantly outperforms the state-of-the-art [39] in the single ally/adversary case. *Distinct from all prior work in PRL, we consider multiple sensitive attributes and distributed learning.*

1.3.5 Fair Representation Learning & Synthetic Data Generation

There are two other related directions in adversarial learning. One addresses privacy-preservation through synthetic data generation [74], [75], which differs from EIGAN's goal of learning a transformation. The other is fair representation learning [76]–[78], which seeks to learn intrinsically fair representations that promote demographic parity on a single attribute [79].

1.4 Outline and Summary of Contributions

In this thesis, we present two different set of contributions in the domain of distributed representation learning. Firstly, in developing EIGAN and D-EIGAN for private representation learning, our main contributions are:

1. We introduce EIGAN (Chapter 2.2), generalizing PRL to account for multiple target and sensitive attributes. We prove that EIGAN's encoder utility is maximized if the adversary outputs follow a uniform distribution, and consider the effect of correlations between ally and adversary objectives (Proposition 1).

2. To the best of our knowledge, D-EIGAN (Chapter 2.3) is the first technique for distributed training of PRL models. We show that when the nodes engaged in the training possess independent and identically distributed (i.i.d) datasets, the objective of D-EIGAN exhibits similar properties to EIGAN (Proposition 3).

3. Our experiments (Chapter 2.4) reveal that EIGAN significantly outperforms the state-of-the-art in PRL (Table 2.1, Fig. 2.9) and is robust to the choice of adversary architectures (Table 2.3). We also demonstrate that D-EIGAN matches the performance of EIGAN even as the number of nodes increases (Fig. 2.13), and is robust even when nodes have different objectives (Fig. 2.15). We further show the resilience of D-EIGAN to non-i.i.d data distributions across nodes, and under communication restrictions that require partial parameter sharing and delayed model aggregations in the system (Fig. 2.17).

Next, in developing `TT-HF`, we summarize our main contributions as:

1. We propose *two timescale hybrid federated learning* (`TT-HF`), which augments the conventional federated learning architecture with aperiodic consensus procedure of models within local device clusters and aperiodic global aggregations by the server (Chapter 3.1).

2. We propose a new model of gradient diversity, and theoretically investigate the convergence behavior of `TT-HF` through techniques including coupled dynamic systems (Chapter 3.2). Our bounds quantify how properties of the ML model, device datasets, consensus process, and global aggregations impact the convergence speed of `TT-HF`. In doing so, we obtain a set of conditions under which `TT-HF` converges at a rate of $\mathcal{O}(1/t)$, similar to centralized stochastic gradient descent.

3. We develop an adaptive control algorithm for `TT-HF` that tunes the global aggregation intervals, the rounds of D2D performed by each cluster, and the learning rate over time to minimize a trade-off between energy consumption, delay, and model accuracy (Chapter 3.3). This control algorithm obtains the $\mathcal{O}(1/t)$ convergence rate by including our derived conditions as constraints in the optimization.

4. Our subsequent experiments on popular learning tasks (Chapter 3.4) verify that `TT-HF` outperforms federated learning with infrequent global aggregations, which is commonly used in literature, substantially in terms of resource consumption and/or training time over D2D-enabled wireless devices. They also confirm that the control algorithm is able to address resource limitations and data heterogeneity across devices by adapting the local and global aggregation periods.

2. EXCLUSION-INCLUSION GENERATIVE ADVERSARIAL NETWORK

©2021 NeurIPS. Reprinted with permission from S. S. Azam, T. Kim, S. Hosseinalipour, C. Joe-Wong, S. Bagchi, and C. G. Brinton, A Generalized and Distributable Generative Model for Private Representation Learning., In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications* [80].

2.1 Overview

Our PRL methodology consists of two phases: training and testing. In the training phase, EIGAN – knowing the sensitive/target labels of interest to adversary/ally on the train dataset – aims to learn the encoder by simulating allies and adversaries. Each of the allies, adversaries, and encoder independently maximize their own utilities by updating their local model parameters. The selfish maximization by each player naturally leads to the minimax optimization in (2.2). In the testing phase, the test data undergoes a transformation through the trained encoder. The transformed data is used for conventional training and inference by the actual allies and adversaries on their respective tasks of interest.

In Chapter 2.2, we present the EIGAN formulation for centralized model training, and derive properties of the solution. Then, we extend it to the distributed learning case, D-EIGAN, in Chapter 2.3.

2.2 EIGAN: Centralized Model Architecture

We first consider a system consisting of n allies, indexed $A_1, ..., A_n$; and m adversaries, indexed $V_1, ..., V_m$. Ally A_i is characterized by model parameters θ_{A_i} and a set of target attributes/labels Y_{A_i} drawn from distribution $\mathcal{Y}_{A_\mathrm{i}}$. A_i aims to associate each input sample with its corresponding target attribute in Y_{A_i}. Similarly, adversary V_j parameterized by θ_{V_j} wishes to associate input samples with a set of (known) sensitive attributes/labels Y_{V_j} following distribution $\mathcal{Y}_{V_\mathrm{j}}$.

The goal of EIGAN is to learn an encoder E parameterized by θ_E that maximizes the performance of $A_1, ..., A_n$ while minimizing the performance of $V_1, ..., V_m$. The encoder uses a centrally-located dataset $\mathcal{X}$ consisting of N samples, where each sample is represented as a d-dimensional feature vector $\boldsymbol{x}_{\mathrm{j}} \in \mathbb{R}^d$, $\mathrm{j} = 1, ..., N$. We let $E(\boldsymbol{x}; \theta_E)$ denote the output of the encoder for a data sample $\boldsymbol{x}$ realized via the parameters θ_E. $E(\boldsymbol{x}; \theta_E) : \mathbb{R}^d \to \mathbb{R}^l$ is in general a non-linear differentiable function (e.g., a neural network), where l is the dimension of the representation output by the encoder, and typically $l \leq d$.

For $\boldsymbol{x} \in \mathcal{X}$, the encoded representation $E(\boldsymbol{x}; \theta_E)$ is what the allies $A_1, .., A_n$ and adversaries $V_1, .., V_m$ are provided with for their tasks, as depicted in Fig. 1.3(a). We quantify the utilities of the allies and adversaries as:

$$\begin{aligned} u_{A_{\mathrm{i}}} &= \mathbb{E}_{Y \sim \mathcal{Y}_{A_{\mathrm{i}}}} \left[\log \left(p_{A_{\mathrm{i}}} \left(Y | E(\mathcal{X}; \theta_E) \right) \right) \right], 1 \leq \mathrm{i} \leq n, \\ u_{V_{\mathrm{j}}} &= \mathbb{E}_{Y \sim \mathcal{Y}_{V_{\mathrm{j}}}} \left[\log \left(p_{V_{\mathrm{j}}} \left(Y | E(\mathcal{X}; \theta_E) \right) \right) \right], 1 \leq \mathrm{j} \leq m, \end{aligned} \tag{2.1}$$

where $p_{A_{\mathrm{i}}} \left(Y | E(\mathcal{X}; \theta_E) \right)$ and $p_{V_{\mathrm{j}}} \left(Y | E(\mathcal{X}; \theta_E) \right)$ denote the probabilities of successful inference of target labels $Y \sim \mathcal{Y}_{A_{\mathrm{i}}}$ and sensitive labels $Y \sim \mathcal{Y}_{V_{\mathrm{j}}}$ for ally A_{i} and adversary V_{j}, respectively, over the outputs that the encoder E provides for the dataset $\mathcal{X}$. This leads to our minimax game among three types of players, in which two (the encoder and allies) are colluding against the third (the adversary). Specifically, we formulate the optimization problem:

$$\min_{\theta_V = \{\theta_{V_{\mathrm{j}}}\}_{\mathrm{j}=1}^m} \quad \max_{\theta_E, \theta_A = \{\theta_{A_{\mathrm{i}}}\}_{\mathrm{i}=1}^n} U(\theta_E, \theta_A, \theta_V), \tag{2.2}$$

where

$$U(\theta_E, \theta_A, \theta_V) = \sum_{\mathrm{i}=1}^{n} \alpha_{A_{\mathrm{i}}} u_{A_{\mathrm{i}}} - \sum_{\mathrm{j}=1}^{m} \alpha_{V_{\mathrm{j}}} u_{V_{\mathrm{j}}}. \tag{2.3}$$

Here, $\alpha_{A_{\mathrm{i}}}, \alpha_{V_{\mathrm{j}}} > 0$ denote normalized importance parameters placed on each objective such that $\sum_{\mathrm{i}=1}^{n} \alpha_{A_{\mathrm{i}}} + \sum_{\mathrm{j}=1}^{m} \alpha_{V_{\mathrm{j}}} = 1$. Similar to the encoder, we assume that the ally and adversary are non-linear, differentiable functions. The encoder in (2.2) seeks to maximize the achievable utility of the allies while minimizing those of the adversaries, operating in conjunction with the allies in the inner max layer of (2.2). The adversaries then operate on the encoder result in the outer min layer, where each adversary V_{j} aims to maximize its utility $u_{V_{\mathrm{j}}}$ by updating $\theta_{V_{\mathrm{j}}}$, as it cannot access other ally/adversary's parameters directly.

It is worth noting that, similar to the formulation based on mutual information in [39], our analysis on the expected posterior distribution of the predictions in EIGAN map directly to interpretable metrics such as accuracy [81] and generalization error [82], instead of the worst case guarantees provided by context-agnostic privacy frameworks such as DP.

Intuitively, the encoder will attempt to diminish the adversary predictions to a random guess, i.e., to a uniform distribution over its target labels [83]. However, this may be difficult to achieve when the interests of the allies and adversaries are related, which makes the weights $\alpha_{A_{\mathrm{i}}}, \alpha_{V_{\mathrm{j}}}$ important to the minimax solution in (2.2) formalized in the proposition below:

Proposition 1. *Let $\mathcal{O}$ denote the set of all (i, j) pairs of allies A_{i} and adversaries V_{j} for which $Y_{A_{\mathrm{i}}} \cap Y_{V_{\mathrm{j}}} \neq \emptyset$, i.e., overlapping interests. Given a fixed encoder E in EIGAN architecture, if $\mathcal{O} = \emptyset$, the overall score in (2.2) is maximized when the adversaries' output predictions follow a uniform distribution. On the other hand, if $\mathcal{O} \neq \emptyset$, then for each overlapping label, the architecture proposed by (2.2) considers the utility of the attributes that have the higher importance weight, i.e., A_{i} if $\alpha_{A_{\mathrm{i}}} > \alpha_{V_{\mathrm{j}}}$ and V_{j} if $\alpha_{A_{\mathrm{i}}} < \alpha_{V_{\mathrm{j}}}$.*

Proof. Suppose $\hat{Y}_{A_{\mathrm{i}}} = p_{A_{\mathrm{i}}}(Y|E(\mathcal{X}))$ and $\hat{Y}_{V_{\mathrm{j}}} = p_{V_{\mathrm{j}}}(Y|E(\mathcal{X}))$, where $p_{A_{\mathrm{i}}}(Y|E(\mathcal{X}))$ and $p_{V_{\mathrm{j}}}(Y|E(\mathcal{X}))$ denote the posterior probabilities of successful inference of target labels $Y \sim \mathcal{Y}_{A_{\mathrm{i}}}$ and sensitive labels $Y \sim \mathcal{Y}_{V_{\mathrm{j}}}$ for ally A_{i} and adversary V_{j}, respectively, given the outputs encoder E provides for the dataset $\mathcal{X}$. Then, the utilities in (2.1) can be expressed as

$$\begin{aligned} u_{A_{\mathrm{i}}} &= \mathbb{E}_{Y \sim \mathcal{Y}_{A_{\mathrm{i}}}}\left[\log \hat{Y}_{A_{\mathrm{i}}}\right], \\ u_{V_{\mathrm{j}}} &= \mathbb{E}_{Y \sim \mathcal{Y}_{V_{\mathrm{j}}}}\left[\log \hat{Y}_{V_{\mathrm{j}}}\right], \end{aligned} \tag{2.4}$$

where $1 \leq \mathrm{i} \leq n$ and $1 \leq \mathrm{j} \leq m$. Let $H_Q = \mathbb{H}(P, Q)$ denote the cross-entropy of Q with respect to P defined as $H_Q = \mathbb{H}(P, Q) = \mathbb{E}_{x \sim P}[-\log Q]$, then (2.4) can be re-stated as:

$$\begin{aligned} u_{A_{\mathrm{i}}} &= -H_{A_{\mathrm{i}}} = -\mathbb{H}(Y \sim \mathcal{Y}_{A_{\mathrm{i}}}, \hat{Y}_{A_{\mathrm{i}}}), \quad 1 \leq \mathrm{i} \leq n, \\ u_{V_{\mathrm{j}}} &= -H_{V_{\mathrm{j}}} = -\mathbb{H}(Y \sim \mathcal{Y}_{V_{\mathrm{j}}}, \hat{Y}_{V_{\mathrm{j}}}), \quad 1 \leq \mathrm{j} \leq m. \end{aligned} \tag{2.5}$$

The maximization of ally utilities u_{A_i} and minimization of adversary utilities u_{V_j} $\forall \mathrm{i}, \mathrm{j}$ in the optimization objective (2.3) can be re-written as minimization of its negative given by,

$$U' = -\sum_{\mathrm{i}=1}^{n} \alpha_{A_\mathrm{i}} u_{A_\mathrm{i}} + \sum_{\mathrm{j}=1}^{m} \alpha_{V_\mathrm{j}} u_{V_\mathrm{j}} = \sum_{\mathrm{i}=1}^{n} \alpha_{A_\mathrm{i}} H_{A_\mathrm{i}} - \sum_{\mathrm{j}=1}^{m} \alpha_{V_\mathrm{j}} H_{V_\mathrm{j}}. \tag{2.6}$$

Through (2.6), it can be observed that the minimization occurs when entropy of allies $\sum_{\mathrm{i}=1}^{n} \alpha_{A_\mathrm{i}} H_{A_\mathrm{i}}$ is minimized while that of adversaries $\sum_{\mathrm{j}=1}^{m} \alpha_{V_\mathrm{j}} H_{V_\mathrm{j}}$ is maximized. Using the definition of entropy, each of the allies and adversaries has a global optimum and can be optimized separately if their labels are non-overlapping. Note that ally and adversary entropies are non-negative, and given a fixed encoder E, the sum of ally entropies is minimized when individual entropies are minimized. For each ally, individual entropy H_{A_i} is minimized when $\hat{Y}_{A_\mathrm{i}}$ takes the value of 1 $\forall \mathrm{i}$ as every ally label is then predicted correctly. Similarly for adversaries, each individual entropy H_{V_j} is maximized when $\hat{Y}_{V_\mathrm{j}} = 1/|Y_{V_\mathrm{j}}|$ is the uniform distribution. Thus, it can be seen that, at the optimal solution, the adversaries' output follows a uniform distribution, as it minimizes the overall entropy in (2.6), or equivalently maximizes the utility in (2.3).

Given that $(A_\mathrm{i}, V_\mathrm{j}) \in \mathcal{O}$ is the set of all (i, j) pairs of allies A_i and adversaries V_j for which $Y_{A_\mathrm{i}} \cap Y_{V_\mathrm{j}} \neq \emptyset$, the ally and adversary objectives in (2.6) are overlapping if $\mathcal{O} \neq \emptyset$. Given that the encoder is fixed, for allies/adversaries not included in $\mathcal{O}$, the associated utilities can be independently optimized. We are thus left with the maximization of the following:

$$U_{\mathcal{O}} = \sum_{(A_\mathrm{i}, V_\mathrm{j}) \in \mathcal{O}} \alpha_{A_\mathrm{i}} \cdot u_{A_\mathrm{i}} - \alpha_{V_\mathrm{j}} \cdot u_{V_\mathrm{j}}. \tag{2.7}$$

For the kth element in $\mathcal{O}$, $(A_{\mathrm{i}(k)}, V_{\mathrm{j}(k)})$, we have $Y_{A_{\mathrm{i}(k)}}(c) = Y_{V_{\mathrm{j}(k)}}(c)$ $\forall c \in \mathcal{C}_k$ $\forall k$;, where $\mathcal{C}_k$ is the set of indices of elements in $Y_{A_{\mathrm{i}(k)}} \cap Y_{V_{\mathrm{j}(k)}} \neq \emptyset$. Separating the indices c for which the ally/adversary try to predict the same label (i.e. $u_{A_\mathrm{i}}(c) = u_{V_\mathrm{j}}(c)$), we can express (2.7) as follows:

$$U_{\mathcal{O}} = \sum_{k} \Bigg(\underbrace{\sum_{c \in \mathcal{C}_k} (\alpha_{A_\mathrm{i}} - \alpha_{V_\mathrm{j}}) u_{A_\mathrm{i}}(c)}_{\text{utility w.r.t. overlapping labels, } U_{\mathcal{O}+}} + \underbrace{\sum_{c \notin \mathcal{C}_k} \alpha_{A_\mathrm{i}} u_{A_\mathrm{i}}(c) - \alpha_{V_\mathrm{j}} u_{V_\mathrm{j}}(c)}_{\text{utility w.r.t. non-overlapping labels,} U_{\mathcal{O}-}} \Bigg). \tag{2.8}$$

The utilities in (2.8) reward only one of the two discriminators $(A_{\mathrm{i}}, V_{\mathrm{j}}) \in \mathcal{O}$ predicting on overlapping label $c \in \mathcal{C}$ if $\alpha_{A_{\mathrm{i}}} \neq \alpha_{V_{\mathrm{j}}}$. If $\alpha_{A_{\mathrm{i}}} = \alpha_{V_{\mathrm{j}}}$ for $(A_{\mathrm{i}}, V_{\mathrm{j}}) \in \mathcal{O}$, then $U_{\mathcal{O}+} = 0$, and no optimization occurs w.r.t. the overlapping labels in $Y \sim \mathcal{Y}_{A_{\mathrm{i}}}$. □

Proposition 1 shows that given an encoded representation, if the allies and adversaries possess non-overlapping interests, then a uniform prediction distribution among the sensitive parameters of interest to the adversaries is adopted by the optimal solution. In Section 2.4.1, we consider an experiment with such overlapping interests and equal importance weights, and find that EIGAN is unable to balance the objectives.

In practice, coincidental overlaps between ally and adversary interests would be relatively rare, but could nonetheless occur. In such cases, EIGAN must balance predictivity and privacy, which leads to different ally and adversary outputs described in Proposition 1. We further analyze EIGAN's characteristics when there is a linear relationship between the target distribution of an ally and an adversary:

Proposition 2. *Assume that the number of labels of interest is the same among all the allies and adversaries. For any adversary V_{j}, the distribution of its prediction over its set of labels of interest does not follow a uniform distribution if sufficient weight is given to the ally utilities (i.e., $\alpha_{A_{\mathrm{i}}}$, $\forall A_{\mathrm{i}}$, is sufficiently large) and the distribution of prediction of one ally A_{i}, can be defined as a linear combination of the distribution of predictions of V_{j} and that of other allies/adversaries.*

Proof. Without loss of generality, consider a system with one ally network with a scalar output $\hat{Y}_A$ and m adversary networks with scalar outputs $\hat{Y}_{V_{\mathrm{j}}}$ for $1 \leq \mathrm{j} \leq m$. The true distribution of each predicted output is $\mathcal{Y}_A$ for the ally and $\mathcal{Y}_{V_{\mathrm{j}}}$ for the adversaries, and Y_A and $Y_{V_{\mathrm{j}}}$ are the actual labels drawn from those distributions respectively. The true values and predictions between that of the ally and the adversaries have the relation, $Y_A = \sum_{\mathrm{j}=1}^{m} w_{\mathrm{j}} Y_{V_{\mathrm{j}}}$, and $\hat{Y}_A = \sum_{\mathrm{j}=1}^{m} w_{\mathrm{j}} \hat{Y}_{V_{\mathrm{j}}}$ where w_{j} is scaling weight. The cross entropy of the entire system is given by $U = \alpha_A Y_A \log(\hat{Y}_A) - \sum_{\mathrm{j}=1}^{m} \alpha_{V_{\mathrm{j}}} Y_{V_{\mathrm{j}}} \log(\hat{Y}_{V_{\mathrm{j}}})$. Optimizing for the output of a specific adversary V_n, we obtain:

$$\hat{Y}_{V_n} = \frac{\sum_{\mathrm{j} \neq n} w_{\mathrm{j}} \hat{Y}_{V_{\mathrm{j}}}}{\alpha_A Y_A w_n} \left(\frac{1}{\alpha_{V_n} Y_{V_n}} - \frac{1}{\alpha_A Y_A} \right)^{-1}. \tag{2.9}$$

Notably, $\hat{Y}_{V_n}$ only returns a non-uniform distribution when $\alpha_{V_n} Y_{V_n} < \alpha_A Y_A$. If the weight α_A is not large enough to maintain the inequality, the value of $\hat{Y}_{V_n}$ cannot be obtained via (2.9) and will have a uniform distribution. If $\alpha_{V_n} Y_{V_n} = \alpha_A Y_A$, then the cross entropy $U = 0$ and no optimization occurs. □

Model training. We train the encoder and the allies/adversaries in EIGAN by alternately updating their parameters using stochastic gradient descent (SGD) to minimize their cross-entropy (CE) loss. For the encoder, we define the CE-loss $\mathcal{L}_E$ for a single training instance as a weighted combination of the predictive capability of the allies and adversaries as

$$\mathcal{L}_E = \sum_{\mathrm{i}=1}^{n} \underbrace{-\langle \boldsymbol{y}_{A_\mathrm{i}}, \log \hat{\boldsymbol{y}}_{A_\mathrm{i}} \rangle}_{\text{loss of ally } A_\mathrm{i},\ \mathcal{L}_{A_\mathrm{i}}} - \alpha \cdot \sum_{\mathrm{j}=1}^{m} \underbrace{-\langle \boldsymbol{y}_{V_\mathrm{j}}, \log \hat{\boldsymbol{y}}_{V_\mathrm{j}} \rangle}_{\text{loss of adversary } V_\mathrm{j},\ \mathcal{L}_{V_\mathrm{j}}}, \tag{2.10}$$

where $\langle ., . \rangle$ denotes inner product, and log is applied element-wise. $\boldsymbol{y}_{A_\mathrm{i}}$ and $\boldsymbol{y}_{V_\mathrm{j}}$ are the binary vector representations of the true class labels for ally A_i and adversary V_j, respectively, while $\hat{\boldsymbol{y}}_{A_\mathrm{i}}$ and $\hat{\boldsymbol{y}}_{V_\mathrm{i}}$ are the vectors of soft predictions (i.e., probabilities) for each class. Here, we have made the simplifications $\alpha_{A_\mathrm{i}} = \alpha/n\ \forall\mathrm{i}$ and $\alpha_{V_\mathrm{j}} = (1-\alpha)/m\ \forall\mathrm{j}$, where $\alpha \in (0,1)$ is tuned to emphasize either predictivity (higher α) or privacy (lower α). It can be seen that the minimization of loss $\mathcal{L}_E$ is equivalent to the maximization of utility defined by (2.3). In each epoch, we average $\mathcal{L}_E$ over a minibatch of size J to obtain an estimate of (2.1), and update θ_E based on the gradient. Then, we update the θ_{A_i} and θ_{V_i} according to (2.10). See Algorithm 1 for further details.

Loss consideration. Alternative objectives to (2.10) exist in PRL literature. In particular, recent works [39], [73], [84] formulate the adversarial loss using KL divergence. We choose CE-loss over KL-divergence based on the fact that KL divergence fails to give meaningful value under disjoint distributions [85]. Also, our CE-loss formulation is unconstrained as opposed to KL-divergence formulation which is a Lagrangian dual of the constrained formulation [39]. As analyzed in Proposition 1, our formulation naturally pushes adversary prediction towards uniform distribution, however, the same does not hold for the constrained formulation. Our results in Table 2.1 and Fig. 2.9 (discussed later) validate our formula-

Algorithm 1 EIGAN training

```
Notations:
$(\cdot)_{\mathrm{j}}$ denotes the value for the jth minibatch
$\mathcal{L}_{A_{\mathrm{i}}}$ denotes the loss of ally $A_{\mathrm{i}}$
$\mathcal{L}_{V_{\mathrm{i}}}$ denotes the loss of the adversary $V_{\mathrm{i}}$
$\eta_{_E}, \eta_{_A}, \eta_{_V}$: learning rates of the encoders, allies and adversaries
Training:
initialize $\alpha$ used in loss function (2.10)
initialize $\theta_{A_{\mathrm{i}}}$'s and $\theta_{V_{\mathrm{j}}}$'s and $\theta_E$ to start the training
for number of training epochs do
    Sample a minibatch set $J$ of data points
    Compute encoder loss using (2.10): $\mathcal{L}_E = \frac{1}{|J|} \sum_{\mathrm{j} \in J} (\mathcal{L}_E)_{\mathrm{j}}$
    Update encoder parameters: $\theta_E \leftarrow \theta_E - \eta_{_E} \cdot \nabla_{\theta_E} \mathcal{L}_E$
    Compute allies/adversaries losses using (2.10):
        $$\mathcal{L}_{A_{\mathrm{i}}} = -\frac{1}{|J|} \sum_{\mathrm{j} \in J} (\mathcal{L}_{A_{\mathrm{i}}})_{\mathrm{j}}, \quad \mathcal{L}_{V_{\mathrm{i}}} = -\frac{1}{|J|} \sum_{\mathrm{j} \in J} (\mathcal{L}_{V_{\mathrm{i}}})_{\mathrm{j}}$$
    Update local allies/adversaries parameters:
        $$\theta_{A_{\mathrm{i}}} \leftarrow \theta_{A_{\mathrm{i}}} - \eta_{_A} \cdot \nabla_{\theta_{A_{\mathrm{i}}}} \mathcal{L}_{A_{\mathrm{i}}}, \quad \theta_{V_{\mathrm{i}}} \leftarrow \theta_{V_{\mathrm{i}}} - \eta_{_V} \cdot \nabla_{\theta_{V_{\mathrm{i}}}} \mathcal{L}_{V_{\mathrm{i}}}$$
end for
```

tion choice. We show consistent improvements over the state-of-the-art [39] that uses KL divergence.

2.2.1 Proof of Concept Visualizations

In this section, we demonstrate proof of concept visualizations using toy examples.

The first experiment uses a synthetic dataset comprising 4 sets of Gaussian distributed points in 2-D around the means (-0.5, -0.5), (-0.5, 1.5), (1.5, -1.5) and (1.5, 1.5) as shown in Fig. 2.1(a). We implement EIGAN with the ally objective to distinguish between reds and blues and adversary objective to segregate x's and o's. This is the simplest case we consider, as there is a single ally and single adversary, each with binary labels. Decision boundaries are linear. We thus use a logistic regression classifier as it has a convex loss function. The encoder is a neural network with a single hidden layer and output dimension $l = 2$. The learnt representation in Fig. 2.1(b) is intuitive: it maintains linear separability among ally classes, i.e., reds vs blues, but ensures a collapse of adversary classes.

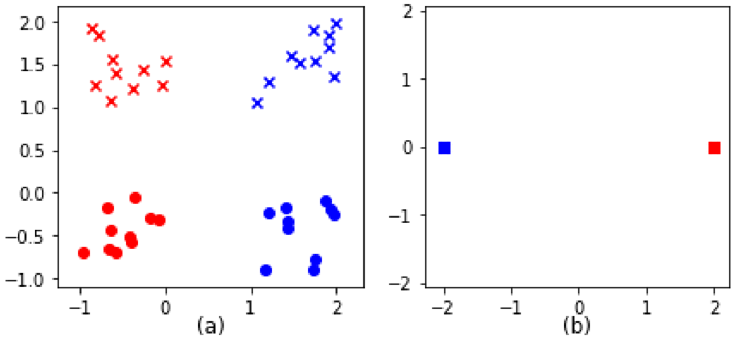


Figure 2.1. (a) Quadrant dataset with four groups of points, one ally, and one adversary. The points are linearly separable with regard to the ally's (classifying reds/blues) and an adversary's (classifying x's/o's) objectives. (b) EIGAN learns a representation that collapses the axes along the adversary's objective while enhancing separation along the ally's.

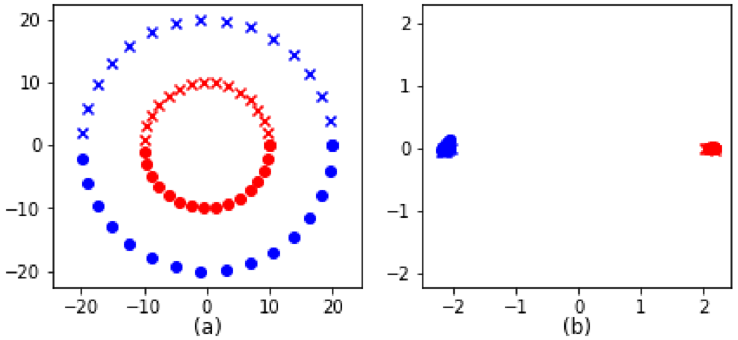
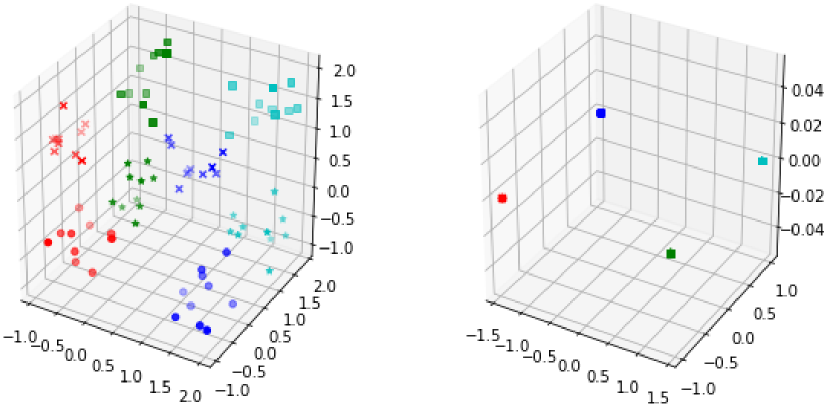


Figure 2.2. (a) Circle dataset with the same objectives as Figure 2.1 but ally classes (reds vs blues) are not linearly separable. (b) EIGAN learns a similar transformation, making the ally's classification task linearly separable.

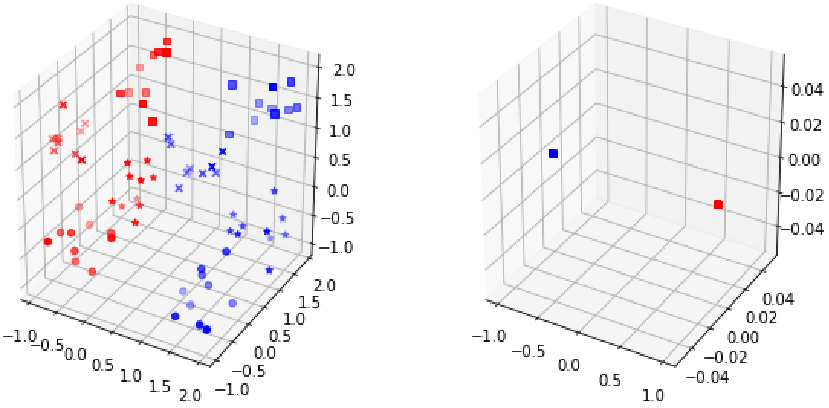
Next we consider a dataset with non-linear decision boundary as shown in Fig. 2.2 (a). The ally is interested in a decision boundary between the red and the blue circle, while the adversary is interested in the upper vs. lower semicircle, i.e., x's vs o's. The same encoder is used as in the previous experiment. We use a neural network with a single hidden layer as the ally and adversary because the ally's decision boundary is not linearly separable. Fig. 2.2(b) shows the learnt representation, which achieves a separability in the encoded space that is qualitatively similar to the representation learnt in Fig. 2.1(b).



(a)

(b)

Figure 2.3. (a) Synthetic dataset with eight groups of points, two allies, and one adversary. The allies are interested in separating the color pairs (the two horizontal axes), and the adversary is interested in classifying shapes (the vertical axis). (b) EIGAN's encoding has collapsed the adversary dimension while preserving the allies.



(a)

(b)

Figure 2.4. (a) Octant dataset with eight groups of points, one ally, and two adversaries. The ally is interested in classifying reds/blues while the adversaries are interested in separation along other axes. (b) EIGAN collapses the two adversary dimensions while maintaining separability for the ally.

We next extend EIGAN from single ally and single adversary to multiple allies and adversaries. We consider two different cases: EIGAN with (i) 2 ally and 1 adversary objective, and (ii) 1 ally and 2 adversaries. In the case (i), we have 8 set of Gaussian distributed points, one in each octant as shown in Fig. 2.3(a). There are two allies A_1 and A_2 which are each interested in separating data points along one of the horizontal axes, and an adversary V that is interested in separation along the vertical axis. We see in (b) that the EIGAN encoding collapses the data along the vertical axis while retaining separability in the other two dimensions. Similarly, in the case (ii), consider the same 8 set of Gaussian distributed points as in Fig. 2.3(a). However, in this case the ally wants to separate reds vs blues, and the adversaries want to separate along the other axes, i.e., top vs bottom (adversary 1) and squares and stars vs x's and o's (adversary 2). The learnt representation only preserves ally's dimension of variation, i.e. reds vs blues. All the other dimensions are collapsed.

2.3 D-EIGAN: Distributed Model Architecture

The distributed setting for EIGAN (D-EIGAN) is depicted in Fig. 1.3(b). There are K nodes in the system, denoted $\mathcal{E}^{(1)}, ..., \mathcal{E}^{(K)}$, and a parameter server for model synchronization. Each node $\mathcal{E}^{(k)}$ has a set of allies, denoted $A_1^{(k)}, ..., A_{n(k)}^{(k)}$ with target label sets $Y_{A^{(k)}} = \{Y_{A_1^{(k)}}, ..., Y_{A_{n(k)}^{(k)}}\}$, a set of adversaries, denoted $V_1^{(k)}, ..., V_{m(k)}^{(k)}$ with target sets $Y_{V^{(k)}} = \{Y_{V_1^{(k)}}, ..., Y_{V_{m(k)}^{(k)}}\}$, and a subset $\mathcal{X}_k \subset \mathcal{X}$ of N_k datapoints from the overall dataset $\mathcal{X}$ of N samples. These local datasets are in general non-overlapping, and may differ in size. While the specific allies and adversaries may differ at each node, the goal is to train encoder models that maximize all allies' and minimizes all adversaries' performances, so that the encodings are meaningful throughout the system. Since sharing the raw datasets could potentially leak sensitive information, each node $\mathcal{E}^{(k)}$ will train its own local encoder $E^{(k)}(\boldsymbol{x}; \theta_{E^{(k)}})$, and the server in Fig. 1.3(b) will periodically aggregate the locally-trained models.

The utility function for node $\mathcal{E}^{(k)}$ is defined as

$$U^{(k)}(\theta_{E^{(k)}}, \theta_{A^{(k)}}, \theta_{V^{(k)}}) = \sum_{\mathrm{i}=1}^{n} \alpha_{A_{\mathrm{i}}^{(k)}} u_{A_{\mathrm{i}}^{(k)}} - \sum_{\mathrm{j}=1}^{m} \alpha_{V_{\mathrm{j}}^{(k)}} u_{V_{\mathrm{j}}^{(k)}}, \tag{2.11}$$

where $\theta_{A^{(k)}} = \left\{\theta_{A_{\mathrm{i}}^{(k)}}\right\}_{\mathrm{i}=1}^{n(k)}$ and $\theta_{V^{(k)}} = \left\{\theta_{V_{\mathrm{j}}^{(k)}}\right\}_{\mathrm{j}=1}^{m(k)}$ denote the sets of ally and adversary parameters at node $\mathcal{E}^{(k)}$, and $u_{A_{\mathrm{i}}^{(k)}}, u_{V_{\mathrm{j}}^{(k)}}$ denote the utility functions of $A_{\mathrm{i}}^{(k)}, V_{\mathrm{j}}^{(k)}$ defined analogously to (2.1). $\alpha_{A_{\mathrm{i}}^{(k)}}, \alpha_{V_{\mathrm{j}}^{(k)}} > 0$ denote the normalized importance parameters for node

$\mathcal{E}^{(k)}$, where $\sum_{\mathrm{i}=1}^{n} \alpha_{A_{\mathrm{i}}^{(k)}} + \sum_{\mathrm{j}=1}^{m} \alpha_{V_{\mathrm{j}}^{(k)}} = 1$. This leads to the following minimax game for the distributed case:

$$\begin{aligned} &\min_{\mathcal{S}_V} \max_{\mathcal{S}_E, \mathcal{S}_A} \frac{1}{K} \sum_{k=1}^{K} U^{(k)}(\theta_{E^{(k)}}, \theta_{A^{(k)}}, \theta_{V^{(k)}}) \\ &\text{s.t.} \qquad \theta_{E^{(k)}} = \theta_{E^{(k')}},\ k \neq k', 1 \leq k, k' \leq K, \end{aligned} \tag{2.12}$$

where $\mathcal{S}_V = \{\theta_{V^{(k)}}\}_{k=1}^{K}, \mathcal{S}_E = \{\theta_{E^{(k)}}\}_{k=1}^{K}$, and $\mathcal{S}_A = \{\theta_{A^{(k)}}\}_{k=1}^{K}$. The constraint in (2.12) ensures that the optimal encoder is the same across all nodes, even though each node may have different allies and adversaries. In this way, an encoded datapoint $E^{(k)}(\boldsymbol{x}; \theta_E)$ at node k could be transferred to another node k' and applied to a task $A_{\mathrm{i}}^{(k')}$ privately, e.g., for anonymized user data sharing during single sign-ons.

Distributed model training. While solving (2.12) in a distributed manner, D-EIGAN learns both a global model and personalized local models (allies and adversaries) [46], unlike standard Federated Learning (FL).

Our algorithm consists of two iterative steps. The first is *local update*: each $\mathcal{E}^{(k)}$ conducts a series of δ SGD iterations. For each minibatch in SGD, training proceeds as in the centralized case, with the encoder, allies', and adversaries' parameters updated via SGD to minimize the CE-losses $\mathcal{L}_E^{(k)}$, $\mathcal{L}_{A_{\mathrm{i}}}^{(k)}$, and $\mathcal{L}_{V_{\mathrm{j}}}^{(k)}$ defined as in (2.10) but in this case for each node. The second step is *global aggregation*, in which each $\mathcal{E}^{(k)}$ uploads its locally-trained encoder to the parameter server to construct a global version, after every δ SGD iterations. We introduce a sparsification technique here in which each node selects a fraction ϕ of its parameters at random to upload for each aggregation. Letting $\mathcal{Q}_k$ be the indices chosen by $\mathcal{E}^{(k)}$, then the vector recovered at the server is $\tilde{\theta}_{E^{(k)}}$, where $\tilde{\theta}_{E^{(k)}}(q) = \theta_{E^{(k)}}(q)$ if $q \in \mathcal{Q}_k$ and 0 otherwise. With this, the global aggregation becomes the weighted average $\theta_E = \sum_k \frac{N_k}{N} \tilde{\theta}_{E^{(k)}}$. Then, the server also selects a fraction ϕ of indices at random to synchronize each node k with on the downlink. Letting $\mathcal{Q}$ be these indices, each node k sets $\theta_{E^{(k)}}(q) = \theta_E(q)$ if $q \in \mathcal{Q}$, and makes no change to the qth parameter otherwise. The pseudo-code of the training procedure is given in Algorithm 2.

The synchronization frequency δ and sparsification factor ϕ are directly related to the amount of data transferred through the system: as δ increases, uplink transfers to the server occur less frequently; as ϕ decreases, each uplink/downlink transmission requires fewer

Algorithm 2 D-EIGAN training

```
Notation:
$\theta_E$: global parameter vector
$\mathcal{Q}_k$: uniformly random choice of indices at node $\mathcal{E}^{(k)}$
$\tilde{\theta}_{E^{(k)}}$: parameter vector recovered at the server for encoder $E^{(k)}$, with its $q$th element
    denoted $\tilde{\theta}_{E^{(k)}}(q)$
$\phi$: fraction of parameters shared
$\delta$: number of epochs between aggregations
$(\cdot)_{\mathrm{j}}$: value for the jth minibatch
$\mathcal{L}_{A_{\mathrm{i}}^{(k)}}$ and $\mathcal{L}_{V_{\mathrm{i}}^{(k)}}$: loss of ally $A_{\mathrm{i}}^{(k)}$ and adversary $V_{\mathrm{i}}^{(k)}$
$\eta_E, \eta_A$, and $\eta_V$: learning rates
Aggregation at Parameter Server:
Initialize parameter $\theta_E$
for each update round do
    Update parameter vector: $\theta_E \leftarrow \sum_{k=1}^{K} \frac{N_k}{N} \tilde{\theta}_{E^{(k)}}$
end for
Local Training at Node $\mathcal{E}^{(k)}$:
Initialize $\left\{\theta_{A_{\mathrm{i}}^{(k)}}\right\}_{\mathrm{i}=1}^{n^{(k)}}$ and $\left\{\theta_{V_{\mathrm{j}}^{(k)}}\right\}_{\mathrm{j}=1}^{m^{(k)}}$
Download initial $\theta_E$ from parameter server
for number of training epochs do
    After $\delta$ epochs, update $\phi \cdot |\theta_{E^{(k)}}|$ chosen parameters
     from parameter server: $\theta_{E^{(k)}}(q) = \theta_E(q)$ if $q \in \mathcal{Q}$
    Sample a minibatch $J$ from local dataset $\mathcal{X}_k$
    Update encoder: $\theta_{E^{(k)}} \leftarrow \theta_{E^{(k)}} - \eta_E \cdot \nabla_{\theta_{E^{(k)}}} \mathcal{L}_{E^{(k)}}$
    Update ally/adversary parameters:
        $\theta_{A_{\mathrm{i}}^{(k)}} \leftarrow \theta_{A_{\mathrm{i}}^{(k)}} - \eta_A \cdot \nabla_{\theta_{A_{\mathrm{i}}^{(k)}}} \mathcal{L}_{A_{\mathrm{i}}^{(k)}}$,
        $\theta_{V_{\mathrm{i}}^{(k)}} \leftarrow \theta_{V_{\mathrm{i}}^{(k)}} - \eta_V \cdot \nabla_{\theta_{V_{\mathrm{i}}^{(k)}}} \mathcal{L}_{V_{\mathrm{i}}^{(k)}}$
    After $\delta$ epochs, upload $\phi|\theta_{E^{(k)}}|$ encoder parameters:
     $\tilde{\theta}_{E^{(k)}}(q) = \theta_{E^{(k)}}(q)$ if $q \in \mathcal{Q}_k$, else $\tilde{\theta}_{E^{(k)}}(q) = 0$
end for
```

communication resources. This is an important consideration in networking applications where the nodes communicate over a resource-constrained channel [24], [86]. Fractional parameter sharing, similar to pruning (both choose a subset of parameters), mimics the additive-noise DP mechanism [87] on model weights, reducing associated leakage [88], [89] to any untrusted entity with access to the system. We study the effect of δ and ϕ on D-EIGAN performance in Chapter 2.4.2.

In D-EIGAN, the allies and adversaries may differ at each node, and each node trains an individual local encoder. Since the encoder parameters are globally synchronized, however, the local encoder implicitly trains using global union of allies/adversaries across nodes. In the case that the nodes have same objectives and i.i.d. datasets, we show that D-EIGAN yields the same properties as Proposition 1:

Proposition 3. *Given a set of fixed encoders in the D-EIGAN architecture, if all the nodes have the same number of allies and adversaries with the same sets of target labels $Y_{A^{(k)}} = Y_{A^{(k')}}$ and $Y_{V^{(k)}} = Y_{V^{(k')}}$, $1 \leq k, k' \leq K$, then Proposition 1 holds for all the allies and adversaries belonging to different nodes if the local datasets at each node are i.i.d.*

Proof. Given that the global encoder is the average of the local encoders in the federated learning procedure for a single synchronization across K local nodes, the maximization of the expectation in (2.12) can be described as the maximization of ally utilities and minimization of adversary utilities given by:

$$U = \frac{1}{K} \sum_{k=1}^{K} \left(\sum_{\mathrm{i}=1}^{n_{(k)}} \alpha_{A_{\mathrm{i}}^{(k)}} u_{A_{\mathrm{i}}^{(k)}} - \sum_{\mathrm{j}=1}^{m_{(k)}} \alpha_{V_{\mathrm{j}}^{(k)}} u_{V_{\mathrm{j}}^{(k)}} \right). \tag{2.13}$$

In (2.13), $A_{\mathrm{i}}^{(k)}$ and $V_{\mathrm{i}}^{(k)}$ refer to the i$^{\text{th}}$ ally or adversary of the k^{th} local node. Since data at each node is i.i.d, the distributions $\mathcal{Y}$ are the same at each node, and thus each node has the same objective function. Using the result of Proposition 1 and assuming that $A_{\mathrm{i}}^{(k_1)}, V_{\mathrm{j}}^{(k_1)} = A_{\mathrm{i}}^{(k_2)}, V_{\mathrm{j}}^{(k_2)}$ $\forall \mathrm{i}, \mathrm{j}, k_1, k_2$ (i.e., the ally and adversary labels are same across all nodes), the output of the adversaries at each node follow a uniform distribution.

The ally and adversary objectives in (2.13) are overlapping if $\mathcal{O} \neq \emptyset$ given that $(A_{\mathrm{i}}, V_{\mathrm{j}}) \in \mathcal{O}$ is the set of all $A_{\mathrm{i}}, V_{\mathrm{j}}$ pairs for which $Y_{A_{\mathrm{i}}} = Y_{V_{\mathrm{j}}}$. Since each of the local nodes have the same overlapping ally/adversary labels with potentially different weights $\alpha_{A_{\mathrm{i}}^{(k)}}$ and $\alpha_{V_{\mathrm{j}}^{(k)}}$, their utilities can be expressed using entropy as in (2.5). The final optimization of the distributed system can be expressed as the minimization of following:

$$U_{\mathcal{O}} = \sum_{(A_{\mathrm{i}}, V_{\mathrm{j}}) \in \mathcal{O}} \left(\sum_{k=1}^{K} (\alpha_{A_{\mathrm{i}}^{(k)}} - \alpha_{V_{\mathrm{j}}^{(k)}}) \cdot u_{A_{\mathrm{i}}^{k}} \right). \tag{2.14}$$

The entropy values given in (2.14) reward only one of the two discriminators predicting label $Y_{A_{\mathrm{i}}}$ if $\sum_{k=1}^{K} \alpha_{A_{\mathrm{i}}^{(k)}} \neq \sum_{k=1}^{k} \alpha_{V_{\mathrm{i}}^{(k)}}$. If $\sum_{k=1}^{K} \alpha_{A_{\mathrm{i}}^{(k)}} = \sum_{k=1}^{K} \alpha_{V_{\mathrm{i}}^{(k)}}$, these two networks have no contribution to $U_{\mathcal{O}}$, and no optimization occurs. □

When the nodes have different objectives, we further show that the importance of each objective is proportional to the number of nodes implementing it:

Proposition 4. *If the allies and adversaries located at the K nodes of D-EIGAN have non-overlapping target sets, i.e., $Y_{A^{(k)}} \neq Y_{A^{(k')}}$ and $Y_{V^{(k)}} \neq Y_{V^{(k')}}$, $1 \leq k, k' \leq K$, then individual encoders under D-EIGAN consider the union of these local allies, $\bigcup_{k=1}^{K} Y_{A^{(k)}}$, and adversaries ,$\bigcup_{k=1}^{K} Y_{V^{(k)}}$ for optimization as a result of the global aggregation step. The weights $\alpha_{A_{\mathrm{i}}^{(k)}}$ and $\alpha_{V_{\mathrm{i}}^{(k)}}$ associated with the allies/adversaries are scaled by the ratio of the number of nodes that implement them locally to the total number of nodes.*

Proof. Without loss of generality, consider a two network D-EIGAN. Let node 1 have 2 allies and 1 adversary with objectives: Y_{A_c}, Y_{A_1}, and Y_{V_1}, and node 2 have 2 allies and 1 adversary with objectives: Y_{A_c}, Y_{A_2} and Y_{V_2}. Here, objective Y_{A_c} is common among them, while the rest are different. Utilities of individual nodes can be calculated using (2.3):

$$U^{(1)} = \alpha_{A_c} \cdot u_{A_c} + \alpha_{A_1} \cdot u_{A_1} - \alpha_{V_1} \cdot u_{V_1}, \tag{2.15}$$

$$U^{(2)} = \alpha_{A_c} \cdot u_{A_c} + \alpha_{A_2} \cdot u_{A_2} - \alpha_{V_2} \cdot u_{V_2}. \tag{2.16}$$

Under federated training, the equivalent loss function that is optimized by the D-EIGAN can be calculated using (2.12):

$$U = \alpha_{A_c} \cdot u_{A_c} + \frac{\alpha_{A_1}}{2} \cdot u_{A_1} - \frac{\alpha_{V_1}}{2} \cdot u_{V_1} + \frac{\alpha_{A_2}}{2} \cdot u_{A_2} - \frac{\alpha_{V_2}}{2} \cdot u_{V_2}, \tag{2.17}$$

which shows that the overall objective under D-EIGAN considers all the objectives, but the associated weights are lower for non-common allies/adversaries. In contrast to a D-EIGAN where all allies and adversaries are common across nodes, the difference is the weights associated with objectives. □

2.4 Experimental Evaluation and Discussion

We now turn to an experimental evaluation of our methodology. We analyze EIGAN's convergence characteristics and compare its performance with relevant baselines in Chapter 2.4.1, and evaluate D-EIGAN compared to the centralized case and as the system characteristics change in Chapter 2.4.2.

Datasets. We consider datasets: MNIST [90], MIMIC-III [91], Adult [92], and FaceScrub [93]. MNIST consists of 60,000 handwritten digits with labels 0-9. MIMIC has medical information from hospitals with attributes, such as vitals and medication; we obtain a dataset consisting of 58,976 patients by joining multiple tables on patient IDs. Adult consists of 45,223 records extracted from the 1994 census data. Facescrub is a dataset comprising over 22,000 images of celebrities with identity and gender labels.

Objectives. In MIMIC, we consider survival (2-class) as the ally objective, and gender (2-class) and race (3-class) as adversary objectives. In the FaceScrub dataset, as in [39], the ally objective is user identity (200-class), and the adversary objective is gender (2-class). In MNIST, we consider whether a digit is even or odd (2-class) as the ally objective, and the label of the digit (10-class) as the adversary objective. In Adult, as in [38], the ally objective is an annual income classification (more or less than 50K) and the adversary objective is gender. We also generate synthetic Gaussian datasets to analyze the effect of ally/adversary class overlap in some experiments.

Implementation. We use fully connected networks (FCNs) for the encoder, allies, and adversaries in the experiments on MIMIC and the synthetic datasets. The FCN encoder uses ReLU [94] activation for the hidden layers and tanh activation for the final fully-connected layer, whereas the ally and adversaries use sigmoid activation in the final layer. We use dropout [95] and L2-regularization to prevent network overfitting. For FaceScrub, we employ U-Net [96] for the encoder and Xception-Net [97] for the ally/adversary as in [39]. For Adult, we employ linear FCN as in [38]. Unless otherwise stated, we set $\alpha = 0.5$ (i.e., equal privacy/predictivity importance). We train to minimize CE loss over 70/30 training/test splits on a system with 8 GB GPU and 64 GB RAM.

Table 2.1. Performance comparison between EIGAN, [38] (Linear-ARL, Kernel-ARL), and [39] (Bertran-PRL) on the Adult & FaceScrub datasets considered in those works. For the same adversary performance, EIGAN obtains a notable improvement over [39] (ally improvement of 47.01%). It also reaches the optimal closed form solution of [38].

	Adult Dataset		Facescrub Dataset	
Objective	**Ally** (identity)	**Adversary** (gender)	**Ally** (income)	**Adversary** (gender)
Unencoded	0.85	0.85	0.98	0.99
Linear-ARL	**0.84**	0.67	-	-
Kernel-ARL	**0.84**	0.67	-	-
Bertran-PRL	0.82	0.67	**0.56**	0.68
EIGAN	**0.84**	0.67	**0.82**	0.68
% Improv.	Matches closed form solution	Controlled to be equal	47.01%	Controlled to be equal

Baselines. We consider six baselines: principal component analysis (PCA) [98], autoencoders [99], differential privacy (DP) in the form of Laplace Mechanism as in [35], and the methods in [38], [39]. Autoencoders and PCA preserve information content and do not have explicit privacy objectives; they are expected to give encoded data that has good predictivity. PCA chooses the number of components retaining 99% of the variance, and we train the autoencoder to transform data to the same dimensional space as PCA. As discussed in Chapter 1.2, DP is widely used for context-agnostic privacy. For DP, we employ the Laplace mechanism [35]. [39] is the most recent state-of-the-art in adversarial PRL; in this case, we use their open-source implementation and compare on the setting described in their paper. We also compare against the closed form optimal solution of [38] for linear maps on their Adult dataset use case, where [38] outperforms [37], [70]–[72].

All of our code using PyTorch [100] and trained models are available at https://github.com/shams-sam/PrivacyGANs. For each experiments, we report cross-entropy loss and/or accuracy from the testing step of PRL.

2.4.1 Centralized EIGAN

Performance comparison with prior works

We first compare EIGAN with [38] and [39] on the Adult and Facescrub dataset settings considered in these works, respectively. Note that the linearity requirement in [38] impedes its usage on non-linear models like the U-Net and Xception-Net employed for Facescrub by [39]. For comparison, we adjust α in (2.10) to equalize the resulting adversary performances between the models. Table 2.1 gives the results: EIGAN matches the performance of [38]'s optimal closed-form solution on Adult. On the Facescrub dataset, it displays a 47% improvement in the ally's task of identity recognition when compared to [39]. This validates our choice of optimization using cross-entropy loss in (2.10) for PRL over the technique of optimization using KL divergence that is common in recent PRL literature [39], [73], [84].

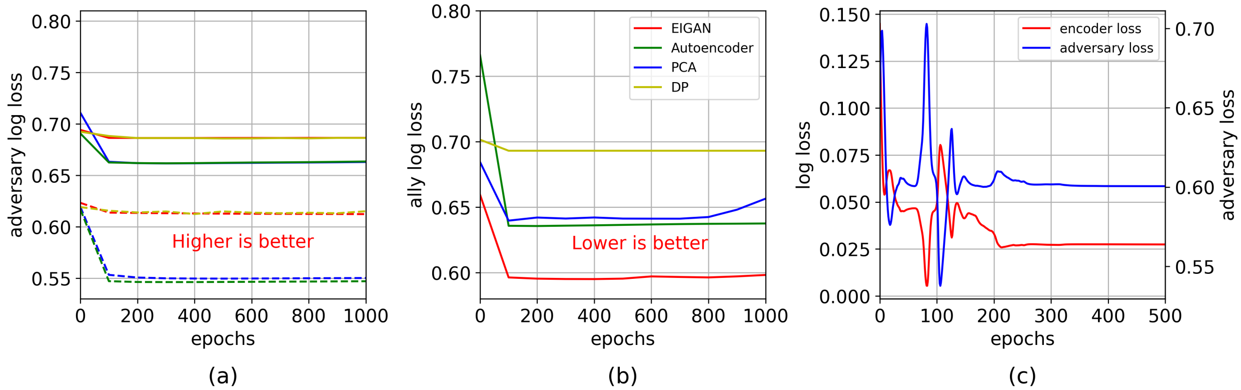


Figure 2.5. Predictivity and privacy comparison between EIGAN and the baselines across one ally and two adversaries on the MIMIC-III dataset. (a) On the adversary objectives (gender prediction, solid lines and race prediction, dashed lines) EIGAN matches DP's performance (by design of the experiment, as determined by the selection of the DP ϵ parameter). Hence, the red and the khaki colored curves overlap. (b) On the ally objective (survival prediction), EIGAN achieves noticeable improvement over the baselines. (c) EIGAN training converges after initial oscillations corresponding to the minimax game.

Comparison on MIMIC dataset.

We next compare the ally and adversary losses over training epochs between EIGAN, autoencoder, PCA, and DP on the MIMIC dataset in Fig. 2.5. Note that the recent baselines [38], [39] cannot handle multiple adversary objectives. It is observed in (a) that EIGAN is able to match the adversary losses of DP, while in (b) the EIGAN ally loss matches that of PCA and autoencoder while outperforming DP by a significant margin. Thus, EIGAN is capable of achieving private representations while simultaneously maintaining the predictivity of the encoded representations. Also, (c) shows the loss progression of encoder and adversary as the EIGAN training proceeds. It can be observed that increase/decrease in encoder loss is corresponding to the decrease/increase in adversary loss during the same epoch, consistent with the definition of the encoder loss in (2.10). The magnitude of the oscillations decreases as we progress through the training and eventually the networks (i.e., the players in the game) reach a steady state.

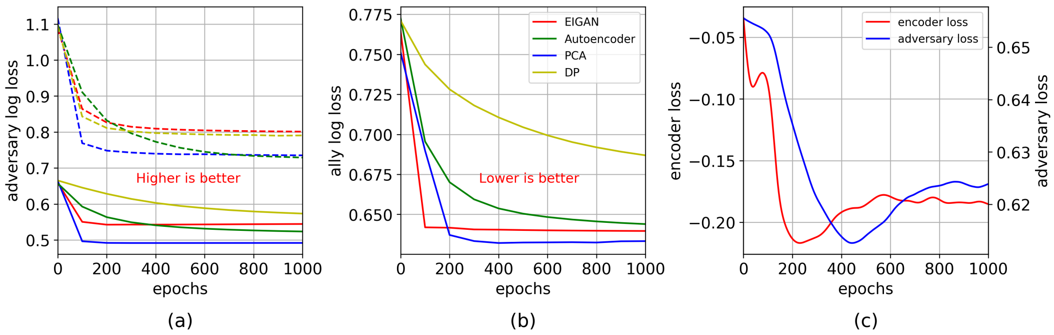


Figure 2.6. Predictivity and privacy comparison between EIGAN and the baselines across one ally and two adversaries on the Titanic dataset. (a) On one of the adversary objectives (gender prediction, solid lines) EIGAN matches DP's performance (by design of the experiment, as determined by the selection of the DP ϵ parameter), but in this case it does not match the other adversary prediction (passenger class prediction, dashed lines), which could be matched for another value of ϵ. (b) On the ally objective (survival prediction), EIGAN achieves marginal improvement over the the baseline Autoencoder. (c) EIGAN training converges after initial oscillations corresponding to the minimax game.

Comparison on Titanic dataset

For completeness, we also evaluate EIGAN algorithm on another dataset, Titanic, which consists of data listing the details of roughly 800 of the passengers that were onboard the Titanic ship. This experiment aims at understanding the convergence behaviour of EIGAN under limited training data.

Table 2.2. Comparison of log-loss achieved on the test set between the algorithms for the Titanic dataset. EIGAN matches autoencoder on the ally and performs slightly better than DP on adversary 2, while slightly worse on adversary 1.

Algorithm	**Ally**	**Adversary 1**	**Adversary 2**
	(Survival)	(Gender)	(P-Class)
Autoencoder	**0.6333**	0.4918	0.7351
PCA	0.6439	0.5236	0.7289
DP	0.6869	**0.5733**	0.7904
EIGAN	**0.6396**	0.5444	**0.8011**

Similar to result on MIMIC-III from Fig. 2.5, Fig. 2.6 (a) shows that while EIGAN is able to perform as well or nearly as well as any of the baselines on adversary obfuscation, (b) it obtains the best predictivity on ally objective. (c) shows that the training reaches a steady-state.

Table 2.2 summarizes the loss-values of the trained allies/adversaries on encoded data using different techniques. It can be seen that while EIGAN is able to match DP's performance on adversary 2, it performs marginally worse than it on adversary 1, while having a considerable gain on the corresponding ally.

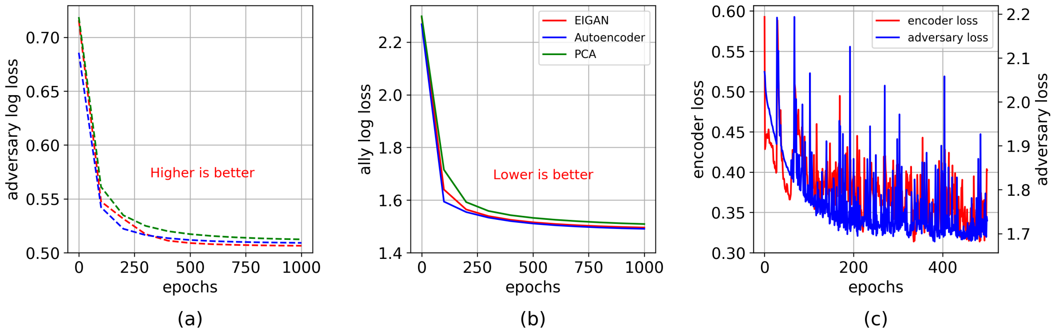


Figure 2.7. Comparison across one ally and two adversaries on the MNIST dataset. The (a) adversary objective (odd-even prediction, a binary classification with virtually identical trends) converge to roughly the same loss for each algorithm, and (b) ally objective (digit prediction, 10-class classification). With dependencies (in particular, partial overlaps) between the ally and adversary objectives, EIGAN training in (c) is unable to fully converge, consistent with Proposition 2.

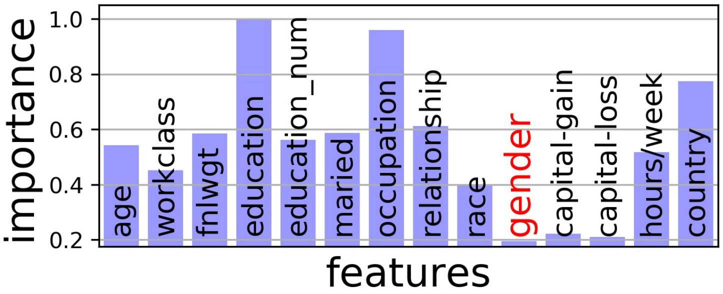


Figure 2.8. Feature importance derived from EIGAN encoder on Adult dataset results summarized in Table 2.1.

Comparison on MNIST

We conduct an additional experiment on the MNIST dataset of handwritten digits to validate the findings in Proposition 1&2 when dependencies exist between the ally and adversary objectives. In this case, we use digit recognition (0-9) as the ally objective and even vs odd as adversary objective, which exhibits a clear dependence because if someone could recover the digit (ally objective), then inferring odd-vs-even (adversary objective) becomes trivial. Formally, referring to the propositions, we have $\mathcal{Y}_{\text{odd}} = \mathcal{Y}_1 + \mathcal{Y}_3 + \cdots + \mathcal{Y}_9$ and $\mathcal{Y}_{\text{even}} = 1 - \mathcal{Y}_{\text{odd}}$ where $\mathcal{Y}_{(\cdot)}$ is the true probability distribution on the labels and thus can be added. Similarly, $\hat{Y}_{\text{odd}} = \hat{Y}_1 + \hat{Y}_3 + \cdots + \hat{Y}_9$ and $\hat{Y}_{\text{even}} = 1 - \hat{Y}_{\text{odd}}$, where $\hat{Y}_{(\cdot)}$ are probabilities of correct predictions. Proposition 2 follows when we substitute these in (2.9), i.e. the adversary is not forced to a follow uniform distribution if sufficient weight is given to the ally.

Fig. 2.7 shows the result of this experiment, where the weights of the allies and adversaries are set equal. (a) shows that the adversary is not able to achieve any separation from the Autoencoder or PCA. Observing (c), we realize that the training process does not reach a steady state-convergence point, consistent with the propositions.

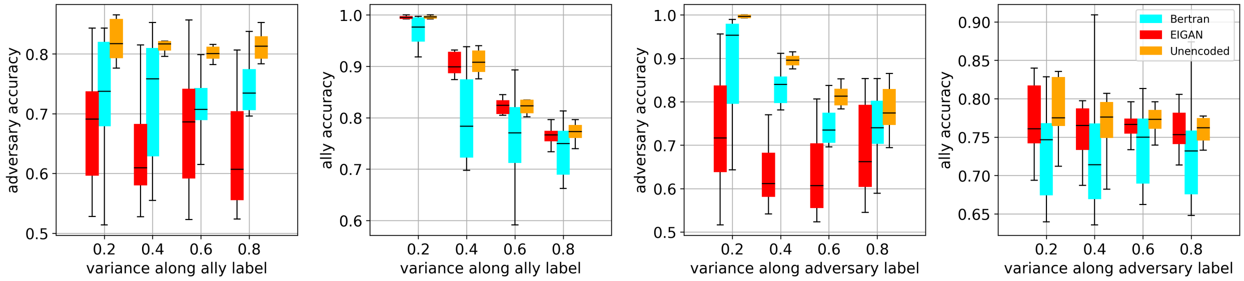
Robustness of learned representation

In Fig. 2.8 we consider the importance placed by EIGAN encoder on input features of Adult dataset for learning the private representations. It can be observed that the importance of gender and it's correlated features is very low. This implies that the learnt representations

Table 2.3. Accuracy of various architectures used to infer ally (even/ odd) and adversary (digits 0-9) objectives on MNIST encoded using ResNet152-trained EIGAN. We see that the ally accuracies are consistent across network architectures, and the adversary accuracies remain significantly below the performance on the unencoded data.

Model	**Ally** (accuracy)	**Adversary** (accuracy)
Resnet152 Unencoded	0.99	0.99
Resnet152	0.85	0.45
ResNext101	0.86	0.42
Resnet101	0.88	0.64
Resnet50	0.87	0.56
WideResnet101	0.85	0.42
VGG19	0.77	0.42

minimize the signals w.r.t adversary's interest, i.e., gender. We next consider the robustness of EIGAN's learned representation to ally and adversary architectures that deviate from the one used for training. Table 2.3 shows the performance of varying architectures (ResNet [9], ResNext [101], etc.) for allies and adversaries applied to the data encoded using EIGAN trained with ResNet152 adversary on MNIST. We see that the representations learned by EIGAN are able to obfuscate adversary targets from the other networks. Adversary accuracy remains significantly below the performance on the unencoded data, validating the robustness to differences between simulated and actual adversaries.



(a) (b) (c) (d)

Figure 2.9. Effect of change in ally (a-b) and adversary (c-d) overlap (by changing the variances of synthetic Gaussian data) on the performance of EIGAN, [39], and the unencoded data. EIGAN is able to consistently outperform both baselines on the adversary objective, and obtains performance close to the unencoded data for the ally.

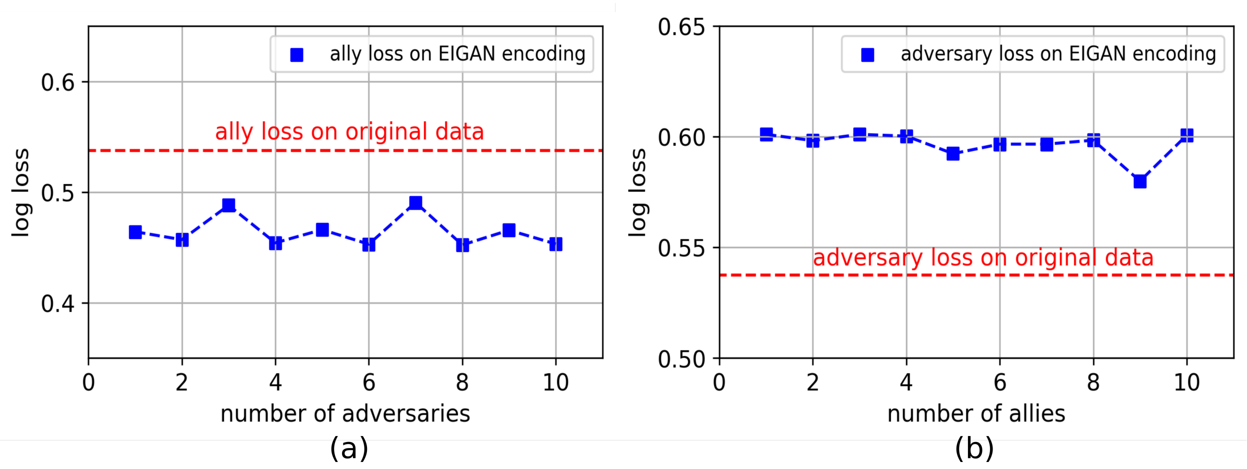


Figure 2.10. EIGAN's effect of the number of (a) adversaries, and (b) allies on the testing loss for MIMIC-III. The ally/adversary objectives are chosen as different attributes from the source. The achievable loss is reasonably constant and is not affected by addition of more allies/adversaries.

Varying ally/adversary overlap

Next, we consider the effect of class overlap for the ally and adversary objectives on model performance. To do this, we generate a 2D dataset consisting of four Gaussians with means at $(x, y) = (1, 1), (1, 2), (2, 1), (2, 2)$, each corresponding to one class. The variance of these Gaussian-distributed classes is adjusted to achieve varying degrees of overlap. Fig. 2.13(c) shows an instance of this dataset: the ally is interested in differentiating color, while the adversary wants to differentiate shape. Fig. 2.9 shows the effect of the ally and adversary label variance on the resulting accuracies for EIGAN, the method in [39], and the unencoded data. As the ally variance increases, we observe that (a) the accuracy of the adversary for EIGAN remains consistently lower than that of the others, while (b) the accuracy on the ally objective for EIGAN remains higher than that of [39] and is comparable to the unencoded case. Similarly, EIGAN outperforms [39] consistently as the adversary exhibits more variance: (c) the accuracy of adversary for EIGAN is lower than others while (d) the corresponding accuracy of ally for EIGAN is higher and close to unencoded case. The p-values of the improvements EIGAN makes over the method in [39] are below 0.002 in all 16 cases of comparisons between boxplot distributions.

Effect of varying number of ally/adversary in EIGAN

We also consider the impact of the number of allies/adversaries on EIGAN's performance using MIMIC. We observe (in Fig. 2.10) that the final test loss obtained by an adversary (ally) under varying number of allies (adversaries) stays reasonably constant. Thus, encodings are robust to the number of objectives that are included in EIGAN.

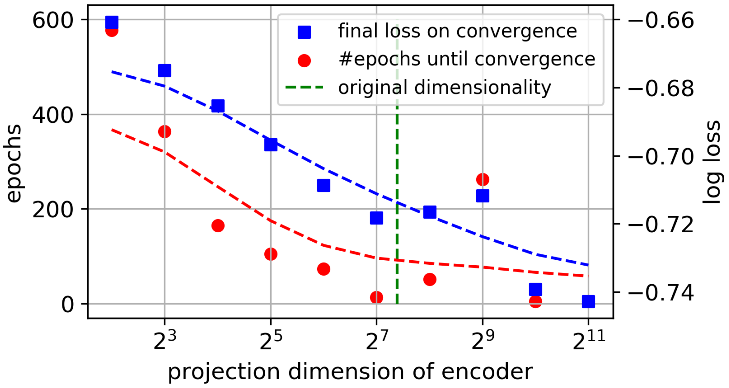


Figure 2.11. Effect of EIGAN's encoding dimension space on the number of training epochs required to reach within 1% of training loss convergence (left axis) and the achieved final testing loss (right axis) for MIMIC-III. The achieved loss decreases sharply as the dimension increases, emphasizing a trade-off between model quality and the memory needed for the encoded data. In fact, beyond the right end of the X-axis value, the model runs out of memory on our high performance machine. (Dashed curves are fit using weighted moving averages.)

Varying Encoder Dimensionality

Fig. 2.11 depicts the results for the MIMIC-III dataset while Fig. 2.12 depicts the result of a similar experiment on Titanic dataset. In the two experiments, as the encoder output dimension l is increased, we observe that the training mostly requires fewer epochs to converge and is able to achieve a lower encoder testing loss. This could be explained by the fact that larger networks (i.e. more number of trainable parameters) have more degrees of freedom in training. Interestingly, while there is some variation, the test loss continues to decrease beyond d, the original dimension of the data samples, i.e., when $l \geq d$. The relevant consideration with EIGAN, then, appears to be the tradeoff between encoding quality,

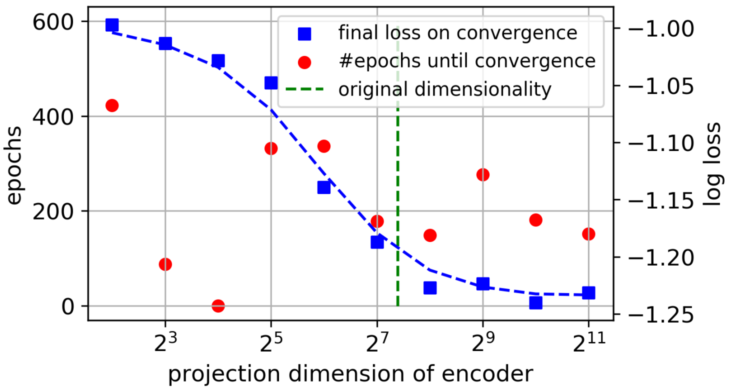


Figure 2.12. Effect of EIGAN's encoding dimension space on the number of training epochs required to reach within 1% of training loss convergence (left axis) and the achieved final testing loss (right axis) for the Titanic dataset. The achieved loss decreases sharply as the dimension increases, emphasizing a tradeoff between model quality and required memory. (The dashed curve is fit using a weighted moving average.)

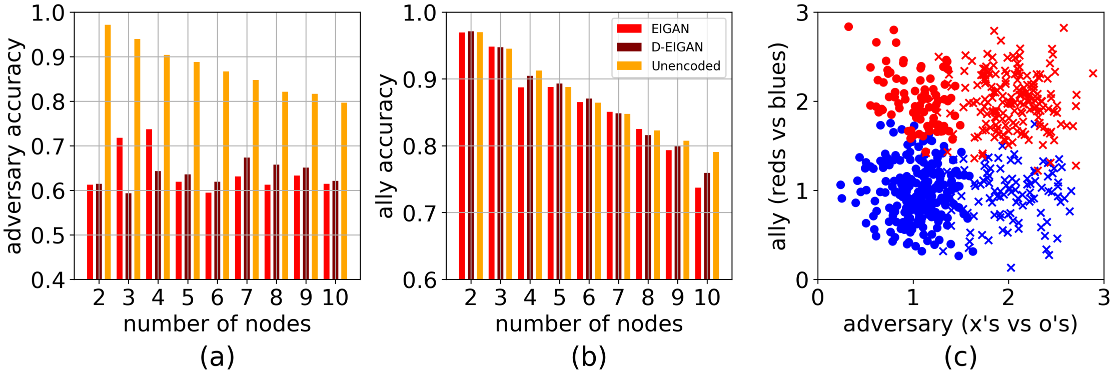


Figure 2.13. Comparison of (a) adversary and (b) ally performance as the number of nodes in the system is increased from $K = 2$ to 10, for D-EIGAN $(\phi, \delta = 1)$, EIGAN, and unencoded. Node k's data, $k = 1, ..., K$ is generated from four Gaussians centered on a unit square, each with $\sigma^2 = 0.1k$, i.e. increasing variance. (c) visualizes the ally (reds vs. blues) and adversary (x's vs. o's) objectives for node $k = 3$. As expected, the ally performs worse with higher K, but D-EIGAN is able to match EIGAN's performance.

as measured by the encoding space dimension, and the memory required for training the encoder, which increases with the dimension of the encoder.

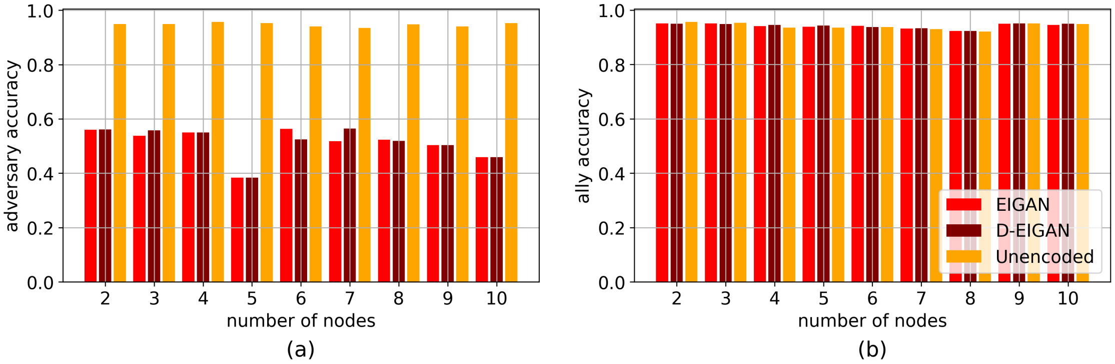


Figure 2.14. Comparison of (a) adversary and (b) ally performance using synthetic Gaussian data while increasing the number of nodes and sharing all the model weights ($\phi = 1$) after every minibatch ($\delta = 1$) during federated training. The distribution of data is i.i.d. across the nodes, which is obtained by generating Gaussian data with constant mean and variance across nodes. It can be observed that EIGAN and D-EIGAN converge to similar performances regardless of the number of nodes.

2.4.2 Distributed EIGAN (D-EIGAN)

Varying number of nodes

For the distributed case, we first study the effect of increasing the number of training nodes K. We use synthetic Gaussian data and generate non-i.i.d. data distributions across the nodes by increasing the variance of the Gaussians at each subsequent node k (Fig. 2.13(c) shows the distribution for $k = 3$). Fig. 2.13(a)&(b) show the resulting ally and adversary accuracies obtained when trained on D-EIGAN, on EIGAN, and on the unencoded data. As K increases, the ally performance degrades in each case, due to the higher variance for each class exhibited in the overall dataset $\mathcal{X}$. Overall, we see that D-EIGAN matches the performance of the centrally-trained EIGAN in both metrics, which shows that distributed learning can yield a comparable solution when all parameters ($\phi = 1$) are synchronized frequently ($\delta = 1$). Fig. 2.14 shows the result of the experiment when the nodes instead have i.i.d data. We observe that the performance of the ally and adversary remains reasonably constant (and similar to EIGAN) as we increase the number of nodes under D-EIGAN. From the two experiments, we can conclude that D-EIGAN can readily extend to scenarios where

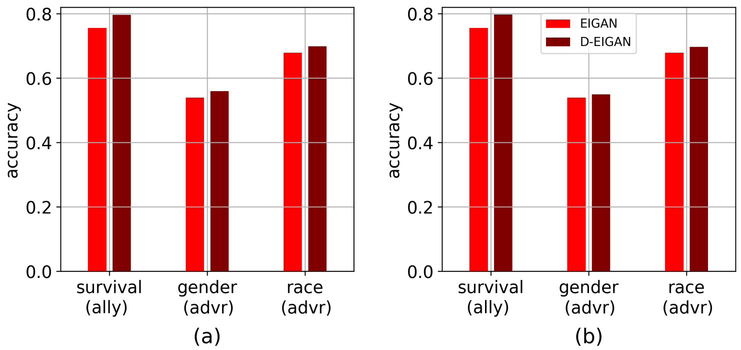


Figure 2.15. Performance of ally and adversary objectives trained on D-EIGAN ($K = 10, \phi = 0.8, \delta = 2$, non-i.i.d) for MIMIC in the cases of (a) all nodes having all three objectives and (b) each node having the ally but only one of the adversaries. The distribution of objectives across the nodes does not affect the resulting accuracies.

data is distributed over larger number of nodes without sacrificing the performance on ally and adversary objectives.

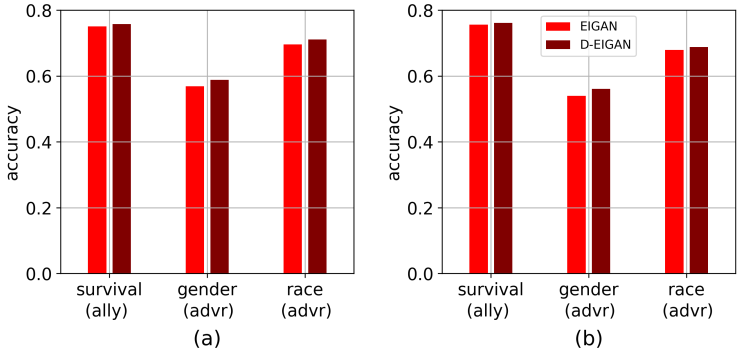


Figure 2.16. Comparison of distributed ($K = 2$ nodes) EIGAN with centralized EIGAN. Survival is the ally objective, and gender and race are the chosen adversary objectives for the experiment. (a) Training of distributed EIGAN involves same adversary objectives, i.e., obfuscating gender and race across the both the nodes. (b) Each node has a different adversary objective, while they share the same ally objective.

Varying objectives across nodes

Next, we study the effect of varying ally and adversary objectives across nodes. For this, we consider the MIMIC dataset and allocate the dataset across $K = 10$ nodes randomly so that each has a different distribution of patient data. In Fig. 2.15, we show the accuracies achieved by D-EIGAN on the one ally and two adversary objectives for two cases: (a) when each node has all three objectives, and (b) when each node has the ally objective, but half have one adversary objective and half have the other. The EIGAN performance on the full dataset is included for comparison. The dataset is distributed in a non-i.i.d manner across nodes by non-uniform random sampling. We see that D-EIGAN in (a) only has a slight improvement over (b) in the case of the gender adversary, which indicates that D-EIGAN is robust to varying node objectives, even though the aggregation period has increased ($\delta = 2$) and the fraction of parameters shared has decreased ($\phi = 0.8$) from Fig. 2.13. The implication of this is that once a data sample is encoded at a node via D-EIGAN, it can be transferred to another node with different objectives and securely applied to ally tasks there, e.g., referring to the healthcare use case in Chapter 1.2, if a patient moves to a different hospital with different health regulations. Similar conclusions are drawn when the data is i.i.d across 2 nodes as shown in Fig. 2.16. This observed behavior, i.e., that a privacy and/or predictivity objective at one node is adopted across all the encoders, is consistent with Proposition 4.

Varying synchronization parameters

Finally, we consider the impact of the aggregation period δ and the sparsification factor ϕ on D-EIGAN. This has implications for the communication resources between the nodes and the server required for training, as discussed in Chapter 2.3. For this experiment, we use the setting from the experiment in Fig. 2.15(a), i.e., with non-i.i.d data and all nodes having all three objectives. In Fig. 2.17, we show the performance of D-EIGAN as (a) δ increases and (b) ϕ increases (EIGAN shown for comparison). In (a), we see that D-EIGAN is robust to the number of training epochs between aggregations, implying that it can be increased to limit the frequency of transmissions to/from the server. In (b), we similarly observe generally robust performance as the fraction of sharing changes, though surprisingly, the performance noticeably *decreases* once ϕ reaches 1 and all are shared. A similar effect was observed by [56], that in the case of distributed model training over non-i.i.d datasets, sparsification actually can *enhance* performance because it minimizes the effect of data bias at each node on the global model. Indeed, in the i.i.d case as shown in Fig. 2.18, we do not observe this effect: We see that there is no visible benefit of sharing only a fraction of parameters, as seen in Fig. 2.18(b). Similarly, in (a) it can be observed that performance over the adversary degrades as the frequency of sync is decreased, i.e., number of epochs between aggregation is increased. This is because the reduction in model bias is not desirable in the case of i.i.d.

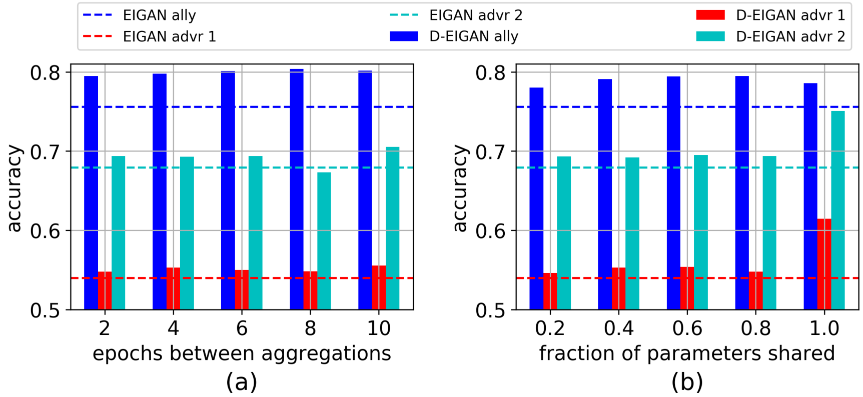


Figure 2.17. Effect of (a) aggregation frequency δ ($\phi = 0.8$) and (b) sparsification factor ϕ ($\delta = 2$) on ally and adversary performance on D-EIGAN for the non-i.i.d case in Fig. 2.15(a). The robust performance shows that D-EIGAN can be applied in communication-constrained environments.

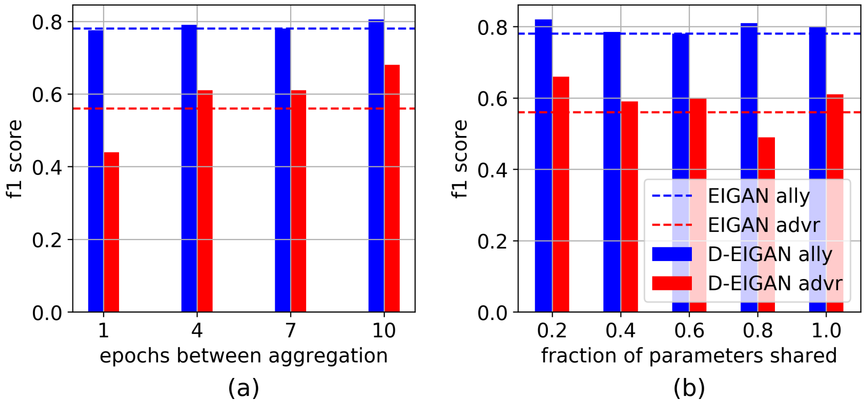


Figure 2.18. Effect of varying (a) frequency of sync (δ, measured in terms of number of epochs between parameter sharing) and (b) fraction of parameters uploaded/downloaded (ϕ) on a distributed implementation consisting of $K = 2$ nodes. The results shows that as the frequency of sync/fraction of parameters shared increases, the performance of the system on hiding the sensitive variable is increased considerably, while there is little effect on the ally convergence.

3. TWO-TIMESCALE HYBRID FEDERATED LEARNING

3.1 System Model and Learning Methodology

In this section, we first describe our edge network system model of D2D-enabled clusters (Chapter 3.1.1) and formalize the ML task for the system (Chapter 3.1.2). Then, we develop our two timescale hybrid federated learning algorithm, `TT-HF` (Chapter 3.1.3).

3.1.1 Edge Network System Model

We consider model learning over the network architecture depicted in Fig. 1.2. The network consists of an edge server (e.g., at a base station) and I edge devices gathered by the set $\mathcal{I} = \{1, \cdots, I\}$. We consider a *cluster-based representation* of the edge, where the devices are partitioned into N sets $\mathcal{S}_1, \cdots, \mathcal{S}_N$. Cluster $\mathcal{S}_c$ contains $s_c = |\mathcal{S}_c|$ edge devices, where $\sum_{c=1}^{N} s_c = I$. We assume that the clusters are formed based on the ability of devices to conduct low-energy D2D communications, e.g., geographic proximity. Thus, one cluster may be a fleet of drones while another is a collection of local IoT sensors. In general, we do not place any restrictions on the composition of devices within a cluster, as long as they possess a common D2D protocol [26] and communicate with a common server.

For edge device $\mathrm{i} \in \mathcal{S}_c$, we let $\mathcal{N}_{\mathrm{i}} \subseteq S_c$ denote the set of its D2D neighbors, determined based on the transmit power of the nodes, the channel conditions between them, and their physical distances (cluster topology is evaluated numerically in Chapter 3.4 based on a wireless communications model). We assume that D2D communications are bidirectional, i.e., $\mathrm{i} \in \mathcal{N}_{\mathrm{i}'}$ if and only if $\mathrm{i}' \in \mathcal{N}_{\mathrm{i}}$, $\forall \mathrm{i}, \mathrm{i}' \in \mathcal{S}_c$. Based on this, we associate a network graph $G_c = (\mathcal{S}_c, \mathcal{E}_c)$ to each cluster, where $\mathcal{E}_c$ denotes the set of edges: $(\mathrm{i}, \mathrm{i}') \in \mathcal{E}_c$ if and only if $\mathrm{i}, \mathrm{i}' \in \mathcal{S}_c$ and $\mathrm{i} \in \mathcal{N}_{\mathrm{i}'}$.

The model training is carried out through a sequence of global aggregations indexed by $k = 1, 2, \cdots$, as will be explained in Chapter 3.1.3. Between global aggregations, the edge devices $\mathrm{i} \in \mathcal{S}_c$ will participate in cooperative consensus procedure with their neighbors $\mathrm{i}' \in \mathcal{N}_\mathrm{i}$. Due to the mobility of the devices, the topology of each cluster (i.e., the number of nodes and their positions inside the cluster) can change over time, although we will assume this evolution is slow compared to a the time in between two global aggregations.

The model learning topology in this paper (Fig. 1.2) is a distinguishing feature compared to the conventional federated learning star topology (Fig. 1.1). Most existing literature is based on Fig. 1.1, e.g., [23], [43]–[49], where devices only communicate with the edge server, while the rest consider fully decentralized (server-less) architectures [60]–[63].

3.1.2 Machine Learning Task Model

Each edge device i owns a dataset $\mathcal{D}_\mathrm{i}$ with $D_\mathrm{i} = |\mathcal{D}_\mathrm{i}|$ data points. Each data point $(\mathbf{x}, y) \in \mathcal{D}_\mathrm{i}$ consists of an m-dimensional feature vector $\mathbf{x} \in \mathbb{R}^m$ and a label $y \in \mathbb{R}$. We let $\hat{f}(\mathbf{x}, y; \mathbf{w})$ denote the *loss* associated with the data point $(\mathbf{x}, y)$ based on *learning model parameter vector* $\mathbf{w} \in \mathbb{R}^M$, where M denotes the dimension of the model. For example, in linear regression, $\hat{f}(\mathbf{x}, y; \mathbf{w}) = \frac{1}{2}(y - \mathbf{w}^\top \mathbf{x})^2$. The *local loss function* at device i is defined as

$$F_\mathrm{i}(\mathbf{w}) = \frac{1}{D_\mathrm{i}} \sum_{(\mathbf{x},y)\in\mathcal{D}_\mathrm{i}} \hat{f}(\mathbf{x}, y; \mathbf{w}). \tag{3.1}$$

We define the *cluster loss function* for $\mathcal{S}_c$ as the average local loss across the cluster,

$$\hat{F}_c(\mathbf{w}) = \sum_{\mathrm{i}\in\mathcal{S}_c} \rho_{\mathrm{i},c} F_\mathrm{i}(\mathbf{w}), \tag{3.2}$$

where $\rho_{\mathrm{i},c}= 1/s_c$ is the weight associated with edge device $\mathrm{i} \in \mathcal{S}_c$ within its cluster. The *global loss function* is then defined as the average loss across the clusters,

$$F(\mathbf{w}) = \sum_{c=1}^{N} \varrho_c \hat{F}_c(\mathbf{w}), \tag{3.3}$$

weighted by the relative cluster size $\varrho_c = s_c\Big(\sum_{c'=1}^{N} s_{c'}\Big)^{-1}$. The weight of each edge node $\mathrm{i} \in \mathcal{S}_c$ relative to the network can thus be obtained as $\rho_{\mathrm{i}} = \varrho_c \cdot \rho_{\mathrm{i},c} = 1/I$, meaning each node contributes equally to the global loss function. The goal of the ML model training is to find the optimal model parameters $\mathbf{w}^* \in \mathbb{R}^M$ for F:

$$\mathbf{w}^* = \underset{\mathbf{w}\in\mathbb{R}^M}{\arg\min}\, F(\mathbf{w}). \tag{3.4}$$

Remark 1. *An alternative way of defining (3.3) is as an average performance over the datapoints, i.e., $F(\mathbf{w}) = \sum_{\mathrm{i}=1}^{I} D_{\mathrm{i}} F_{\mathrm{i}}(\mathbf{w}) / \sum_{\mathrm{j}} D_{\mathrm{j}}$ [22], [24]. Both approaches can be justified: our formulation promotes equal performance across the devices, at the expense of giving devices with lower numbers of datapoints the same priority in the global model. Our analysis can be readily extended to this other formulation too, in which case the distributed consensus algorithms introduced in Chapter 3.1.3 would take a weighted form instead.*

In the following, we make some standard assumptions [23], [24], [43], [44], [53], [103]–[107] on the ML loss function that also imply the existence and uniqueness of $\mathbf{w}^*$. Then, we define a new generic metric to measure the statistical heterogeneity/degree of non-i.i.d. across the local datasets:

Assumption 1. *The following assumptions are made throughout the paper:*

- ***Strong convexity**: F is μ-strongly convex, i.e.,*[1] $\forall \mathbf{w}_1, \mathbf{w}_2$,

$$F(\mathbf{w}_1) \geq F(\mathbf{w}_2) + \nabla F(\mathbf{w}_2)^\top (\mathbf{w}_1 - \mathbf{w}_2) + \frac{\mu}{2}\Big\|\mathbf{w}_1 - \mathbf{w}_2\Big\|^2. \tag{3.5}$$

- ***Smoothness:** F_{i} is β-smooth $\forall \mathrm{i}$, i.e.,*

$$\Big\|\nabla F_{\mathrm{i}}(\mathbf{w}_1) - \nabla F_{\mathrm{i}}(\mathbf{w}_2)\Big\| \leq \beta \Big\|\mathbf{w}_1 - \mathbf{w}_2\Big\|, \ \forall \mathrm{i}, \mathbf{w}_1, \mathbf{w}_2, \tag{3.6}$$

where $\beta > \mu$. This implies β-smoothness of F and $\hat{F}_c$ as well.[2]

[1] ↑Convex ML loss functions, e.g., squared SVM and linear regression, are implemented with a regularization term in practice to improve convergence and avoid model overfitting, which makes them strongly convex [103].

[2] ↑Throughout, $\|\cdot\|$ is always used to denote ℓ_2 norm, unless otherwise stated.

While we leverage these assumptions in our theoretical development, our experiments in Appendix D of [108] demonstrate that our resulting methodology is still effective in the case of non-convex loss functions (in particular, for neural networks). We also remark that strong-convexity of the *global* loss function entailed by Assumption 1 is a much looser requirement than strong-convexity enforced on each device's local function, which we do not assume in this paper.

Definition 3.1.1 (Gradient Diversity)**.** *There exist $\delta \geq 0$ and $\zeta \in [0, 2\beta]$ such that the cluster and global gradients satisfy*

$$\left\|\nabla \hat{F}_c(\mathbf{w}) - \nabla F(\mathbf{w})\right\| \leq \delta + \zeta\|\mathbf{w} - \mathbf{w}^*\|, \ \forall c, \mathbf{w}. \tag{3.7}$$

The conventional definition of gradient diversity used in literature, e.g., as in [24], is a special case of (3.7) with $\zeta = 0$. However, we observe that solely using δ on the right hand side of (3.7) may be troublesome since it can be shown to be not applicable to quadratic functions (such as linear regression problems), and since δ may be prohibitively large,[3] leading to overly pessimistic convergence bounds. Indeed, for all functions satisfying Assumption 1, Definition 3.1.1 holds. To see this, note that we can upper bound the gradient diversity using the triangle inequality as

$$\begin{aligned}
\|\nabla \hat{F}_c(\mathbf{w}) - \nabla F(\mathbf{w})\| &= \|\nabla \hat{F}_c(\mathbf{w}) - \nabla \hat{F}_c(\mathbf{w}^*) + \nabla \hat{F}_c(\mathbf{w}^*) - \underbrace{\nabla F(\mathbf{w}^*)}_{=0} - \nabla F(\mathbf{w})\| \\
&\leq \|\nabla \hat{F}_c(\mathbf{w}) - \nabla \hat{F}_c(\mathbf{w}^*)\| + \|\nabla \hat{F}_c(\mathbf{w}^*)\| \\
&\quad + \|\nabla F(\mathbf{w}) - \nabla F(\mathbf{w}^*)\| \leq \delta + 2\beta\|\mathbf{w} - \mathbf{w}^*\|,
\end{aligned} \tag{3.8}$$

where in the last step above we used the smoothness condition and upper bounded the cluster gradients at the optimal model as $\|\nabla \hat{F}_c(\mathbf{w}^*)\| \leq \delta$. We then introduce a ratio $\omega = \frac{\zeta}{2\beta}$, where $\omega \leq 1$ according to (3.8). Considering ζ in (3.7) changes the dynamics of the convergence analysis and requires new techniques to obtain the convergence bounds, which are part of our contributions in this work.

[3] ↑This is especially true at initialization, where the initial model may be far off the optimal model $\mathbf{w}^*$.

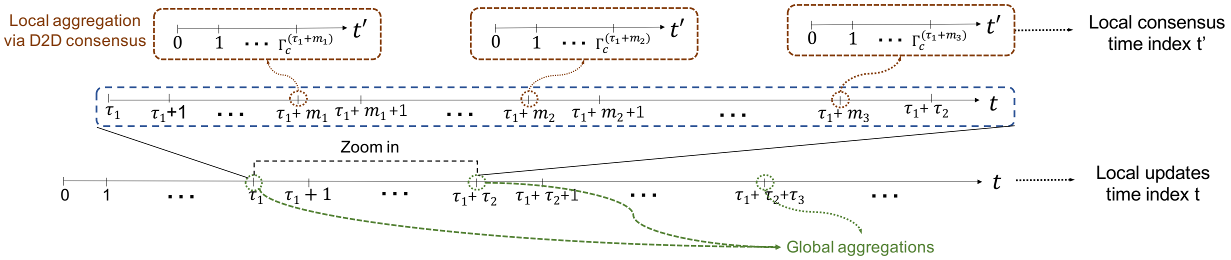


Figure 3.1. Depiction of two timescales in TT-HF. Time index t captures the local descent iterations and global aggregations. In each local training interval, the nodes will aperiodically engage in consensus procedure. Time index t' captures the rounds of these local aggregations.

3.1.3 TT-HF: Two Timescale Hybrid Federated Learning

Overview and rationale

TT-HF is comprised of a sequence of local model training intervals in-between aperiodic global aggregations. During each interval, the devices conduct local stochastic gradient descent (SGD) iterations and aperiodically synchronize their model parameters through local consensus procedure within their clusters.

There are three main practical reasons for incorporating the local consensus procedure into the learning paradigm. First, local consensus can help further suppress any bias of device models to their local datasets, which is one of the main challenges faced in federated learning in environments where data may be non-i.i.d. across the network [24]. Second, local D2D communications during the consensus procedure, typically performed over short ranges [109], [110], are expected to incur much lower device power consumption compared with the global aggregations, which require uplink transmissions to potentially far-away aggregation points (e.g., from smartphone to base station). Third, D2D is becoming a prevalent feature of 5G-and-beyond wireless networks [111], [112].

TT-HF procedure

We index time as a set of discrete time indices $\mathcal{T} = \{1, 2, ...\}$. Global aggregation k occurs at time $t_k \in \mathcal{T}$ (with $t_0 = 0$), so that $\mathcal{T}_k = \{t_{k-1} + 1, ..., t_k\}$ denotes the kth *local* *model training interval* between aggregations $k-1$ and k, of duration $\tau_k = t_k - t_{k-1}$. Since global aggregations are aperiodic, in general $\tau_k \neq \tau_{k'}$ for $k \neq k'$.

The model computed by the server at the kth global aggregation is denoted as $\hat{\mathbf{w}}^{(t_k)} \in \mathbb{R}^M$, which will be defined in (3.17). The model training procedure starts with the server broadcasting $\hat{\mathbf{w}}^{(0)}$ to initialize the devices' local models.

Local SGD iterations: Each device $\mathrm{i} \in \mathcal{I}$ has its own local model, denoted $\mathbf{w}_{\mathrm{i}}^{(t-1)} \in \mathbb{R}^M$ at time $t-1$. Device i performs successive local SGD iterations on its model over time. Specifically, at time $t \in \mathcal{T}_k$, device i randomly samples a mini-batch $\xi_{\mathrm{i}}^{(t-1)}$ of fixed size from its own local dataset $\mathcal{D}_{\mathrm{i}}$, and calculates the *local gradient estimate*

$$\widehat{\mathbf{g}}_{\mathrm{i}}^{(t-1)} = \frac{1}{|\xi_{\mathrm{i}}^{(t-1)}|} \sum_{(\mathbf{x},y)\in\xi_{\mathrm{i}}^{(t-1)}} \hat{f}(\mathbf{x}, y; \mathbf{w}_{\mathrm{i}}^{(t-1)}). \tag{3.9}$$

It then computes its *intermediate updated local model* as

$$\widetilde{\mathbf{w}}_{\mathrm{i}}^{(t)} = \mathbf{w}_{\mathrm{i}}^{(t-1)} - \eta_{t-1}\widehat{\mathbf{g}}_{\mathrm{i}}^{(t-1)},\ t \in \mathcal{T}_k, \tag{3.10}$$

where $\eta_{t-1} > 0$ denotes the step size. The local model $\mathbf{w}_{\mathrm{i}}^{(t)}$ is then updated according to the following consensus-based procedure.

Local model update: At each time $t \in \mathcal{T}_k$, each cluster may engage in local consensus procedure for model updating. The decision of whether to engage in this consensus process at time t – and if so, how many iterations of this process to run – will be developed in Chapter 3.3 based on a performance-efficiency trade-off optimization. If the devices do not execute consensus procedure, we have the conventional model update rule $\mathbf{w}_{\mathrm{i}}^{(t)} = \widetilde{\mathbf{w}}_{\mathrm{i}}^{(t)}$ from (3.10). Otherwise, multiple *rounds* of D2D communication take place, where in each round parameter transfers occur between neighboring devices. In particular, assuming $\Gamma_c^{(t)} > 0$ rounds for cluster c at time t, and letting $t' = 0, \ldots, \Gamma_c^{(t)} - 1$ index the rounds, each node $\mathrm{i} \in \mathcal{S}_c$ carries out the following for $t' = 0, \ldots, \Gamma_c^{(t)} - 1$:

$$\mathbf{z}_{\mathrm{i}}^{(t'+1)} = v_{\mathrm{i},\mathrm{i}}\mathbf{z}_{\mathrm{i}}^{(t')} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}}} v_{\mathrm{i},\mathrm{j}}\mathbf{z}_{\mathrm{j}}^{(t')}, \tag{3.11}$$

where $\mathbf{z}_{\rm i}^{(0)} = \widetilde{\mathbf{w}}_{\rm i}^{(t)}$ is the node's intermediate local model from (3.10), and $v_{\rm i,j} \geq 0$, $\forall {\rm i,j}$ is the consensus weight that node i applies to the vector received from j. At the end of this process, node i takes $\mathbf{w}_{\rm i}^{(t)} = \mathbf{z}_{\rm i}^{(\Gamma_c^{(t)})}$ as its updated local model.

The index t' corresponds to the second timescale in `TT-HF`, referring to the consensus process, as opposed to the index t which captures the time elapsed by the local gradient iterations. Fig. 3.1 illustrates these two timescales, where at certain local iterations t the consensus process t' is run.

To analyze this update process, we will find it convenient to express the consensus procedure in matrix form. Let $\widetilde{\mathbf{W}}_c^{(t)} \in \mathbb{R}^{s_c \times M}$ denote the matrix of intermediate updated local models of the s_c nodes in cluster $\mathcal{S}_c$, where the i-th row of $\widetilde{\mathbf{W}}_c^{(t)}$ corresponds to device i's intermediate local model $\widetilde{\mathbf{w}}_{\rm i}^{(t)}$. Then, the matrix of updated device parameters after the consensus stage, $\mathbf{W}_c^{(t)}$, can be written as

$$\mathbf{W}_c^{(t)} = (\mathbf{V}_c)^{\Gamma_c^{(t)}} \widetilde{\mathbf{W}}_c^{(t)}, \ t \in \mathcal{T}_k, \tag{3.12}$$

where $\Gamma_c^{(t)}$ denotes the rounds of D2D consensus in the cluster, and $\mathbf{V}_c = [v_{\rm i,j}]_{1 \leq {\rm i,j} \leq s_c} \in \mathbb{R}^{s_c \times s_c}$ denotes the *consensus matrix*, which we characterize further below. The i-th row of $\mathbf{W}_c^{(t)}$ corresponds to device i's local update $\mathbf{w}_{\rm i}^{(t)}$, which is then used in (3.9) to calculate the gradient estimate for the next local update. For the times $t \in \mathcal{T}_k$ where consensus is not used, we set $\Gamma_c^{(t)} = 0$, implying $\mathbf{W}_c^{(t)} = \widetilde{\mathbf{W}}_c^{(t)}$ so that devices use their individual gradient updates.

Remark 2. *Note that the graph G_c may change over time t. In this paper, we only require that the set of devices in each cluster remain fixed during each global aggregation period k. We drop the dependency on t for simplicity of presentation, although the analysis implicitly accommodates it. We similarly do so in notations for node and cluster weights $\rho_{\rm i,c}$, ϱ_c introduced in Chapter 3.1.2 and consensus parameters $v_{\rm i,j}$, $\mathbf{V}_c$, λ_c in Chapter 3.1.3. Assuming a fixed vertex set during each global aggregation period is a practical assumption, especially when the devices move slowly and do not leave the cluster during each local training interval. Moreover, although in the analysis we assume that transmissions are outage- and error-free, in Chapter 3.4 we will perform a numerical evaluation to evaluate the impact of fast fad-*

ing and limited channel state information (CSI), resulting in outages and time-varying link configurations.

Consensus characteristics: The consensus matrix $\mathbf{V}_c$ can be constructed in several ways based on the cluster topology G_c. In this paper, we make the following standard assumption [67]:

Assumption 2. *The consensus matrix $\mathbf{V}_c$ satisfies the following conditions: (i) $(\mathbf{V}_c)_{m,n} = 0$ if $(m,n) \notin \mathcal{E}_c$, i.e., nodes only receive from their neighbors; (ii) $\mathbf{V}_c\boldsymbol{1} = \boldsymbol{1}$, i.e., row stochasticity; (iii) $\mathbf{V}_c = {\mathbf{V}_c}^\top$, i.e., symmetry; and (iv) $\rho\Big(\mathbf{V}_c - \frac{\boldsymbol{1}\boldsymbol{1}^\top}{s_c}\Big) < 1$, i.e., the largest eigenvalue of $\mathbf{V}_c - \frac{\boldsymbol{1}\boldsymbol{1}^\top}{s_c}$ has magnitude < 1.*

For example, from the distributed consensus literature [67], one common choice that satisfies this property is $v_{\mathrm{i,j}} = d_c, \forall \mathrm{j} \in \mathcal{N}_\mathrm{i}$ and $v_{\mathrm{i,i}} = 1 - d_c|\mathcal{N}_\mathrm{i}|$, where $0 < d_c < 1/D_c$ and D_c denotes the maximum degree among the nodes in G_c.

The consensus procedure process can be viewed as an imperfect aggregation of the models in each cluster. Specifically, we can write the local parameter at device $\mathrm{i} \in \mathcal{S}_c$ as

$$\mathbf{w}_\mathrm{i}^{(t)} = \bar{\mathbf{w}}_c^{(t)} + \mathbf{e}_\mathrm{i}^{(t)}, \tag{3.13}$$

where $\bar{\mathbf{w}}_c^{(t)} = \sum_{\mathrm{i}\in\mathcal{S}_c} \rho_{\mathrm{i},c}\tilde{\mathbf{w}}_\mathrm{i}^{(t)}$ is the average of the local models in the cluster and $\mathbf{e}_\mathrm{i}^{(t)} \in \mathbb{R}^M$ denotes the *consensus error* caused by limited D2D rounds (i.e., $\Gamma_c^{(t)} < \infty$) among the devices, which can be bounded as in the following lemma.

Lemma 1. *After performing $\Gamma_c^{(t)}$ rounds of consensus in cluster $\mathcal{S}_c$ with the consensus matrix $\mathbf{V}_c$, the consensus error $\mathbf{e}_\mathrm{i}^{(t)}$ satisfies*

$$\|\mathbf{e}_\mathrm{i}^{(t)}\| \leq (\lambda_c)^{\Gamma_c^{(t)}} \sqrt{s_c} \underbrace{\max_{\mathrm{j,j'}\in\mathcal{S}_c} \|\tilde{\mathbf{w}}_\mathrm{j}^{(t)} - \tilde{\mathbf{w}}_{\mathrm{j'}}^{(t)}\|}_{\triangleq \Upsilon_c^{(t)}}, \ \forall \mathrm{i} \in \mathcal{S}_c. \tag{3.14}$$

where $\lambda_c = \rho\Big(\mathbf{V}_c - \frac{\boldsymbol{1}\boldsymbol{1}^\top}{s_c}\Big)$.

Algorithm 3 Two timescale hybrid federated learning `TT-HF` with set control parameters.

```
Input: Length of training $T$, number of global aggregations $K$, D2D rounds $\{\Gamma_c^{(t)}\}_{t=1}^{T}$, $\forall c$, length of local model training intervals $\tau_k,\ k = 1, ..., K$
Output: Final global model $\hat{\mathbf{w}}^{(T)}$
// Initialization by the server
Initialize $\hat{\mathbf{w}}^{(0)}$ and broadcast it among the devices along with the indices $n_c$ of the sampled devices for the first global aggregation.
for $k = 1 : K$ do
    for $t = t_{k-1} + 1 : t_k$ do
        for $c = 1 : N$ do
            // Procedure at the clusters
            Each device $\mathrm{i} \in \mathcal{S}_c$ performs local SGD update based on (3.9) and (3.10) using $\mathbf{w}_{\mathrm{i}}^{(t-1)}$ to obtain $\widetilde{\mathbf{w}}_{\mathrm{i}}^{(t)}$.
            Devices inside the cluster conduct $\Gamma_c^{(t)}$ rounds of consensus procedure based on (3.11), initializing $\mathbf{z}_{\mathrm{i}}^{(0)} = \widetilde{\mathbf{w}}_{\mathrm{i}}^{(t)}$ and setting $\mathbf{w}_{\mathrm{i}}^{(t)} = \mathbf{z}_{\mathrm{i}}^{(\Gamma_c^{(t)})}$.
        end for
        if $t = t_k$ then
            // Procedure at the clusters
            Each sampled device $n_c$ sends $\mathbf{w}_{n_c}^{(t_k)}$ to the server.
            // Procedure at the server
            Compute $\hat{\mathbf{w}}(t)$ using (3.17), and broadcast it among the devices along with the indices $n_c$ chosen for the next global aggregation.
        end if
    end for
end for
```

Sketch of Proof: Let $\overline{\mathbf{W}}_c^{(t)} = \frac{1}{s_c}\mathbf{1}\mathbf{1}^\top\widetilde{\mathbf{W}}_c^{(t)}$ be the matrix with rows given by the average model parameters across the cluster, and let

$$\mathbf{E}_c^{(t)} = \mathbf{W}_c^{(t)} - \overline{\mathbf{W}}_c^{(t)} = [\,(\mathbf{V}_c)^{\Gamma_c^{(t)}} - \mathbf{1}^\top\mathbf{1}/s_c][\widetilde{\mathbf{W}}_c^{(t)} - \overline{\mathbf{W}}_c^{(t)}], \tag{3.15}$$

so that $[\mathbf{E}_c^{(t)}]_{\mathrm{i},:}$ (ith column of $\mathbf{E}_c^{(t)}$) $= \mathbf{e}_{\mathrm{i}}^{(t)}$, where in the second step we used (3.12) and the fact that $\mathbf{1}^\top\mathbf{E}_c^{(t)} = \mathbf{0}$ (hence $\mathbf{E}_c^{(t)} = [\mathbf{I} - \mathbf{1}^\top\mathbf{1}/s_c]\mathbf{E}_c^{(t)}$). Therefore, using Assumption 2, we can bound the consensus error as

$$
\begin{aligned}
\|\mathbf{e}_{\mathrm{i}}^{(t)}\|^2 &\leq \mathrm{trace}((\mathbf{E}_c^{(t)})^\top \mathbf{E}_c^{(t)}) \\
&= \mathrm{trace}\Big([\widetilde{\mathbf{W}}_c^{(t)} - \overline{\mathbf{W}}_c^{(t)}]^\top [\,(\mathbf{V}_c)^{\Gamma_c^{(t)}} - \mathbf{1}^\top \mathbf{1}/s_c]^2 [\widetilde{\mathbf{W}}_c^{(t)} - \overline{\mathbf{W}}_c^{(t)}]\Big) \\
&\leq (\lambda_c)^{2\Gamma_c^{(t)}} \sum_{\mathrm{j}=1}^{s_c} \|\tilde{\mathbf{w}}_{\mathrm{j}}^{(t)} - \bar{\mathbf{w}}_c^{(t)}\|^2 \\
&\leq (\lambda_c)^{2\Gamma_c^{(t)}} \frac{1}{s_c} \sum_{\mathrm{j},\mathrm{j}'=1}^{s_c} \|\tilde{\mathbf{w}}_{\mathrm{j}}^{(t)} - \tilde{\mathbf{w}}_{\mathrm{j}'}^{(t)}\|^2 \\
&\leq (\lambda_c)^{2\Gamma_c^{(t)}} s_c \max_{\mathrm{j},\mathrm{j}' \in \mathcal{S}_c} \|\tilde{\mathbf{w}}_{\mathrm{j}}^{(t)} - \tilde{\mathbf{w}}_{\mathrm{j}'}^{(t)}\|^2,
\end{aligned}
\tag{3.16}
$$

so that the result directly follows. For the complete proof, refer to Appendix C.4 in our technical report [108]. ■

Note that $\Upsilon_c^{(t)}$ defined in (3.14) captures the divergence of intermediate updated local model parameters in cluster $\mathcal{S}_c$ at time $t \in \mathcal{T}_k$ (before consensus is performed). Intuitively, according to (3.14), to make the consensus error smaller, more rounds of consensus need to be performed. However, this may be impractical due to energy and delay considerations, hence a trade-off arises between the consensus error and the energy/delay cost. This trade-off will be optimized by tuning $\Gamma_c^{(t)}$, via our adaptive control algorithm developed in Chapter 3.3.

Global aggregation: At the end of each local model training interval $\mathcal{T}_k$, the global model $\mathbf{w}$ will be updated based on the trained local model updates. Referring to Fig. 1.2, the main server will *sample* one device from each cluster c uniformly at random, and request these devices to upload their local models, so that the new global model is updated as

$$
\hat{\mathbf{w}}^{(t)} = \sum_{c=1}^{N} \varrho_c \mathbf{w}_{n_c}^{(t)}, \quad t = t_k, k = 1, 2, ... \tag{3.17}
$$

where n_c is the node sampled from cluster c at time t. This sampling technique is introduced to reduce the uplink communication cost by a factor of the cluster sizes, and is enabled by the consensus procedure, which mimics a local aggregation procedure within a cluster (albeit imperfectly due to consensus errors, see (3.13)) [26]. The global model is then broadcast by the main server to all of the edge devices, which override their local models at time t_k: $\mathbf{w}_{\mathrm{i}}^{(t_k)} = \hat{\mathbf{w}}^{(t_k)}$, $\forall$i. The process then repeats for $\mathcal{T}_{k+1}$.

A summary of the `TT-HF` algorithm developed in this section (for set control parameters) is given in Algorithm 3.

Remark 3. *Note that we consider digital transmission (in both D2D and uplink/downlink communications) where using state-of-the-art techniques in encoding/decoding, e.g., low density parity check (LDPC) codes, the bit error rate (BER) is reasonably small and negligible [113]. Moreover, the effect of quantized model transmissions can be readily incorporated using techniques developed in [66], and precoding techniques may be used to mitigate the effect of signal outage due to fading [114]. Therefore, in this analysis, we assume that the model parameters transmitted by the devices to their neighbors (during consensus) and then to the server (during global aggregation) are received with no errors at the respective receivers. The impact of outages due to fast fading and lack of CSI will be studied numerically in Chapter 3.4.*

3.2 Convergence Analysis of `TT-HF`

In this section, we theoretically analyze the convergence behavior of `TT-HF`. Our main results are presented in Chapter 3.2.2 and Chapter 3.2.3. Before then, in Chapter 3.2.1, we introduce some additional definitions and a key proposition for the analysis.

3.2.1 Definitions and Bounding Model Dispersion

We first introduce a standard assumption on the noise of gradient estimation, and then define an upper bound on the average of consensus error for the clusters:

Assumption 3. *Let* $\mathbf{n}_{\mathrm{i}}^{(t)}{=}\widehat{\mathbf{g}}_{\mathrm{i}}^{(t)}{-}\nabla F_{\mathrm{i}}(\mathbf{w}_{\mathrm{i}}^{(t)})$ $\forall \mathrm{i}, t$ *denote the noise of the estimated gradient through the SGD process for device* i*. We assume that it is unbiased with bounded variance, i.e.* $\mathbb{E}[\mathbf{n}_{\mathrm{i}}^{(t)}|\mathbf{w}_{\mathrm{i}}^{(t)}]{=}0$ *and* $\exists\sigma{>}0$: $\mathbb{E}[\|\mathbf{n}_{\mathrm{i}}^{(t)}\|^2|\mathbf{w}_{\mathrm{i}}^{(t)}]{\leq}\sigma^2$, $\forall \mathrm{i}, t$.

Moreover, the following condition bounds the consensus error within each cluster.

Condition 1. *Let $\epsilon_c^{(t)}$ be an upper bound on the average of the consensus error inside cluster c at time t, i.e.,*

$$\frac{1}{s_c}\sum_{\mathrm{i}\in\mathcal{S}_c}\|\mathbf{e}_{\mathrm{i}}^{(t)}\|^2 \leq (\epsilon_c^{(t)})^2. \tag{3.18}$$

We further define $(\epsilon^{(t)})^2 = \sum\limits_{c=1}^{N}\rho_c(\epsilon_c^{(t)})^2$ as the average of these upper bounds over the network at time t.

In fact, using Lemma 1, this condition can be satisfied by tuning the number of consensus steps. In our analysis, we will derive conditions on $\epsilon_c^{(t)}$ that are sufficient to guarantee convergence of `TT-HF` (see Proposition 3.2.1).

We next define the expected variance in models across clusters at a given time, which we refer to as *model dispersion*:

Definition 3.2.1. *We define the expected model dispersion across the clusters at time t as*

$$A^{(t)} = \mathbb{E}\left[\sum_{c=1}^{N}\varrho_c\left\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\right\|^2\right], \tag{3.19}$$

where $\bar{\mathbf{w}}_c^{(t)}$ is defined in (3.13) and $\bar{\mathbf{w}}^{(t)} = \sum\limits_{c=1}^{N}\varrho_c\bar{\mathbf{w}}_c^{(t)}$ is the global average of the local models at time t.

$A^{(t)}$ measures the degree to which the cluster models deviate from their average throughout the training process. Obtaining an upper bound on this quantity is non-trivial due to the coupling between the gradient diversity and the model parameters imposed by (3.7). For an appropriate choice of step size in (3.10), we upper bound this quantity at time t through a set of new techniques that include the mathematics of *coupled dynamic systems*. Specifically, we have the following result:

Proposition 3.2.1. *If $\eta_t = \frac{\gamma}{t+\alpha}$ for some $\gamma > 0$, $\epsilon^{(t)}$ is non-increasing for $t \in \mathcal{T}_k$, i.e., $\epsilon^{(t+1)} \leq \epsilon^{(t)}$, and $\alpha \geq \gamma\beta\max\{\lambda_+ - 2 + \frac{\mu}{2\beta}, \frac{\beta}{\mu}\}$, then*

$$A^{(t)} \leq \frac{16\omega^2}{\mu}(\Sigma_{+,t})^2[F(\bar{\mathbf{w}}(t_{k-1})) - F(\mathbf{w}^*)] + 25(\Sigma_{+,t})^2\left(\frac{\sigma^2+\delta^2}{\beta^2} + (\epsilon^{(0)})^2\right), \quad t \in \mathcal{T}_k, \tag{3.20}$$

where

$$\Sigma_{+,t} = \sum_{\ell=t_{k-1}}^{t-1} \left(\prod_{\mathrm{j}=t_{k-1}}^{\ell-1} (1+\eta_{\mathrm{j}}\beta\lambda_+) \right) \beta\eta_\ell \left(\prod_{\mathrm{j}=\ell+1}^{t-1} (1+\eta_{\mathrm{j}}\beta) \right),$$

and $\lambda_+ = 1 - \frac{\mu}{4\beta} + \sqrt{(1+\frac{\mu}{4\beta})^2 + 2\omega}$.

Sketch of Proof: See Appendix A.1. ■

The bound in (3.20) demonstrates how the expected model dispersion across the clusters increases with respect to the consensus error ($\epsilon^{(0)}$), the noise in the gradient estimation (σ^2), and the heterogeneity of local datasets (δ, ω). Intuitively, the upper bound in (3.20) dictates that, the larger $\epsilon^{(0)}$, σ^2, δ or ω, the larger the dispersion, due to error propagation in the network. Proposition 3.2.1 will be an instrumental result in the convergence proof developed in the next section.

3.2.2 General Convergence Behavior of $\hat{\mathbf{w}}^{(t)}$

Next, we focus on the convergence of the global loss. In the following theorem, we bound the expected distance that the global loss is from the optimal over time, as a function of the model dispersion.

Theorem 3.2.1. *When using* `TT-HF` *for ML model training with* $\eta_t \leq 1/\beta$ $\forall t$*, the one-step behavior of the global model* $\hat{\mathbf{w}}^{(t)}$ *(see (3.17)) satisfies, for* $t \in \mathcal{T}_k$*,*

$$\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)\right] \leq \underbrace{(1-\mu\eta_t)\mathbb{E}[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)]}_{(a)}$$
$$+ \underbrace{\frac{\eta_t\beta^2}{2}A^{(t)} + \frac{1}{2}[\eta_t\beta^2(\epsilon^{(t)})^2 + \eta_t^2\beta\sigma^2 + \beta(\epsilon^{(t+1)})^2]}_{(b)}, \tag{3.21}$$

where $A^{(t)}$ *is the model dispersion from Definition 3.2.1.*

Sketch of Proof: See Appendix A.2. ■

Theorem C.2.1 quantifies the dynamics of the global model relative to the optimal model during a given update period $\mathcal{T}_k$ of `TT-HF`. Since the theorem holds for all $t \in \mathcal{T}_k$, it also quantifies the suboptimality gap when global aggregation is performed at time $t+1 =$

t_k. Note that the term (a) corresponds to the one-step progress of a *centralized* gradient descent under strongly-convex global loss (Assumption 1), so that the term (b) quantifies the additional loss incurred as a result of the model dispersion across the clusters ($A^{(t)}$, which in turn is bounded by Proposition 3.2.1), consensus errors ($\epsilon^{(t)}$), and SGD noise (σ^2). In fact, without careful choice of our control parameters, the sequence $\mathbb{E}[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)]$ may diverge. Thus, we are motivated to find conditions under which convergence is guaranteed, and furthermore, under which the upper bound in (3.21) will approach zero.

Specifically, we aim for `TT-HF` to match the asymptotic convergence behavior of centralized stochastic gradient descent (SGD) under a diminishing step size, which is $\mathcal{O}(1/t)$ [115]. From (3.21), we see that to match SGD, the terms in (b) should be of order $\mathcal{O}(\eta_t^2)$, i.e., the same as the degradation due to the SGD noise, $\eta_t^2\beta\sigma^2/2$. This implies that control parameters need to be tuned in such a way that $A^{(t)}{=}\mathcal{O}(\eta_t)$ and $\epsilon^{(t)}{=}\mathcal{O}(\eta_t)$. Proving that these conditions hold under proper choice of parameters will be part of Theorem C.3.1.

3.2.3 Sublinear Convergence Rate of $\hat{\mathbf{w}}^{(t)}$

Among the quantities involved in Theorem C.2.1, η_t, τ_k and $\epsilon^{(t)}$ are the three tunable parameters that directly impact the learning performance of `TT-HF`. We now prove that with proper choice of these parameters, `TT-HF` enjoys sub-linear convergence with rate of $\mathcal{O}(1/t)$.

Theorem 3.2.2. *Under Assumptions 1, 2, and 3, suppose* $\eta_t = \frac{\gamma}{t+\alpha}$ *and* $\epsilon^{(t)} = \eta_t\phi$*, where* $\gamma > 1/\mu$*,* $\phi > 0$*,* $\alpha \geq \alpha_{\min}$ *and* $\omega < \omega_{\max}(\alpha)$*. Then, by using* `TT-HF` *for ML model training,*

$$\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)\right] \leq \frac{\nu}{t+\alpha}, \quad \forall t, \tag{3.22}$$

where $\tau = \max_{1 \le \ell \le k}\{\tau_\ell\}$,

$$\alpha_{\min} \triangleq \gamma\beta \max\left\{\frac{\mu}{4\beta} - 1 + \sqrt{(1+\frac{\mu}{4\beta})^2 + 2\omega}, \frac{\beta}{\mu}\right\}, \tag{3.23}$$

$$\omega_{\max}(\alpha) \triangleq \frac{1}{\beta\gamma}\sqrt{\frac{\alpha}{Z_1}}\sqrt{\mu\gamma - 1 + \frac{1}{1+\alpha}}, \tag{3.24}$$

$$\nu \triangleq \max\left\{\frac{\beta^2\gamma^2 Z_2}{\mu\gamma - 1}, \frac{\alpha Z_2/Z_1}{\omega_{\max}^2 - \omega^2}, \alpha\left[F(\hat{\mathbf{w}}^{(0)}) - F(\mathbf{w}^*)\right]\right\}, \tag{3.25}$$

$$Z_1 = \frac{32\beta^2\gamma}{\mu}(\tau - 1)\left(1 + \frac{\tau}{\alpha - 1}\right)^2\left(1 + \frac{\tau - 1}{\alpha - 1}\right)^{6\beta\gamma} \tag{3.26}$$

$$Z_2 = \frac{\sigma^2 + 2\phi^2}{2\beta} + 50\gamma(\tau - 1)\left(1 + \frac{\tau - 2}{\alpha + 1}\right) \times \left(1 + \frac{\tau - 1}{\alpha - 1}\right)^{6\beta\gamma}\left(\sigma^2 + \phi^2 + \delta^2\right). \tag{3.27}$$

Sketch of Proof: See Appendix A.3. ■

Theorem 3.2.2 is one of the central contributions of this paper, revealing how several parameters (some controllable and others characteristic of the environment) affect the convergence of `TT-HF`, and conditions under which $\mathcal{O}(1/t)$ convergence can be achieved. We make several key observations. First, to achieve $\mathcal{O}(1/t)$ convergence, the gradient diversity parameter $\omega = \frac{\zeta}{2\beta}$ should not be too large ($\omega < \omega_{\max}(\alpha)$); in fact, ω induces error propagation of order $\sim \|\mathbf{w}_c - \mathbf{w}^*\|$, so that too large values of ω may cause the error to diverge. Since $\omega_{\max}(\alpha)$ is an increasing function of α (see (3.24)), larger values of ω may be tolerated by increasing α, i.e., by using a smaller step-size η_t, confirming the intuition that larger gradient diversity requires a smaller step-size for convergence. However, the penalty incurred may be slower convergence of the suboptimality gap (since ν increases with α, see (3.25)).

We now discuss the choice of the consensus error $\epsilon^{(t)}$. To guarantee $\mathcal{O}(1/t)$ convergence, Theorem 3.2.2 dictates that it should be chosen as $\epsilon^{(t)} = \eta_t\phi$ for a constant $\phi > 0$, i.e. it should decrease over time according to the step-size. To see that this is a feasible and practical condition, note from Lemma 1 that the upper bound of $\|\mathbf{e}_i^{(t)}\|$ increases proportionally to the divergence $\Upsilon_c^{(t)}$ (see (3.14)), and decreases at geometric rate with the number of consensus steps. In turn, $\Upsilon_c^{(t)}$ can be shown to be of the order of the step-size η_t (assuming $\eta_t \approx \eta_{t-1}$):

$$\Upsilon_c^{(t)} = \max_{\mathrm{j},\mathrm{j}'\in\mathcal{S}_c} \|\widetilde{\mathbf{w}}_{\mathrm{j}}^{(t)} - \widetilde{\mathbf{w}}_{\mathrm{j}'}^{(t)}\| \approx \eta_t \max_{\mathrm{j},\mathrm{j}'\in\mathcal{S}_c} \left\|\widehat{\mathbf{g}}_{\mathrm{j}}^{(t-1)} - \widehat{\mathbf{g}}_{\mathrm{j}'}^{(t-1)}\right\|,$$

where we used (3.10), and the approximation holds if we assume the difference $\mathbf{w}_{\mathrm{j}}^{(t-1)} - \mathbf{w}_{\mathrm{j}'}^{(t-1)}$ in initial model parameters at $t-1$ is negligible compared to the gradients. This is a legit assumption, since $\mathbf{w}_{\mathrm{j}}^{(t-1)}$ is the model parameter at node j, *after* the consensus rounds at time $t-1$. Using Lemma 1, it then follows that, to make $\epsilon^{(t)} = \eta_t\phi$, the number of consensus rounds should be chosen such that $(\lambda_c)^{\Gamma_c^{(t)}} \approx \frac{1}{\sqrt{s_c}}\phi / \max_{\mathrm{j},\mathrm{j}'\in\mathcal{S}_c} \left\|\widehat{\mathbf{g}}_{\mathrm{j}}^{(t-1)} - \widehat{\mathbf{g}}_{\mathrm{j}'}^{(t-1)}\right\|$, and are thus dominated by the divergence of local gradients within the cluster and SGD noise, irrespective of the step-size. We will use this property in the development of our control algorithm for $\Gamma_c^{(t)}$ in Chapter 3.3.

The bound also shows the impact of the duration of local model training intervals τ on the convergence, through the term ν in (3.22). In particular, from (3.25), it can be seen that increasing τ results in a sharp increase of ν (through the factors Z_1 and Z_2 defined in (3.26) and (3.27)). Moreover, we also observe a quadratic impact on ν with respect to the consensus error $\epsilon^{(t)}$ (through ϕ). It then follows that, all else constant, increasing the value of τ requires a smaller value of ϕ (i.e., more accurate consensus) to achieve a desired value of ν in (3.25). This is consistent with how `TT-HF` is designed, since the motivation for including consensus rounds (to decrease $\epsilon^{(t)}$) is to reduce the global aggregation frequency, which results in uplink bandwidth utilization and power consumption savings.

These observations reveal a trade-off between accuracy, delay, and energy consumption. In the next section, we leverage these relationships in developing an adaptive algorithm for `TT-HF` that tunes the control parameters to achieve the convergence bound in Theorem 3.2.2 while minimizing network costs.

3.3 Adaptive Control Algorithm for `TT-HF`

There are three parameters in `TT-HF` that can be tuned over time: (i) local model training intervals τ_k, (ii) gradient descent step size η_t, and (iii) rounds of D2D communications $\Gamma_c^{(t)}$. In this section, we develop a control algorithm (Chapter 3.3.4) based on Theorem 3.2.2

for tuning (i), (ii) at the main server at the beginning of each global aggregation, and (iii) at each device cluster in a decentralized manner. To do so, we propose an approach for determining the learning-related parameters (Chapter 3.3.1), a resource-performance tradeoff optimization for τ_k and $\Gamma_c^{(t)}$ (Chapter 3.3.2), and estimation procedures for dataset-related parameters (Chapter 3.3.3).

3.3.1 Learning-Related Parameters (α, γ, ϕ, η_t)

We aim to tune the step size-related parameters (α, γ) and the consensus error coefficient (ϕ) to satisfy the conditions in Theorem 3.2.2. In this section, we present a method for doing so given properties of the ML model, local datasets, and SGD noise ($\beta, \mu, \zeta, \delta, \sigma$, and thus $\omega = \zeta/(2\beta)$). Later in Chapter 3.3.3, we will develop methods for estimating ζ, δ, σ at the server.[4] We assume that the latency-sensitivity of the learning application specifies a tolerable amount of time that `TT-HF` can wait between consecutive global aggregations, i.e., the value of τ.

To tune the step size parameters, first, a value of γ is determined such that $\gamma > 1/\mu$. Then, since smaller values of α are associated with faster convergence, the minimum value of α that simultaneously satisfies the conditions in the statement of Theorem 3.2.2 is chosen, i.e., $\alpha \geq \alpha_{\min}$ and $\omega_{\max} > \omega$ (note that $\omega_{\max}$ is a function of α, see (3.24).

Let T be a (maximum) desirable duration of the entire `TT-HF` algorithm, and ξ be a (maximum) desirable loss at the end of the model training, which may be chosen based on the learning application. To satisfy the loss requirement, from Theorem 3.2.2 the following condition needs to be satisfied,

$$\frac{\nu}{T+\alpha} \leq \xi, \tag{3.28}$$

yielding a maximum value tolerated for ν, i.e., $\nu^{\mathsf{max}} = \xi(T+\alpha)$. Since ν is a function of the local model training period τ and consensus coefficient ϕ (see (3.25)), this bound places a condition on the parameters τ and ϕ. Furthermore, with the values of α and γ chosen above, along with the value of τ, the algorithm may not always be able to provide any arbitrary

[4] ↑We assume that β and μ can be computed at the server prior to training given the knowledge of the deployed ML model.

desired loss ξ at time T. Therefore, considering the expression for ν from Theorem 3.2.2, the following feasibility check is conducted:

$$\max\left\{\frac{\beta^2\gamma^2 Z_2^{\mathsf{min}}}{\mu\gamma-1}, \frac{\alpha Z_2^{\mathsf{min}}/Z_1}{\omega_{\max}^2-\omega^2}, \frac{\alpha\|\nabla F(\hat{\mathbf{w}}^{(0)})\|^2}{2\mu}\right\}\leq \nu^{\mathsf{max}}, \tag{3.29}$$

where

$$Z_2^{\mathsf{min}} = \frac{\sigma^2}{2\beta} + 50\gamma(\tau-1)\Big(1+\frac{\tau-2}{\alpha+1}\Big)\Big(1+\frac{\tau-1}{\alpha-1}\Big)^{6\beta\gamma}\Big(\sigma^2+\delta^2\Big)$$

is the value of Z_2 obtained by setting the consensus coefficient $\phi = 0$ in (3.27). The third term inside the max of (3.29) is obtained via the Polyak-Lojasiewicz inequality $\left\|\nabla F(\hat{\mathbf{w}}^{(t)})\right\|^2 \geq 2\mu[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)]$, since the value of $F(\mathbf{w}^*)$ is not known, whereas $\nabla F(\hat{\mathbf{w}}^{(t)})$ can be estimated using the local gradient of the sampled devices at the server. If (3.29) is not satisfied, the chosen values of τ, ξ and/or T must be loosened, and this procedure must be repeated until (3.29) becomes feasible.

Once α, γ and τ are chosen, we move to selecting ϕ. All else constant, larger consensus errors would be more favorable in `TT-HF` due to requiring less rounds of D2D communications (Lemma 1). The largest possible value of ϕ, denoted ϕ^{max}, can be obtained directly from (3.29) via replacing Z_2^{min} with Z_2 and considering the definition of Z_2 in (3.27):[5]

$$\phi^{\mathsf{max}} = \sqrt{\beta}\sqrt{\frac{\nu^{\mathsf{max}}\min\left\{\frac{\mu\gamma-1}{\beta^2\gamma^2}, \frac{Z_1(\omega_{\max}^2-\omega^2)}{\alpha}\right\} - Z_2^{\mathsf{min}}}{1+50\beta\gamma(\tau-1)\left(1+\frac{\tau-2}{\alpha+1}\right)\left(1+\frac{\tau-1}{\alpha-1}\right)^{6\beta\gamma}}}. \tag{3.30}$$

Note that (3.30) exists if the feasibility check in (3.29) is satisfied.

The values of ν^{max} and α are re-computed at the server at each global aggregation. The devices use this to set their step sizes η_t during the next local update period accordingly.

[5] ↑In the max function in (3.30), only the first two arguments from the function in (3.29) are present as the third is independent of Z_2 and ϕ.

3.3.2 Local Training Periods (τ_k) and Consensus Rounds ($\Gamma_c^{(t)}$)

One of the main motivations behind TT-HF is minimizing the resource consumption among edge devices during model training. We thus propose tuning the τ_k and $\Gamma_c^{(t)}$ parameters according to the joint impact of three metrics: energy consumption, training delay imposed by consensus, and trained model performance. To capture this trade-off, we formulate an optimization problem ($\boldsymbol{\mathcal{P}}$) solved by the main server at the beginning of each global aggregation period $\mathcal{T}_k$, i.e., when $t = t_{k-1}$:

$$(\boldsymbol{\mathcal{P}}): \quad \min_{\tau_k} \underbrace{\frac{c_1\Big(E_{\text{Glob}} + \sum\limits_{t=t_{k-1}}^{t_{k-1}+\tau_k} \sum\limits_{c=1}^{N} \Gamma_c^{(t)} s_c E_{\text{D2D}}\Big)}{\tau_k}}_{(a)} +$$

$$\underbrace{\frac{c_2\Big(\Delta_{\text{Glob}} + \sum\limits_{t=t_{k-1}}^{t_{k-1}+\tau_k} \sum\limits_{c=1}^{N} \Gamma_c^{(t)} \Delta_{\text{D2D}}\Big)}{\tau_k}}_{(b)} + c_3 \Bigg(\underbrace{1 - \frac{t_{k-1}+\alpha}{t_{k-1}+\tau_k+\alpha}}_{(c)}\Bigg)$$

s.t.

$$\Gamma_c^{(t)} = \max\left\{\left\lceil \log\left(\frac{\eta_t \phi}{\sqrt{s_c}\Upsilon_c^{(t)}}\right) / \log\left(\lambda_c\right)\right\rceil, 0\right\}, \forall c, \tag{3.31}$$

$$1 \le \tau_k \le \min\{\tau, T - t_{k-1}\}, \tau_k \in \mathbb{Z}^+, \tag{3.32}$$

$$\Upsilon_c^{(t_{k-1})} = 0, \ \forall c, \tag{3.33}$$

$$\Upsilon_c^{(t)} = 1_{\{\Gamma_c^{(t-1)}=0\}} \underbrace{(A_c^{(k)} \Upsilon_c^{(t-1)} + B_c^{(k)})}_{(d)} +$$

$$\Big(1 - 1_{\{\Gamma_c^{(t-1)}=0\}}\Big) \underbrace{(a_c^{(k)} \Upsilon_c^{(t-1)} + b_c^{(k)})}_{\text{(e)}}, \ \forall c, \tag{3.34}$$

where E_{D2D} is the energy consumption of each D2D communication round for each device, E_{Glob} is the energy consumption for device-to-server communications, Δ_{D2D} is the communication delay per D2D round conducted in parallel among the devices, and Δ_{Glob} is the device-to-server communication delay. The objective function captures the trade-off between average energy consumption (term (a)), average D2D delay (term (b)), and expected

ML model performance (term (c)). In particular, term (c) is a penalty on the ratio of the upper bound given in (3.22) between the updated model and the previous model at the main server. A larger ratio implies the difference in performance between the aggregations is smaller, and thus that synchronization is occurring frequently, consistent with τ_k appearing in the denominator. This term also contains a diminishing marginal return from global aggregations as the learning proceeds: smaller values of τ_k are more favorable in the initial stages of ML model training, i.e., for smaller t_{k-1}. This matches well with the intuition that ML model performance has a sharper increase at the beginning of model training, so frequent aggregations at smaller t_{k-1} will have larger benefit to the model performance stored at the main server. The coefficients $c_1, c_2, c_3 \geq 0$ are introduced to weigh each of the design considerations.

The equality constraint on $\Gamma_c^{(t)}$ in (3.31) forces the condition $\epsilon^{(t)} = \eta_t \phi$ imposed by Theorem 3.2.2, obtained using the result in Lemma 1. This equality reveals the condition under which the local aggregations, i.e., D2D communication, are triggered. Note that since the spectral radius is less than one, we have $\log(\lambda_c) < 0$, thus the requirement to conduct D2D communications, i.e., triggering in cluster model synchronization, is $\sqrt{s_c}\Upsilon_c^{(t)} > \eta\phi$. In other words, when the divergence of local models exceeds a predefined threshold $\Upsilon_c^{(t)} > \frac{\eta\phi}{\sqrt{s_c}}$, local synchronization is triggered via D2D communication, and the number of D2D rounds is given by $\Gamma_c^{(t)}$. Also, (3.32) ensures the feasible ranges for τ_k.

As can be seen from (3.31), to obtain the desired consensus rounds for future times $t \in \mathcal{T}_k$, the values of $\Upsilon_c^{(t)}$ – the divergence of model parameters in each cluster – are needed. Obtaining these exact values at $t = t_{k-1}$ is not possible since it requires the knowledge of the model parameters $\tilde{\mathbf{w}}_i^{(t)}$ of the devices for the future timesteps, which is non-causal. To address this challenge, problem $(\boldsymbol{\mathcal{P}})$ incorporates the additional constraints (3.33) and (3.34), which aim to estimate the future values of $\Upsilon_c^{(t)}$, $\forall c$ through a time-series predictor, initialized as $\Upsilon_c^{(t_{k-1})} = 0$ in (3.33) (since, at the beginning of the period, the nodes start with the same model provided by the server). In the expression (3.34), $1_{\{\Gamma_c^{(t-1)}=0\}}$ takes the value of 1 when no D2D communication rounds are performed at $t-1$, and 0 otherwise. Two linear terms ((d) and (e)) are included, one for each of these cases, characterized by coefficients $A_c^{(k)}, B_c^{(k)}, a_c^{(k)}, b_c^{(k)} \in \mathbb{R}$ which vary across clusters and global aggregations. These coefficients are estimated through fitting the linear functions to the values of $\Upsilon_c^{(t)}$ obtained from the previous global aggregation $\mathcal{T}_{k-1}$. These values of $\Upsilon_c^{(t)}$ from $\mathcal{T}_{k-1}$ are in turn estimated in a distributed manner through a method presented in Chapter 3.3.3.

Note that ($\boldsymbol{\mathcal{P}}$) is a non-convex and integer optimization problem. Given the parameters in Chapter 3.3.1, the solution for τ_k can be obtained via a line search over the integer values in the range of τ_k given in (3.32). Solving our optimization problem involves two steps: (i) linear regression of the constants used in (53), i.e., $A_c^{(k)}, B_c^{(k)}, a_c^{(k)}, b_c^{(k)}$ using the history of observations, and (ii) line search over the feasible integer values for τ_k. The complexity of part (i) is $\mathcal{O}(\tau_{k-1})$, since the dimension of each observant, i.e., $\Upsilon_c^{(t)}$, is one and the observations are obtained via looking back into the previous global aggregation interval. Also, the complexity of (ii) is $\mathcal{O}(\tau_{\max})$ since it is just an exhaustive search over the range of $\tau \leq \tau_{\max}$, where $\tau_{\max}$ is the maximum tolerable interval that satisfies the feasibility conditions in Chapter 3.3.1. While the optimization produces predictions of $\Gamma_c^{(t)}$ for $t \in \mathcal{T}_k$ through (3.31), the devices will later compute (3.31) at time t when the real-time estimates of $\Upsilon_c^{(t)}$ can be made through (3.36), as will be discussed next.

3.3.3 Data and Model-Related Parameters ($\delta, \zeta, \sigma^2, \Upsilon_c^{(t)}$)

We also need techniques for estimating the gradient diversity (δ, ζ), SGD noise (σ^2), and cluster parameter divergence ($\Upsilon_c^{(t)}$).

Estimation of δ, ζ, σ^2

These parameters can be estimated by the main server during model training. The server can estimate δ and ζ at each global aggregation by receiving the latest gradients from SGD at the sampled devices. σ^2 can first be estimated locally at the sampled devices, and then decided at the main server.

Specifically, to estimate δ, ζ, since the value of $\mathbf{w}^*$ is not known, we upper bound the gradient diversity in Definition 3.1.1 by introducing a new parameter δ':

$$\|\nabla \hat{F}_c(\mathbf{w}) - \nabla F(\mathbf{w})\| \leq \delta + \zeta\|\mathbf{w} - \mathbf{w}^*\| \leq \delta' + \zeta\|\mathbf{w}\|, \tag{3.35}$$

which satisfies $\delta' \geq \delta + \zeta\|\mathbf{w}^*\|$. Thus, a value of $\zeta < 2\beta$ is set, and then the value of δ' is estimated using (3.35), where the server uses the SGD gradients $\widehat{\mathbf{g}}_{n_c}^{(t_k)}$ from the sampled devices n_c at the instance of each global aggregation k, and chooses the smallest δ' such that $\|\nabla \hat{F}_c(\hat{\mathbf{w}}^{(t_k)}) - \nabla F(\hat{\mathbf{w}}^{(t_k)})\| \approx \|\widehat{\mathbf{g}}_{n_c}^{(t_k)} - \sum_{c'=1}^{N} \varrho_{c'} \widehat{\mathbf{g}}_{n_{c'}}^{(t_k)}\| \leq \delta' + \zeta\|\hat{\mathbf{w}}^{(t_k)}\| \; \forall c$.

From Assumption 3, a simple way of obtaining the value of σ^2 would be comparing the gradients from sampled devices with their full-batch counterparts. But this might be impractical if the local datasets $\mathcal{D}_{\mathrm{i}}$ are large. Thus, we propose an approach where σ^2 is computed at each device through two independent mini-batches of data. Recall $|\xi_{\mathrm{i}}|$ denotes the mini-batch size used at node i during the model training. At each instance of global aggregation, the sampled devices each select two mini-batches of size $|\xi_{\mathrm{i}}|$ and compute two SGD realizations $\mathbf{g}_1$, $\mathbf{g}_2$ from which $\widehat{\mathbf{g}}_{\mathrm{i}}^{(t_k)} = (\mathbf{g}_1 + \mathbf{g}_2)/2$. Since $\mathbf{g}_1 = \nabla F_{\mathrm{i}}(\mathbf{w}^{(t_k)}) + \mathbf{n}_1$, $\mathbf{g}_2 = \nabla F_{\mathrm{i}}(\mathbf{w}^{(t_k)}) + \mathbf{n}_2$, we use the fact that $\mathbf{n}_1$ and $\mathbf{n}_2$ are independent random variables with the same upper bound on variance σ^2, and thus $\|\mathbf{g}_1 - \mathbf{g}_2\|^2 = \|\mathbf{n}_1 - \mathbf{n}_2\|^2 \leq 2\sigma^2$, from which σ^2 can be approximated locally. These scalars are then transferred to the main server, which in turn chooses the maximum reported σ^2 from the sampled devices.

Estimation of $\Upsilon_c^{(t)}$

Based on (3.14), we propose the following approximation to estimate the value of $\Upsilon_c^{(t)}$:

$$\Upsilon_c^{(t)} = \max_{\mathrm{j},\mathrm{j}' \in \mathcal{S}_c} \|\tilde{\mathbf{w}}_{\mathrm{j}}^{(t)} - \tilde{\mathbf{w}}_{\mathrm{j}'}^{(t)}\| \approx \underbrace{\max_{\mathrm{j} \in \mathcal{S}_c} \|\tilde{\mathbf{w}}_{\mathrm{j}}^{(t)}\|}_{(a)} - \underbrace{\min_{\mathrm{j} \in \mathcal{S}_c} \|\tilde{\mathbf{w}}_{\mathrm{j}}^{(t)}\|}_{(b)}, \tag{3.36}$$

where we have used the lower bound $\|\mathbf{a} - \mathbf{b}\| \geq \|\mathbf{a}\| - \|\mathbf{b}\|$ for vectors $\mathbf{a}$ and $\mathbf{b}$, which we experimentally observe gives a better approximation of $\Upsilon_c^{(t)}$. In (3.36), (a) and (b) can be both obtained in a distributed manner through scalar message passing, where each device $\mathrm{i} \in \mathcal{S}_c$ computes $\|\tilde{\mathbf{w}}_{\mathrm{i}}^{(t)}\|$ and shares it with its neighbors $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}$. The devices update their max and min accordingly, share these updated values, and the process continues. After the rounds of message passing has exceeded the diameter of the graph, each node has the value of (a) and (b), and thus the estimate of $\Upsilon_c^{(t)}$. The server can obtain these values for $t \in \mathcal{T}_k$ from the node n_c it samples for cluster c at $t = t_k$.

3.3.4 TT-HF with Adaptive Parameter Control

The full TT-HF procedure with adaptive parameter control is summarized in Algorithm 4. The values of τ, desired ξ and T, and model characteristics μ, β are provided as inputs.

First, estimates of different parameters are initialized, the value of ϕ is determined, and the first period of model training is set (lines 2-6). Then, during the local model training intervals, in each timestep, the devices (i) compute the SGD updates, (ii) estimate the cluster model divergence, (iii) determine the number of D2D consensus rounds, and (iv) conduct the consensus process with their neighboring nodes (lines 12-16).

At global aggregation instances, the sampled devices compute their estimated local SGD noise, and transmit it along with their model parameter vector, gradient vector, and estimates of cluster parameter divergence over the previous global aggregation round to the server (lines 20-21). Then, the main server (i) updates the global model, (ii) estimates ζ, δ', σ for the step size, (iii) estimates the linear model coefficients used in (3.34), (iv) obtains the optimal length τ_{k+1} of the next local model training interval, and (v) broadcasts the updated global model, step size coefficients, local model training interval, and consensus coefficient, along with the indices of the sampled devices for the next global aggregation (line 23-29).

3.4 Numerical Evaluations

In this section, we conduct numerical experiments to verify the performance of TT-HF. After describing the setup in Chapter 3.4.1, we study model performance/convergence in Chapter 3.4.2 and the impact of our adaptive control algorithm in Chapter 3.4.3. Overall, we will see that TT-HF provides substantial improvements in training time, accuracy, and/or resource utilization compared to conventional federated learning [24], [116].

3.4.1 Experimental Setup

Network architecture. We consider a network consisting of $I = 125$ edge devices placed into $N = 25$ clusters, each with $s_c = 5$ devices placed uniformly at random in a 50 m $\times$ 50 m square field (in each cluster). The channel model and D2D network configuration are explain below.

Algorithm 4 `TT-HF` with adaptive control parameters.

```
Input: Desirable loss criterion $\xi$, length of model training $T$, maximum tolerable $\tau$,
       and model-related parameters $\beta, \mu$
Output: Global model $\hat{\mathbf{w}}^{(T)}$
// Start of initialization by the server
Initialize $\hat{\mathbf{w}}^{(0)}$ and broadcast it among the devices along with the indices $n_c$ of the sampled devices for the first global aggregation.
Initialize estimates of $\zeta \ll 2\beta, \delta', \sigma$.
Initialize $\alpha$ and $\gamma > 1/\mu$ for the step size $\eta_t = \frac{\gamma}{t+\alpha}$, where $\alpha$ is the smallest solution that satisfies the condition mentioned in Chapter 3.3.1, and $\alpha, \gamma, \xi, \tau, T$ satisfy (3.29).
Obtain $\phi^{\mathsf{max}}$ from (3.30).
Initialize $\tau_1$ randomly, where $\tau_1 \leq \tau$.
// End of initialization by the server
Initialize $t = 1,\ k = 1,\ t_0 = 0,\ t_1 = \tau_1$.
while $t \leq T$ do
    while $t \leq t_k$ do
        for $c = 1 : N$ do
            // Operation at the clusters
            Each device $\mathrm{i} \in \mathcal{S}_c$ performs a local SGD update based on (3.9) and (3.10)
             using $\widehat{\mathbf{w}}_{\mathrm{i}}^{(t-1)}$ to obtain $\widetilde{\mathbf{w}}_{\mathrm{i}}^{(t)}$.
            Devices estimate the value of $\Upsilon_c^{(t)}$ using (3.36) with distributed message passing.
            Devices compute the number of D2D communication consensus rounds
             $\Gamma_c^{(t)}$ according to (3.31).
            Devices inside the cluster conduct $\Gamma_c^{(t)}$ rounds of consensus procedure based on (3.11),
             initializing $\mathbf{z}_{\mathrm{i}}^{(0)} = \widetilde{\mathbf{w}}_{\mathrm{i}}^{(t)}$, and setting $\mathbf{w}_{\mathrm{i}}^{(t)} = \mathbf{z}_{\mathrm{i}}^{(\Gamma_c^{(t)})}$.
        end for
        if $t = t_k$ then
            // Operation at the clusters
            Each sampled device $n_c$ estimates the local SGD noise
             as described in Chapter 3.3.3.
            Each sampled devices $n_c$ sends $\mathbf{w}_{n_c}^{(t_k)}$, $\widehat{\mathbf{g}}_{n_c}^{(t_k)}$, the estimated local SGD noise,
             and the estimated values of $\Upsilon_c(t)$, $t \in \mathcal{T}_k$ to the server.
            // Operation at the server
            Compute $\widehat{\mathbf{w}}^{(t_k)}$ using (3.17).
            Set $\zeta \ll 2\beta$, and compute $\delta' = \left[\max_c\{\|\widehat{\mathbf{g}}_{n_c}^{(t_k)} - \sum_{c'=1}^{N} \varrho_{c'} \widehat{\mathbf{g}}_{n_{c'}}^{(t_k)}\| - \zeta\|\hat{\mathbf{w}}^{(t_k)}\|\}\right]^+$.
            Choose the maximum among the reported local SGD noise values as $\sigma^2$.
            Characterize $\alpha$ and $\gamma > 1/\mu$ for the step size $\eta_t = \frac{\gamma}{t+\alpha}$ according to the condition
             on $\alpha$ in Chapter 3.3.1 and (3.29), and compute $\phi^{\mathsf{max}}$ according to (3.30).
            Estimate $A_c^{(k+1)}$, $B_c^{(k+1)}$, $a_c^{(k+1)}$, and $b_c^{(k+1)}$, $\forall c$ in (3.34) via linear data fitting.
            Solve the optimization $(\boldsymbol{\mathcal{P}})$ to obtain $\tau_{k+1}$.
            Broadcast $\hat{\mathbf{w}}^{(t_k)}$ among the devices along with (i) the $n_c$ for $k+1$, (ii) $\alpha$,
             (iii) $\gamma$, (iv) $\tau_{k+1}$, and (v) $\phi$.
        end if
    end while
end while
```

Channel model: We assume that the D2D communications are conducted using orthogonal frequency division techniques, e.g., OFDMA, to reduce the interference across the devices. We consider the instantaneous channel capacity for transmitting data from node i to i′, both belonging to the same cluster c following this formula:

$$C^{(t)}_{\mathrm{i},\mathrm{i}'} = W\log_2\left(1+\frac{p^{(t)}_{\mathrm{i}}|h^{(t)}_{\mathrm{i},\mathrm{i}'}|^2}{\sigma^2}\right), \tag{3.37}$$

where $\sigma^2 = N_0W$ is the noise power, with $N_0 = -173$ dBm/Hz denoting the white noise power spectral density; $W = 1$ MHz is the bandwidth; $p^{(t)}_{\mathrm{i}} = 24$ dBm, $\forall \mathrm{i}, t$ is the transmit power; $h^{(t)}_{\mathrm{i},\mathrm{i}'}$ is the channel coefficient. We incorporate the effect of both large-scale and small scaling fading in $h^{(t)}_{\mathrm{i},\mathrm{i}'}$, given by [117], [118]:

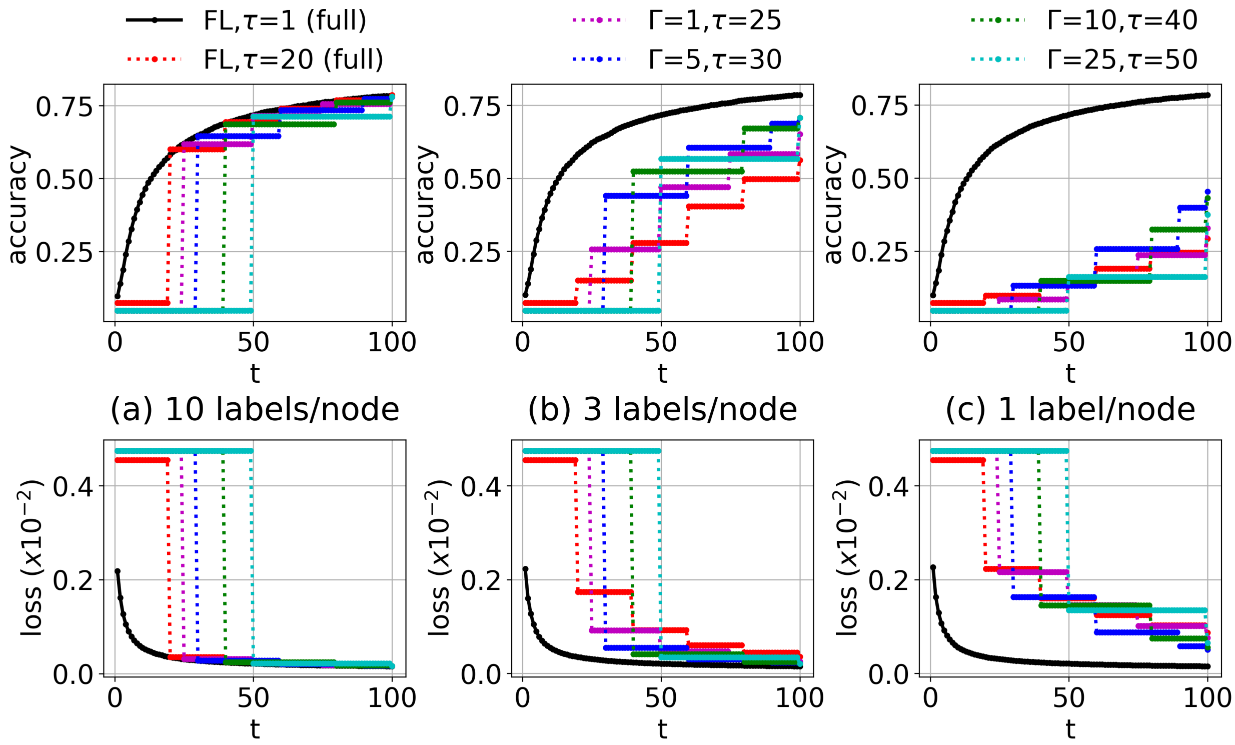
$$h^{(t)}_{\mathrm{i},\mathrm{i}'} = \sqrt{\beta^{(t)}_{\mathrm{i},\mathrm{i}'}}u^{(t)}_{\mathrm{i},\mathrm{i}'}, \tag{3.38}$$

where $\beta^{(t)}_{\mathrm{i},\mathrm{i}'}$ is the large-scale pathloss coefficient and $u^{(t)}_{\mathrm{i},\mathrm{i}'} \sim \mathcal{CN}(0,1)$ captures Rayleigh fading, varying i.i.d. over time. We assume channel reciprocity, i.e., $h^{(t)}_{\mathrm{i},\mathrm{i}'} = h^{(t)}_{\mathrm{i}',\mathrm{i}}$, for simplicity. We model $\beta^{(t)}_{\mathrm{i},\mathrm{i}'}$ as [117], [118]

$$\beta^{(t)}_{\mathrm{i},\mathrm{i}'} = \beta_0 - 10\alpha\log_{10}(d^{(t)}_{\mathrm{i},\mathrm{i}'}/d_0). \tag{3.39}$$

where $\beta_0 = -30$ dB denotes the large-scale pathloss coefficient at a reference distance of $d_0 = 1$ m, α is the path loss exponent chosen as 3.75 suitable for urban areas, and $d^{(t)}_{\mathrm{i},\mathrm{i}'}$ denotes the instantaneous Euclidean distance between the respective nodes.

D2D network configuration: We incorporate the wireless channel model explained above into our scenario to define the set of D2D neighbors and configure the cluster topologies. We assume that the nodes moves slowly so that their locations remain static during each global aggregation period, although it may change between consecutive global aggregations. We build the cluster topology based on channel reliability across the nodes quantified via the outage probability. Specifically, considering (3.37), the probability of outage upon transmitting with data rate of $R^{(t)}_{\mathrm{i},\mathrm{i}'}$ between two nodes i, i' is given by



(a) 10 labels/node (b) 3 labels/node (c) 1 label/node

(d) 10 labels/node (e) 3 labels/node (f) 1 label/node

Figure 3.2. Performance comparison between `TT-HF` and baseline methods when varying the local model training interval (τ) and the number of D2D consensus rounds (Γ). With a larger τ, `TT-HF` can still outperform the baseline federated learning [24], [116] if Γ is increased, i.e., local D2D communications can be used to offset the frequency of global aggregations. *full* implies that baseline schemes do not leverage D2D and instead require all the device to engage in uplink transmissions. SVM is used for classification.

$$\mathrm{p}_{\mathrm{i,i'}}^{\mathsf{out},(t)} = 1 - \exp\left(\frac{-(2^{R_{\mathrm{i,i'}}^{(t)}} - 1)}{\mathrm{SNR}_{\mathrm{i,i'}}^{(t)}}\right), \tag{3.40}$$

where $\mathrm{SNR}_{\mathrm{i,i'}}^{(t)} = \frac{p_{\mathrm{i}}^{(t)}|h_{\mathrm{i,i'}}^{(t)}|^2}{\sigma^2}$. To construct the graph topology of each cluster c, we create an edge between two nodes i and i′ if and only if their respective outage probability satisfies $\mathrm{p}_{\mathrm{i,i'}}^{\mathsf{out},(t)} \leq 5\%$ given a defined common data rate $R_{\mathrm{i,i'}}^{(t)} = R_c^{(t)}$, chosen as $R_c^{(t)} = 14$ Mbps. This value is used since it is large enough to neglect the effect of quantization error in digital communication of the signals, and at the same time results in connected graphs inside the clusters (numerically, we found an average degree of 2 nodes in each cluster). After creating the topology based on the large-scale pathloss and outage probability requirements, we model outages during the consensus phase as follows: if the instantaneous channel capacity (given by (3.37), which captures the effect of fast fading) on an edge drops below $R_c^{(t)}$, outage occurs, so that the packet is lost and the model update is not received at the respective receiver. Therefore, although nodes are assumed to be static during each global aggregation period, the instantaneous cluster topology, i.e., the communication configuration among the nodes, changes with respect to every local SGD iteration in a model training interval due to outages.

Given a communication graph we choose $d_c = 1/8$ to form the consensus iteration at each node i as $\mathbf{z}_{\mathrm{i}}^{(t'+1)} = \mathbf{z}_{\mathrm{i}}^{(t')} + d_c \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}}} (\mathbf{z}_{\mathrm{j}}^{(t')} - \mathbf{z}_{\mathrm{i}}^{(t')})$ (refer to the discussion provided after Assumption 2). Note that given d_c, broadcast by the server at the beginning of each global aggregation, each node can conduct D2D communications and local averaging without any global coordination.

Datasets. We consider MNIST [119] and Fashion-MNIST (F-MNIST) [120], two datasets commonly used in image classification tasks. Each dataset contains 70K images (60K for training, 10K for testing), where each image is one of 10 labels of hand-written digits and fashion products, respectively. For brevity, we present the results for MNIST here, and refer the reader to Appendix D in our technical report [108] for FMNIST; the results are qualitatively similar.

Data distributions. To simulate varying degrees of statistical data heterogeneity among the devices, we divide the datasets into the devices' local $\mathcal{D}_{\mathrm{i}}$ in three ways: (a) *extreme non-i.i.d.*, where each local dataset has only data points from a single label; (b) *moderate non-i.i.d.*, where each local dataset contains datapoints from three of the 10 labels; and (c) *i.i.d.*, where each local dataset has datapoints covering all 10 labels. In each case, $\mathcal{D}_{\mathrm{i}}$ is selected randomly (without replacement) from the full dataset of labels assigned to device i.

ML models. We consider loss functions from two different ML classifiers: regularized (squared) support vector machines (SVM) and a fully connected neural network (NN). In both cases, we use the standard implementations in PyTorch which results in a model dimension of $M = 7840$ on MNIST. Note that the SVM satisfies Assumption 1, while the NN does not. The numerical results obtained for both classifiers are qualitatively similar. Thus,

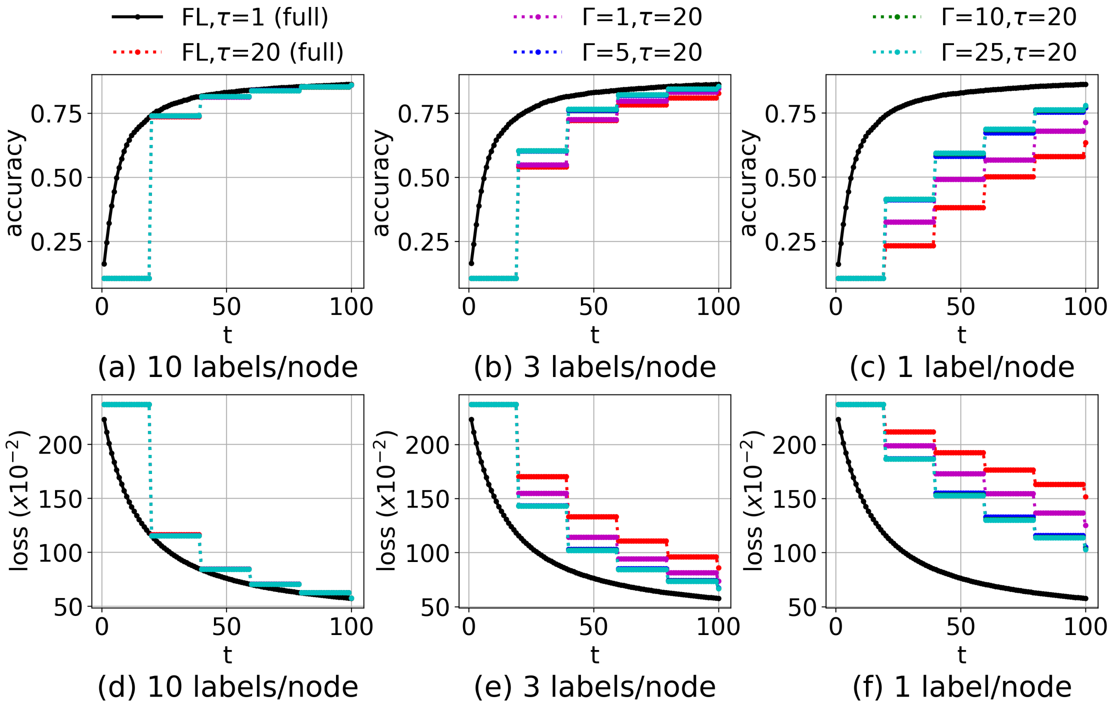


Figure 3.3. Performance comparison between TT-HF and baseline methods when varying the number of D2D consensus rounds (Γ). Under the same period of local model training (τ), increasing Γ results in a considerable improvement in the model accuracy/loss over time as compared to the current art [24], [116] when data is non-i.i.d. *full* implies that baseline schemes do not leverage D2D and instead require all the device to engage in uplink transmissions. NN is used for classification.

for brevity, we show a selection of results for each classifier here, and refer the reader to Appendix D in our technical report [108] for the extensive simulation results on both classifiers, where we also explain the implementation of our control algorithm for non-convex loss functions. The SVM uses a linear kernel, and the weights initialization follows a uniform distribution, with mean and variance calculated according to [121]. All of our implementations can be accessed at https://github.com/shams-sam/TwoTimeScaleHybridLearning.

3.4.2 TT-HF Model Training Performance and Convergence

One of the main premises of TT-HF is that cooperative consensus procedure within clusters during the local model training interval can (i) preserve model performance while reducing the required frequency of global aggregations and/or (ii) increase the model training accuracy, especially when statistical data heterogeneity is present across the devices. Our first set of experiments seek to validate these facts:

Local consensus reducing global aggregation frequency

In Fig. 3.2, we compare the performance of TT-HF for increased local model training intervals τ against the current federated learning algorithms that do not exploit local D2D model consensus procedure. The baselines both assume full device participation (i.e., all devices upload their local model to the server at each global aggregation), and thus are 5x more uplink resource-intensive at each aggregation. One baseline conducts global aggregations after each round of training ($\tau = 1$), and the other, based on [24], has local update intervals of 20 ($\tau = 20$). Recall that longer local training periods are desirable to reduce the frequency of communication between devices and the main server. We conduct consensus after every $t = 5$ time instances, and increase Γ as τ increases. The $\tau = 1$ baseline is an upper bound on the achievable performance since it replicates centralized model training.

Fig. 3.2 confirms that TT-HF can still outperform the baseline FL with $\tau = 20$ when the frequency of global aggregations is decreased: in other words, increasing τ can be counteracted with a higher degree of local consensus procedure $\Gamma_c^{(t)} = \Gamma$, $\forall c, t$. Considering the moderate non-i.i.d. plots ((b) and (e)), we also see that the jumps in global model performance, while less frequent, are substantially larger for TT-HF than the baseline. This result shows that D2D communications can reduce reliance on the main server for a more distributed model training process. It can also be noted that TT-HF achieves this performance gain despite the communication impairments, i.e., packet lost due to fast fading, that we assumed in D2D communications. This implies the robustness of TT-HF to imperfect D2D communications among the devices.

D2D enhancing ML model performance

In Fig. 3.3, we compare the performance of TT-HF with the baseline methods, where we set $\tau_k = \tau = 20$ and conduct a fixed number of D2D rounds in clusters after every 5 time instances, i.e., $\Gamma_c^{(t)} = \Gamma$ for different values of Γ. Fig. 3.3 verifies that local D2D

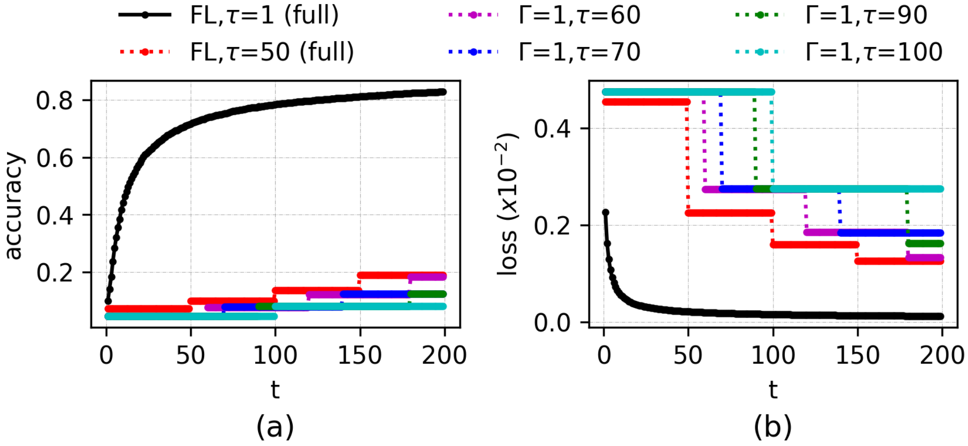


Figure 3.4. Performance of TT-HF in the extreme non-i.i.d. case for the setting in Fig. 3.2 when Γ is small and the local model training interval length is increased substantially. TT-HF exhibits poor convergence behavior when τ exceeds a certain value, due to model dispersion. SVM is used for classification.

communications can significantly boost the performance of ML model training. Specifically, when the data distributions are moderate non-i.i.d ((b) and (e)) or extreme non-i.i.d. ((c) and (f)), we see that increasing Γ improves the trained model accuracy/loss substantially from FL with $\tau = 20$. It also reveals that there is a diminishing reward of increasing Γ as the performance of TT-HF approaches that of FL with $\tau = 1$. Finally, we observe that the gains obtained through D2D communications are only present when the data distributions across the nodes are non-i.i.d., as compared to the i.i.d. scenario ((a) and (d)), which emphasizes the purpose of TT-HF for handling statistical heterogeneity. This result further shows the applicability of TT-HF to non-convex classifiers such as NN.

Convergence behavior

Recall that the upper bound on convergence in Theorem C.2.1 is dependent on the expected model dispersion $A^{(t)}$ and the consensus error $\epsilon^{(t)}$ across clusters. For the settings in Figs. 3.3&3.2, increasing the local model training period τ and decreasing the consensus rounds Γ will result in increased $A^{(t)}$ and $\epsilon^{(t)}$, respectively, for a given t. In Fig. 3.4, we show that TT-HF suffers from poor convergence behavior in the extreme non-i.i.d. case when the period of local descents τ are excessively prolonged, similar to the baseline FL when $\tau = 50$ [24]. This further emphasizes the importance of Algorithm 4 tuning these parameters around Theorem 3.2.2's result.

3.4.3 TT-HF with Adaptive Parameter Control

We turn now to evaluating the efficacy and analyzing the behavior of TT-HF under parameter tuning from Algorithm 4.

Improved resource efficiency compared with baselines

Fig. 3.5 compares the performance of TT-HF under our control algorithm with the two baselines: (i) FL with full device participation and $\tau = 1$ (from Chapter 3.4.2), and (ii) FL with $\tau = 20$ but only one device sampled from each cluster for global aggregations.[6] The result is shown under different ratios of delays $\frac{\Delta_{\text{D2D}}}{\Delta_{\text{Glob}}}$ and different ratios of energy consumption $\frac{E_{\text{D2D}}}{E_{\text{Glob}}}$ between D2D communications and global aggregations.[7] Three metrics are shown: (a) total cost based on the objective of $(\boldsymbol{\mathcal{P}})$, (b) total energy consumed, and (c) total delay experienced up to the point where 75% of peak accuracy is reached.

Overall, in (a), we see that TT-HF (depicted through the bars) outperforms the baselines (depicted through the horizontal lines) substantially in terms of total cost, by at least 75% in each case. In (b), we observe that for smaller values of $E_{\text{D2D}}/E_{\text{Glob}}$, TT-HF lowers the overall power consumption, but after the D2D energy consumption reaches a certain threshold, it does not result in energy savings anymore. The same impact can be observed regarding the delay from (c), i.e., once $\frac{\Delta_{\text{D2D}}}{\Delta_{\text{Glob}}} \approx 0.1$ there is no longer an advantage in terms of delay. Ratios of 0.1 for either of these metrics, however, is significantly larger than what is being observed in 5G networks [109], [110], indicating that TT-HF would be effective in practical systems.

[6] ↑The baseline of FL, $\tau = 20$ with full participation is omitted because it results in very poor costs.

[7] ↑These plots are generated for some typical ratios observed in the literature. For example, a similar data rate in D2D and uplink transmission can be achieved via typical values of transmit powers of 10dbm in D2D mode and 24dbm in uplink mode [109], [110], which coincides with a ratio of $E_{\text{D2D}}/E_{\text{Glob}} = 0.04$. In practice, the actual values are dependent on many environmental factors.

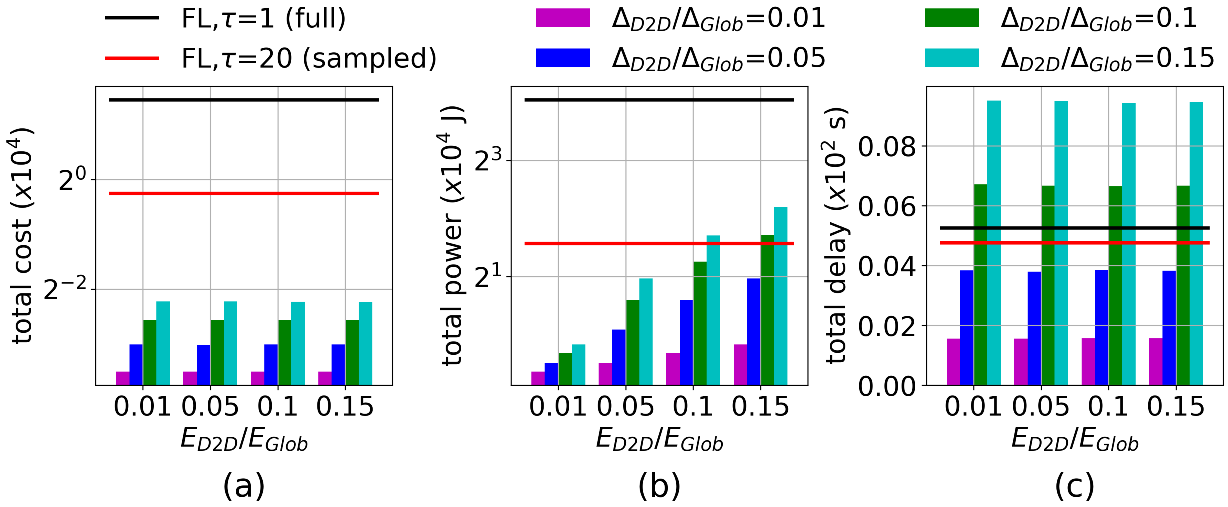


Figure 3.5. Comparing total (a) cost, (b) power, and (c) delay metrics from the optimization objective in ($\boldsymbol{\mathcal{P}}$) achieved by `TT-HF` versus baselines upon reaching 75% of peak accuracy, for different configurations of delay and energy consumption. `TT-HF` obtains a significantly lower total cost in (a). (b) and (c) demonstrate the region under which `TT-HF` attains energy savings and delay gains. SVM is used for classification.

Impact of design choices on local model training interval

We are also interested in how the design weights c_1, c_2, c_3 in ($\boldsymbol{\mathcal{P}}$) affect the behavior of the control algorithm. In Fig. 3.6, we plot the value of τ_2, i.e., the length of the second local model training interval, for different configurations of c_1, c_2 and c_3.[8] The maximum tolerable value of τ is assumed to be 40. As we can see, increasing c_1 and c_2 – which elevates the priority on minimizing energy consumption and delay, respectively – results in a longer local model training interval, since D2D communication is more efficient. On the other hand, increasing c_3 – which prioritizes the global model convergence rate – results in a quicker global aggregation.

3.4.4 Main Takeaways

Data heterogeneity in local dataset across local devices can result in considerable performance degradation of federated learning algorithms. In this case, longer local update

[8] ↑The specific ranges of values chosen gives comparable objective terms (a), (b), and (c) in ($\boldsymbol{\mathcal{P}}$).

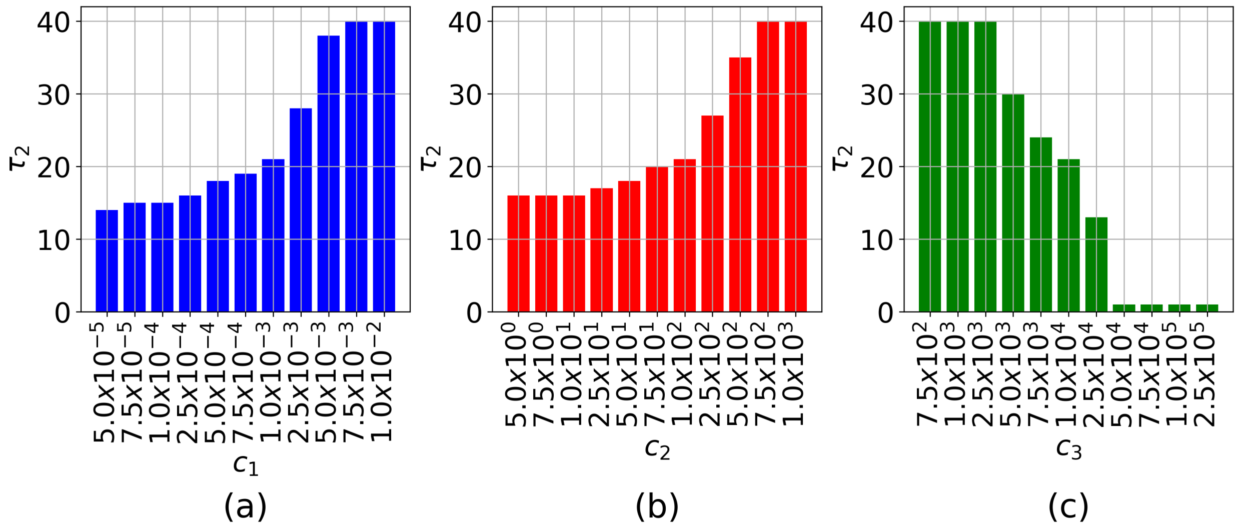


Figure 3.6. Value of the second local model training interval obtained through $(\boldsymbol{\mathcal{P}})$ for different configurations of weighing coefficients c_1, c_2, c_3 (default $c_1 = 10^{-3}, c_2 = 10^2, c_3 = 10^4$). Higher weight on energy and delay (larger c_1 and c_2) prolongs the local training period, while higher weight on the global model loss (larger c_3) decreases the length, resulting in more rapid global aggregations.

periods will result in models that are significantly biased towards local datasets and degrade the convergence speed of the global model and the resulting model accuracy. By blending federated aggregations with cooperative D2D consensus procedure among local device clusters in `TT-HF`, we effectively decrease the bias of the local models to the local datasets and speed up the convergence at a lower cost (i.e., utilizing low power D2D communications to reduce the frequency of performing global aggregation via uplink transmissions). Due to the low network cost in performing D2D transmission, `TT-HF` provides a practical solution for federated learning to achieve faster convergence or to prolong the local model training interval, leading to delay and energy consumption savings.

Although we develop our algorithm based on federated learning with vanilla SGD local optimizer, our method can benefit other counterparts in the literature. This is due to the fact that, intuitively, conducting D2D communications via the method proposed on this paper reduces the local bias of the nodes' models to their local datasets, which is one of the main challenges faced in federated learning. In Appendix D of [108] we conduct some preliminary experiment to show the impact of our method on FedProx [122].

4. CONCLUSION AND FUTURE WORK

We started by introducing two possible solutions to solve problem of communication latency in federated learning: (i) learning private representations of data to enable its sharing, and (ii) reducing the communication latency by minimizing the long-range communication requirements.

To address the first, we developed the first methodology for generalized and distributable PRL. EIGAN accounts for the presence of multiple allies and adversaries with potentially overlapping objectives, and D-EIGAN addresses privacy concerns and resource constraints in scenarios with decentralized data. We proved that for an optimal encoding, the adversary's output from EIGAN follows a uniform distribution, and that dependencies between ally and adversary interests requires careful balancing of objectives in encoder optimization. Our experiments showed that EIGAN outperforms six baselines in jointly optimizing predictivity and privacy on different datasets and system settings. They also showed that D-EIGAN achieves comparable performance to EIGAN with different numbers of training nodes and as the training parameters vary to account for communication constraints.

We next proposed `TT-HF` to tackle the latter. `TT-HF` improves the efficiency of federated learning in D2D-enabled wireless networks by augmenting global aggregations with cooperative consensus procedure among device clusters. We conducted a formal convergence analysis of `TT-HF`, resulting in a bound which quantifies the impact of gradient diversity, consensus error, and global aggregation periods on the convergence behavior. Using this bound, we characterized a set of conditions under which `TT-HF` is guaranteed to converge sublinearly with rate of $\mathcal{O}(1/t)$. Based on these conditions, we developed an adaptive control algorithm that actively tunes the device learning rate, cluster consensus rounds, and global aggregation periods throughout the training process. Our experimental results demonstrated the robustness of `TT-HF` against data heterogeneity among edge devices, and its improvement in trained model accuracy, training time, and/or network resource utilization in different scenarios compared to the current art.

There are several avenues for future work. To further enhance the flexibility of `TT-HF`, one may consider (i) heterogeneity in computation capabilities across edge devices, (ii) different communication delays from the clusters to the server, and (iii) wireless interference caused by D2D communications. Furthermore, using the set of new techniques we provided to conduct convergence analysis in this paper, we aim to extend our convergence analysis to non-convex settings in future work. This includes obtaining the conditions under which approaching a stationary point of the global loss function is guaranteed, and the rate under which the convergence is achieved.

REFERENCES

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single Shot Multibox Detector,” in *European Conference on Computer Vision (ECCV)*, Springer, 2016, pp. 21–37.

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.

[3] S. Ghosh, N. Das, I. Das, and U. Maulik, “Understanding deep learning techniques for image segmentation,” *ACM Comput. Surveys (CSUR)*, vol. 52, no. 4, pp. 1–35, 2019.

[4] K. Kang, W. Ouyang, H. Li, and X. Wang, “Object detection from video tubelets with convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vision Pattern Recog. (CVPR)*, 2016, pp. 817–825.

[5] Y. Goldberg, “Neural network methods for natural language processing,” *Synthesis Lectures on Human Language Technologies*, vol. 10, no. 1, pp. 1–309, 2017.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv preprint arXiv:1907.11692*, 2019.

[7] M. Chiang and T. Zhang, “Fog and iot: An overview of research opportunities,” *IEEE Internet Thing J.*, vol. 3, no. 6, pp. 854–864, 2016.

[8] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language Models are Few-Shot Learners,” in *Neural Information Processing Systems (NeurIPS)*, 2020.

[11] *Turing-NLG: A 17-billion-parameter language model by Microsoft*, Microsoft, 2020. [Online]. Available: https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/.

[12] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, "Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vision Pattern Recog. (CVPR) Workshops*, 2017, pp. 129–137.

[13] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[14] R. Shokri and V. Shmatikov, "Privacy-preserving Deep Learning," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015.

[15] S. S. Azam, T. Kim, S. Hosseinalipour, C. Brinton, C. Joe-Wong, and S. Bagchi, "Towards generalized and distributed privacy-preserving representation learning," *arXiv preprint arXiv:2010.01792*, 2020.

[16] J. Konečn, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated Learning: Strategies for Improving Communication Efficiency," in *Neural Information Processing Systems (NeurIPS)*, 2016.

[17] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization," in *Neural Information Processing Systems (NeurIPS)*, 2020.

[18] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication Efficient Learning of Deep Networks from Decentralized Data," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[19] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[20] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys & Tuts.*, 2020.

[21] M. Bennis, M. Debbah, K. Huang, and Z. Yang, "Guest editorial: Communication technologies for efficient edge learning," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 12–13, 2020. DOI: 10.1109/MCOM.2020.9311909.

[22] Y. Tu, Y. Ruan, S. Wang, S. Wagle, C. G. Brinton, and C. Joe-Wang, "Network-aware optimization of distributed learning for fog computing," *arXiv preprint arXiv:2004.08488*, 2020.

[23] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *arXiv preprint arXiv:1911.02417*, 2019.

[24] S. Wang, T. Tuor, T. Salonidis, *et al.*, "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," *IEEE Journal on Selected Areas in Communications (JSAC)*, 2019.

[25] F. P.-C. Lin, C. G. Brinton, and N. Michelusi, "Federated learning with communication delay in edge networks," in *Proc. IEEE Int. Glob. Commun. Conf. (GLOBECOM)*, 2020, pp. 1–6. DOI: 10.1109/GLOBECOM42002.2020.9322592.

[26] S. Hosseinalipour, C. G. Brinton, V. Aggarwal, H. Dai, and M. Chiang, "From federated to fog learning: Distributed machine learning over heterogeneous wireless networks," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 41–47, 2020. DOI: 10.1109/MCOM.001.2000410.

[27] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Wireless communications for collaborative federated learning," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 48–54, 2020.

[28] S. Chakraborty, K. R. Raghavan, M. P. Johnson, and M. B. Srivastava, "A Framework for Context-aware Privacy of Sensor Data on Mobile Systems," in *Mobile Computing Systems and Applications*, 2013.

[29] N. Saleheen, S. Chakraborty, N. Ali, M. M. Rahman, S. M. Hossain, R. Bari, E. Buder, M. Srivastava, and S. Kumar, "mSieve: Differential Behavioral Privacy in Time Series of Mobile Sensor Data," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016.

[30] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in *Theory of Cryptography*, 2006.

[31] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *IEEE International Conference on Data Engineering*, 2007.

[32] A. C. Y. Tossou and C. Dimitrakakis, "Achieving Privacy in the Adversarial Multi-armed Bandit," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[33] K. E. Henry, D. N. Hager, P. J. Pronovost, and S. Saria, "A Targeted Real-Time Early Warning Score (TREWScore) for Septic Shock," *Science Translational Medicine*, vol. Vol. 7, 2015.

[34] S. S. Azam, M. Raju, V. Pagidimarri, and V. C. Kasivajjala, "CASCADENET: An LSTM based Deep Learning Model for Automated ICD-10 Coding," in *Future of Information and Communication Conference (FICC)*, 2019.

[35] T.-Y. Yang, C. Brinton, P. Mittal, M. Chiang, and A. Lan, “Learning Informative and Private Representations via Generative Adversarial Networks,” in *IEEE International Conference on Big Data*, 2018.

[36] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial Training of Neural Networks,” *Journal of Machine Learning Research (JMLR)*, vol. Vol. 17, 2016.

[37] P. C. Roy and V. N. Boddeti, “Mitigating Information Leakage in Image Representations: A Maximum Entropy Approach,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[38] B. Sadeghi and V. N. Boddeti, “Imparting fairness to pre-trained biased representations,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

[39] M. Bertran, N. Martinez, A. Papadaki, Q. Qiu, M. Rodrigues, G. Reeves, and G. Sapiro, “Adversarially Learned Representations for Information Obfuscation and Inference,” in *International Conference on Machine Learning (ICML)*, 2019.

[40] S. Hosseinalipour, S. S. Azam, C. G. Brinton, N. Michelusi, V. Aggarwal, D. J. Love, and H. Dai, “Multi-Stage Hybrid Federated Learning over Large-Scale Wireless Fog Networks,” *arXiv preprint 2007.09511*, 2020.

[41] B. Chatterjee, D.-H. Seo, S. Chakraborty, S. Avlani, X. Jiang, H. Zhang, M. Abdallah, N. Raghunathan, C. Mousoulis, A. Shakouri, S. Bagchi, and S. Sen, “Context-Aware Collaborative Intelligence with Spatio-Temporal In-Sensor-Analytics for Efficient Communication in a Large-Area IoT Testbed,” *IEEE Internet of Things Journal*, 2020.

[42] D. V. Dimitrov, “Medical Internet of Things and Big Data in Healthcare,” *Healthcare Informatics Research*, vol. 22, no. 3, p. 156, 2016.

[43] N. H. Tran, W. Bao, A. Zomaya, N. M. NH, and C. S. Hong, “Federated learning over wireless networks: Optimization model design and analysis,” in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2019, pp. 1387–1395.

[44] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, “A joint learning and communications framework for federated learning over wireless networks,” *arXiv preprint arXiv:1909.07972*, 2019.

[45] K. Yang, T. Jiang, Y. Shi, and Z. Ding, “Federated learning via over-the-air computation,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, 2020.

[46] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated Multi-task Learning," in *Neural Information Processing Systems (NeurIPS)*, 2017.

[47] L. Corinzia and J. M. Buhmann, "Variational federated multi-task learning," *arXiv preprint arXiv:1906.06268*, 2019.

[48] R. Li, F. Ma, W. Jiang, and J. Gao, "Online federated multitask learning," in *Proc. Int. Conf. Big Data*, 2019, pp. 215–220.

[49] Y. Jiang, J. Konečn, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *arXiv preprint arXiv:1909.12488*, 2019.

[50] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020.

[51] S. A. Rahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet Things J.*, 2020.

[52] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[53] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," *arXiv preprint arXiv:1910.14425*, 2019.

[54] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[55] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Federated learning with quantized global model updates," *arXiv preprint arXiv:2006.10672*, 2020.

[56] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and Communication-Efficient Federated Learning from Non-Iid Data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. Vol. 31, 2019.

[57] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, "Hybrid-fl for wireless networks: Cooperative learning mechanism using non-iid data," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–7.

[58] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, and C. G. Brinton, "Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation," *arXiv preprint arXiv:2101.00787*, 2021.

[59] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[60] H. Xing, O. Simeone, and S. Bi, "Decentralized federated learning via SGD over wireless D2D networks," in *IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2020, pp. 1–5. DOI: 10.1109/SPAWC48557.2020.9154332.

[61] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive iot networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4641–4654, 2020. DOI: 10.1109/JIOT.2020.2964162.

[62] C. Hu, J. Jiang, and Z. Wang, "Decentralized federated learning: A segmented gossip approach," *arXiv preprint arXiv:1908.07782*, 2019.

[63] A. Lalitha, O. C. Kilinc, T. Javidi, and F. Koushanfar, "Peer-to-peer federated learning on graphs," *arXiv preprint arXiv:1901.11173*, 2019.

[64] D. Yuan, S. Xu, and H. Zhao, "Distributed primal–dual subgradient method for multiagent optimization via consensus algorithms," *IEEE Trans. Syst. Man Cybernetics*, vol. 41, no. 6, pp. 1715–1724, 2011.

[65] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, 2014.

[66] C. .-S. Lee, N. Michelusi, and G. Scutari, "Finite rate distributed weight-balancing and average consensus over digraphs," *IEEE Trans Autom. Control.*, pp. 1–1,

[67] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. & Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.

[68] N. Raval, A. Machanavajjhala, and L. P. Cox, "Protecting Visual Secrets using Adversarial Nets," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[69] Z. Wu, Z. Wang, Z. Wang, and H. Jin, "Towards Privacy-preserving Visual Recognition via Adversarial training: A pilot study," in *European Conference on Computer Vision (ECCV)*, 2018.

[70] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, "The Variational Fair Autoencoder," in *International Conference on Learning Representations (ICLR)*, 2016.

[71] Q. Xie, Z. Dai, Y. Du, E. Hovy, and G. Neubig, "Controllable Invariance through Adversarial Feature Learning," in *Neural Information Processing Systems (NeurIPS)*, 2017.

[72] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning Fair Representations," in *International Conference on Machine Learning (ICML)*, 2013.

[73] A. Tripathy, Y. Wang, and P. Ishwar, "Privacy-Preserving Adversarial Networks," in *IEEE Allerton Conference on Communication, Control, and Computing*, 2019.

[74] J. Jordon, J. Yoon, and M. van der Schaar, "PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees," in *International Conference on Learning Representations (ICLR)*, 2018.

[75] J. Xu, X. Ren, J. Lin, and X. Sun, "Diversity-Promoting GAN: A Cross-Entropy Based Generative Adversarial Network for Diversified Text Generation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[76] S. J. Oh, R. Benenson, M. Fritz, and B. Schiele, "Faceless Person Recognition: Privacy Implications in Social Media," in *European Conference on Computer Vision (ECCV)*, 2016.

[77] H. Edwards and A. Storkey, "Censoring Representations with an Adversary," in *International Conference on Learning Representations (ICLR)*, 2016.

[78] M. J. Kusner, J. Loftus, C. Russell, and R. Silva, "Counterfactual Fairness," in *Neural Information Processing Systems (NeurIPS)*, 2017.

[79] D. Madras, E. Creager, T. Pitassi, and R. Zemel, "Learning Adversarially Fair and Transferable Representations," in *International Conference on Machine Learning (ICML)*, 2018.

[80] S. S. Azam, T. Kim, S. Hosseinalipour, C. Joe-Wong, S. Bagchi, and C. Brinton, "A generalized and distributable generative model for private representation learning," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[81] R. Bassily, S. Moran, I. Nachum, J. Shafer, and A. Yehudayoff, "Learners that use little information," in *Algorithmic Learning Theory*, 2018.

[82] M. Feder and N. Merhav, "Relations between Entropy and Error Probability," vol. Vol. 40, 1994.

[83] M. Abadi and D. G. Andersen, "Learning to Protect Communications with Adversarial Neural Cryptography," *arXiv preprint*, vol. *arXiv:1610.06918*, 2016.

[84] C. Huang, P. Kairouz, and L. Sankar, "Generative Adversarial Privacy: A Data-Driven Approach to Information-Theoretic Privacy," in *Asilomar Conference on Signals, Systems, and Computers*, 2018.

[85] J. Adler and S. Lunz, “Banach Wasserstein GAN,” in *Neural Information Processing Systems (NeurIPS)*, 2018.

[86] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[87] Y. Huang, Y. Su, S. Ravi, Z. Song, S. Arora, *et al.*, “Privacy-Preserving Learning via Deep Net Pruning,” *arXiv preprint*, vol. *2003.01876*, 2020.

[88] M. Fredrikson, S. Jha, and T. Ristenpart, “Model Inversion Attacks that exploit Confidence Information and Basic Countermeasures,” in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015.

[89] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership Inference Attacks against Machine Learning Models,” in *IEEE Symposium on Security and Privacy*, 2017.

[90] Y. LeCun and C. Cortes, *MNIST Handwritten Digit Database*, 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/.

[91] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, *et al.*, “MIMIC-III, A Freely Accessible Critical Care Database,” *Scientific Data*, vol. 3, 2016.

[92] D. Dua and C. Graff, *UCI Machine Learning Repository*, 2017. [Online]. Available: http://archive.ics.uci.edu/ml.

[93] H. Ng and S. Winkler, “A Data-Driven Approach to Cleaning Large Face Datasets,” in *IEEE International Conference on Image Processing (ICIP)*, 2014.

[94] V. Nair and G. E. Hinton, “Rectified Linear Units improve Restricted Boltzmann Machines,” in *International Conference on Machine Learning (ICML)*, 2010.

[95] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research (JMLR)*, vol. Vol. 15, 2014.

[96] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional Networks for Biomedical Image Segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[97] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[98] H. Abdi and L. J. Williams, "Principal Component Analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. Vol. 2, 2010.

[99] M. A. Kramer, "Non-Linear Principal Component Analysis using Auto-Associative Neural Networks," *AIChE Journal*, vol. Vol. 37, 1991.

[100] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Neural Information Processing Systems (NeurIPS)*, 2019.

[101] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[102] F. P.-C. Lin, S. Hosseinalipour, S. S. Azam, C. G. Brinton, and N. Michelusi, "Semi-decentralized federated learning with cooperative d2d local model aggregations," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 39, no. 12, pp. 3851–3869, 2021. DOI: 10.1109/JSAC.2021.3118344.

[103] M. P. Friedlander and M. Schmidt, "Hybrid deterministic-stochastic methods for data fitting," *SIAM J. Sci. Comput.*, vol. 34, no. 3, A1380–A1405, 2012.

[104] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, 2020.

[105] C. T. Dinh, N. H. Tran, M. N. H. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, 2021. DOI: 10.1109/TNET.2020.3035770.

[106] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3643–3658, 2021. DOI: 10.1109/TWC.2021.3052681.

[107] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time minimization of federated learning over wireless networks," in *in IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6. DOI: 10.1109/ICC40277.2020.9148815.

[108] F. P.-C. Lin, S. Hosseinalipour, S. S. Azam, C. G. Brinton, and N. Michelusi, "Semi-decentralized federated learning with cooperative D2D local model aggregations," *arXiv preprint arXiv:2103.10481*, 2021.

[109] M. Hmila, M. Fernández-Veiga, M. Rodrıguez-Pérez, and S. Herrerıa-Alonso, “Energy efficient power and channel allocation in underlay device to multi device communications,” *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5817–5832, 2019.

[110] S. Dominic and L. Jacob, “Joint resource block and power allocation through distributed learning for energy efficient underlay D2D communication with rate guarantee,” *Comput. Commun.*, 2020.

[111] A. Zhang and X. Lin, “Security-aware and privacy-preserving D2D communications in 5G,” *IEEE Netw.*, vol. 31, no. 4, pp. 70–77, 2017. DOI: 10.1109/MNET.2017.1600290.

[112] S. Zhang, J. Liu, H. Guo, M. Qi, and N. Kato, “Envisioning device-to-device communications in 6G,” *IEEE Netw.*, vol. 34, no. 3, pp. 86–91, 2020. DOI: 10.1109/MNET.001.1900652.

[113] T. Richardson and S. Kudekar, “Design of low-density parity check codes for 5g new radio,” *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 28–34, 2018. DOI: 10.1109/MCOM.2018.1700839.

[114] T. Sery, N. Shlezinger, K. Cohen, and Y. C. Eldar, “Over-the-air federated learning from heterogeneous data,” *IEEE Trans. Signal Process.*, pp. 1–1, 2021.

[115] S. Bubeck *et al.*, “Convex optimization: Algorithms and complexity,” *Found. Trends® Machine Learn.*, vol. 8, no. 3-4, pp. 231–357, 2015.

[116] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, “On the convergence of FedAvg on non-iid data,” *arXiv:1907.02189*, 2019.

[117] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[118] J. Kim, S. Hosseinalipour, T. Kim, D. J. Love, and C. G. Brinton, “Multi-IRS-assisted multi-cell uplink MIMO communications under imperfect CSI: A deep reinforcement learning approach,” in *IEEE Int. Conf. Commun. Workshop (ICC WKSH)*, 2021, pp. 1–7. DOI: 10.1109/ICCWorkshops50388.2021.9473585.

[119] L. Yan, C. Corinna, and C. J. Burges. (). “The MNIST dataset of handwritten digits,” [Online]. Available: http://yann.lecun.com/exdb/mnist/. (accessed: 07.07.2021).

[120] H. Xiao, K. Rasul, and R. Vollgraf. (). “Fashion-MNIST,” [Online]. Available: https://github.com/zalandoresearch/fashion-mnist. (accessed: 07.07.2021).

[121] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1026–1034.

[122] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "On the convergence of federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, vol. 3, 2018.

A. SKETCH OF THE PROOFS

A.1 Sketch of the Proof of Proposition 3.2.1

The complete proof is contained in Appendix C.1 of our online technical report [108]. We break down the proof into three parts. In Part I, we find the relationship between $\|\bar{\mathbf{w}}^{(t)}-\mathbf{w}^*\|$ and $\sum_{c=1}^{N}\varrho_c\|\bar{\mathbf{w}}_c^{(t)}-\bar{\mathbf{w}}^{(t)}\|$, which forms a coupled dynamic system that we then solve in Part II. Finally, Part III draws the connection between $A^{(t)}$ and the solution of the coupled dynamic system, which yields the bound on $A^{(t)}$. A summary of the steps is given below:

Part I. Finding the relationship between $\|\bar{\mathbf{w}}^{(t)}-\mathbf{w}^*\|$ *and* $\sum_{c=1}^{N}\varrho_c\|\bar{\mathbf{w}}_c^{(t)}-\bar{\mathbf{w}}^{(t)}\|$: Using the definition of $\bar{\mathbf{w}}^{(t+1)}$ from Definition 3.2.1, we have, $\forall t\in\mathcal{T}_k$,

$$\bar{\mathbf{w}}_c^{(t+1)}=\bar{\mathbf{w}}_c^{(t)}-\eta_t\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})-\eta_t\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\mathbf{n}_{\mathrm{j}}^{(t)},\tag{A.1}$$

$$\bar{\mathbf{w}}^{(t+1)}=\bar{\mathbf{w}}^{(t)}-\eta_t\sum_{d=1}^{N}\varrho_d\frac{1}{s_d}\sum_{\mathrm{j}\in\mathcal{S}_d}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})-\eta_t\sum_{d=1}^{N}\varrho_d\frac{1}{s_d}\sum_{\mathrm{j}\in\mathcal{S}_d}\mathbf{n}_{\mathrm{j}}^{(t)}.\tag{A.2}$$

Using the definitions of $\bar{\mathbf{w}}_c^{(t)}$ and $\bar{\mathbf{w}}^{(t)}$, Assumption 1, Condition 1, Definition 3.1.1, Assumption 3 and Fact 1 (see Appendix C.6), and noting that $\eta_t\leq\frac{\mu}{\beta^2}$, we get the following for $t\in\mathcal{T}_k$, after performing some algebraic manipulations:

$$\mathbf{x}^{(t+1)}\leq\Big(\mathbf{I}+\eta_t\beta\mathbf{B}\Big)\mathbf{x}^{(t)}+\eta_t\beta\mathbf{z},\tag{A.3}$$

where

$$\mathbf{x}^{(t)}=\begin{bmatrix}\sqrt{\beta\mathbb{E}[(\sum_{c=1}^{N}\varrho_c\|\bar{\mathbf{w}}_c^{(t)}-\bar{\mathbf{w}}^{(t)}\|)^2]}\\ \sqrt{\beta\mathbb{E}[\|\bar{\mathbf{w}}^{(t)}-\mathbf{w}^*\|^2]}\end{bmatrix},$$

initialized as $\mathbf{x}^{(t_{k-1})}=\mathbf{e}_2\sqrt{\beta}\|\bar{\mathbf{w}}^{(t_{k-1})}-\mathbf{w}^*\|$ at the start of training interval $\mathcal{T}_k$, and we have defined $\mathbf{e}_1=\begin{bmatrix}1 & 0\end{bmatrix}^\top$, $\mathbf{e}_2=\begin{bmatrix}0 & 1\end{bmatrix}^\top$,

$$\mathbf{z}=\begin{bmatrix}2 & 1\end{bmatrix}^\top\Big(\frac{\sigma}{\sqrt{\beta}}+\sqrt{\beta}\sum_{d=1}^{N}\varrho_d\epsilon_d^{(0)}\Big)+\mathbf{e}_1\frac{\delta}{\sqrt{\beta}},$$

$$\mathbf{B} = \begin{bmatrix} 2 & 2\omega \\ 1 & -\frac{\mu}{2\beta} \end{bmatrix}.$$

The bound (A.3) reveals that the error terms in $\mathbf{x}^{(t)}$ are coupled with each other, whose dynamics are analyzed next.

Part II. Solving the coupled dynamic system: We define $\bar{\mathbf{x}}^{(t)}$ to be the upper bound on $\mathbf{x}^{(t)}$ from (A.3), i.e.,

$$\bar{\mathbf{x}}^{(t+1)} = \Big(\mathbf{I} + \eta_t \beta \mathbf{B}\Big)\bar{\mathbf{x}}^{(t)} + \eta_t \beta \mathbf{z}. \tag{A.4}$$

To solve the coupled dynamic system, we consider the eigen-decomposition of $\mathbf{B}$: $\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{U}^{-1}$, where

$$\mathbf{D} = \begin{bmatrix} \lambda_+ & 0 \\ 0 & \lambda_- \end{bmatrix}, \ \mathbf{U} = \begin{bmatrix} \omega & \omega \\ \frac{\lambda_+}{2} - 1 & \frac{\lambda_-}{2} - 1 \end{bmatrix},$$

$$\mathbf{U}^{-1} = \frac{1}{\omega\sqrt{(1+\frac{\mu}{4\beta})^2 + 2\omega}} \begin{bmatrix} 1 - \frac{\lambda_-}{2} & \omega \\ \frac{\lambda_+}{2} - 1 & -\omega \end{bmatrix},$$

and the eigenvalues are $\lambda_+ = 1 - \frac{\mu}{4\beta} + \sqrt{(1+\frac{\mu}{4\beta})^2 + 2\omega} > 0$ and $\lambda_- = -\frac{\mu/\beta + 2\omega}{\lambda_+} < 0$.

Using this eigen-decomposition, we can obtain the following expression for $\bar{\mathbf{x}}^{(t)}$ through recursive expansion of (A.4):

$$\begin{aligned} \bar{\mathbf{x}}^{(t)} = \mathbf{U} \prod_{\ell=t_{k-1}}^{t-1} (\mathbf{I} + \eta_\ell \beta \mathbf{D})\mathbf{U}^{-1}\mathbf{e}_2 \|\bar{\mathbf{w}}^{(t_{k-1})} - \mathbf{w}^*\| \\ + \mathbf{U}\left[\prod_{\ell=t_{k-1}}^{t-1} (\mathbf{I} + \eta_\ell \beta \mathbf{D}) - \mathbf{I}\right] \mathbf{U}^{-1}\mathbf{B}^{-1}\mathbf{z}. \end{aligned} \tag{A.5}$$

Part III. Connecting $A^{(t)}$ with $\mathbf{x}^{(t)}$, and obtaining the bound: To bound $A^{(t)}$, we perform some algebraic manipulations on (A.1), (A.2) to get:

$$\begin{aligned} &\sqrt{\beta \mathbb{E}[\|\bar{\mathbf{w}}_c^{(t+1)} - \bar{\mathbf{w}}^{(t+1)}\|^2]} \leq (1 + \eta_t \beta)\sqrt{\beta \mathbb{E}[\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2]} \\ &+ \eta_t \beta\Big(\sqrt{\beta}\epsilon_c^{(t)} + \sqrt{\beta}\sum_{d=1}^{N} \varrho_d \epsilon_d^{(t)} + \frac{\delta}{\sqrt{\beta}} + \frac{2\sigma}{\sqrt{\beta}}\Big) + \eta_t \beta y^{(t)}, \end{aligned} \tag{A.6}$$

where $y^{(t)} = \begin{bmatrix}1 & 2\omega\end{bmatrix}\mathbf{x}^{(t)}$. Recursive expansion of (A.6) yields:

$$\begin{aligned}\sqrt{\beta\mathbb{E}[\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2]} &\leq \sum_{\ell=t_{k-1}}^{t-1} \eta_\ell\beta \prod_{\mathrm{j}=\ell+1}^{t-1}(1+\eta_{\mathrm{j}}\beta)y^{(\ell)} \\ + \sum_{\ell=t_{k-1}}^{t-1} \eta_\ell\beta \prod_{\mathrm{j}=\ell+1}^{t-1}(1+\eta_{\mathrm{j}}\beta)\Big(\sqrt{\beta}\epsilon_c^{(0)} &+ \sqrt{\beta}\sum_{d=1}^{N}\varrho_d\epsilon_d^{(0)} + \frac{\delta}{\sqrt{\beta}} + \frac{2\sigma}{\sqrt{\beta}}\Big).\end{aligned} \tag{A.7}$$

We then bound $y^{(\ell)}$ in terms of λ_+ and λ_-. Applying properties of these eigenvalues allows us to further bound (A.6) as:

$$\begin{aligned}\sqrt{\beta\mathbb{E}[\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2]} &\leq 2\omega\Sigma_{+,t}\sqrt{\beta}\|\bar{\mathbf{w}}^{(t_{k-1})} - \mathbf{w}^*\| \\ + \Sigma_{+,t}\Big(\frac{3+\sqrt{3}}{2}\frac{\sigma}{\sqrt{\beta}} + \frac{\delta}{\sqrt{\beta}} &+ \frac{\sqrt{\beta}(1+\sqrt{3})}{2}\epsilon^{(0)} + \sqrt{\beta}\epsilon_c^{(0)}\Big).\end{aligned} \tag{A.8}$$

Taking the square and the weighted sum $\sum_{c=1}^{N}\varrho_c$ of both sides, and using the strong convexity of $F(\cdot)$, we obtain the proposition result.

A.2 Sketch of the Proof of Theorem C.2.1

For the detailed proof, see Appendix C.2 of our online technical report [108]. Here, we provide a summary of the steps taken to carry out the proof.

Consider the global average $\bar{\mathbf{w}}$ of the local models defined in Definition 3.2.1. For $t \in \mathcal{T}_k$, using (3.10), (3.13), and the fact that $\sum\limits_{\mathrm{i}\in\mathcal{S}_c}\mathbf{e}_{\mathrm{i}}^{(t)} = 0\ \forall t$, we have

$$\bar{\mathbf{w}}^{(t+1)} = \bar{\mathbf{w}}^{(t)} - \eta_t\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) - \eta_t\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\mathbf{n}_{\mathrm{j}}^{(t)} \tag{A.9}$$

given Assumption 2. Combining the result of (A.9) with β-smoothness and applying Assumption 3, we have:

$$\begin{aligned}\mathbb{E}_t\left[F(\bar{\mathbf{w}}^{(t+1)})-F(\mathbf{w}^*)\right] &\leq F(\bar{\mathbf{w}}^{(t)})-F(\mathbf{w}^*)\\ &-\eta_t\nabla F(\bar{\mathbf{w}}^{(t)})^\top\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})\\ &+\frac{\eta_t^2\beta}{2}\left\|\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})\right\|^2+\frac{\eta_t^2\beta\sigma^2}{2},\end{aligned}\tag{A.10}$$

where $\mathbb{E}_t$ denotes the conditional expectation, conditioned on $\bar{\mathbf{w}}^{(t)}$. Applying the law of total expectation, Assumption 2, and Lemma 3 (see Appendix C.5), since $\eta_t \leq 1/\beta$ we have

$$\begin{aligned}\mathbb{E}\left[F(\bar{\mathbf{w}}^{(t+1)})-F(\mathbf{w}^*)\right] &\leq (1-\mu\eta_t)\mathbb{E}[F(\bar{\mathbf{w}}^{(t)})-F(\mathbf{w}^*)]\\ &+\frac{\eta_t\beta^2}{2}A^{(t)}+\frac{1}{2}(\eta_t\beta^2(\epsilon^{(t)})^2+\eta_t^2\beta\sigma^2).\end{aligned}\tag{A.11}$$

Finally, noting again the β-smoothness and strong convexity of $F(\cdot)$, we establish inequality relationships between $\mathbb{E}[F(\bar{\mathbf{w}}^{(t)})-F(\mathbf{w}^*)]$ and $\mathbb{E}[F(\hat{\mathbf{w}}^{(t)})-F(\mathbf{w}^*)]$, which concludes the proof.

A.3 Sketch of the Proof of Theorem 3.2.2

The complete proof is provided in Appendix C.3 of our technical report [108]. Here, we summarize the key steps.

The proof is carried out by induction. We aim to prove

$$\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t)})-F(\mathbf{w}^*)\right] \leq \frac{\nu}{t+\alpha},\ \forall t,\tag{A.12}$$

holds, considering the effects of all global aggregations and the local model training interval between consecutive global aggregations. To do so, we need to do two inductions: (i) outer induction: induction across all the global aggregation indices, i.e., k, to demonstrate that (A.12) holds across all global aggregations, and (ii) inner induction: induction across the local model training interval $t \in \mathcal{T}_k$, $\forall k$. We start with the outer induction and consider t_0 as the basis of induction. We see that the condition in (A.12) trivially holds when

$t = t_0 = 0$, since $\nu \geq \alpha[F(\hat{\mathbf{w}}^{(0)}) - F(\mathbf{w}^*)]$ by its definition. We then focus on the outer induction hypothesis, and presume that

$$\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t_{k-1})}) - F(\mathbf{w}^*)\right] \leq \frac{\nu}{t_{k-1} + \alpha} \tag{A.13}$$

holds for $t = t_{k-1}$ for some $k \geq 1$. Finally, to complete the proof for the outer induction, we aim to prove that the induction holds for $t = t_k$, i.e., $\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t_k)}) - F(\mathbf{w}^*)\right] \leq \frac{\nu}{t_k+\alpha}$. We prove this step by proving the condition

$$\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)\right] \leq \frac{\nu}{t + \alpha}, \ \forall t \in \mathcal{T}_k, \tag{A.14}$$

which implies the outer induction step (when $t = t_k$), and is proved via the inner induction over $t \in \mathcal{T}_k$. To complete the proof using the inner induction, consider $t = t_{k-1}$ as the basis of induction. We note that the condition in (A.12) holds as a result of the induction hypothesis from the outer induction. Now, suppose that

$$\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)\right] \leq \frac{\nu}{t + \alpha}, \tag{A.15}$$

holds for $t \in \{t_{k-1}, \ldots, t_k - 1\}$. We aim to demonstrate that it holds at $t+1$ as follows. From the result of Theorem C.2.1, using the induction hypothesis, the bound on $A^{(t)}$, $\epsilon^{(t)} = \eta_t \phi$, and the facts that $\eta_{t+1} \leq \eta_t$, $\eta_t \leq \eta_0 \leq \frac{\mu}{\beta^2} \leq 1/\beta$ and $\epsilon^{(0)} = \eta_0 \phi \leq \phi/\beta$, we get

$$\begin{aligned}\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)\right] &\leq (1 - \mu\eta_t)\,\frac{\nu}{t + \alpha} + \frac{\eta_t^2 \beta}{2}\left(\sigma^2 + 2\phi^2\right) \\ &+ \frac{8\eta_t \omega^2 \beta^2}{\mu} \underbrace{(\Sigma_{+,t})^2}_{(a)} \frac{\nu}{t_{k-1} + \alpha} \quad + \frac{25}{2}\eta_t \underbrace{(\Sigma_{+,t})^2}_{(b)} \left(\sigma^2 + \phi^2 + \delta^2\right),\end{aligned} \tag{A.16}$$

where $\Sigma_{+,t}$ is given in Proposition 3.2.1. To get a tight upper bound for (A.16), we bound the two instances of $(\Sigma_{+,t})^2$ appearing in (a) and (b) differently. For (a), we first use the fact that $\lambda_+ = 1 - \frac{\mu}{4\beta} + \sqrt{(1+\frac{\mu}{4\beta})^2 + 2\omega} \in [2, 1+\sqrt{3}]$, which implies

$$\Sigma_{+,t} \leq \gamma\beta \underbrace{\left(\prod_{\mathrm{j}=t_{k-1}}^{t-1}\left(1+\frac{\gamma\beta\lambda_+}{\mathrm{j}+\alpha}\right)\right)}_{\text{(i)}} \underbrace{\sum_{\ell=t_{k-1}}^{t-1}\frac{1}{\ell+\alpha+\gamma\beta\lambda_+}}_{\text{(ii)}}. \tag{A.17}$$

To bound (ii), since $\frac{1}{\ell+\alpha+\gamma\beta\lambda_+}$ is decreasing in ℓ, we have

$$\begin{aligned}\sum_{\ell=t_{k-1}}^{t-1}\frac{1}{\ell+\alpha+\gamma\beta\lambda_+} &\leq \int_{t_{k-1}-1}^{t-1}\frac{1}{\ell+\alpha+\gamma\beta\lambda_+}\mathrm{d}\ell \\ &= \ln\left(1+\frac{t-t_{k-1}}{t_{k-1}-1+\alpha+\gamma\beta\lambda_+}\right).\end{aligned} \tag{A.18}$$

To bound (i), we first rewrite it as $\prod_{\mathrm{j}=t_{k-1}}^{t-1}\left(1+\frac{\gamma\beta\lambda_+}{\mathrm{j}+\alpha}\right) = \mathrm{e}^{\sum_{\mathrm{j}=t_{k-1}}^{t-1}\ln\left(1+\frac{\gamma\beta\lambda_+}{\mathrm{j}+\alpha}\right)}$, and use the fact that $\ln(1+\frac{\gamma\beta\lambda_+}{\mathrm{j}+\alpha})$ is a decreasing function with respect to j, and that $\alpha > 1$, to get

$$\begin{aligned}&\sum_{\mathrm{j}=t_{k-1}}^{t-1}\ln(1+\frac{\gamma\beta\lambda_+}{\mathrm{j}+\alpha}) \leq \int_{t_{k-1}-1}^{t-1}\ln(1+\frac{\gamma\beta\lambda_+}{\mathrm{j}+\alpha})\mathrm{dj} \\ &\leq \gamma\beta\lambda_+\int_{t_{k-1}-1}^{t-1}\frac{1}{\mathrm{j}+\alpha}\mathrm{dj} = \gamma\beta\lambda_+\ln\left(1+\frac{t-t_{k-1}}{t_{k-1}-1+\alpha}\right),\end{aligned}$$

which yields $\prod_{\mathrm{j}=t_{k-1}}^{t-1}\left(1+\frac{\gamma\beta\lambda_+}{\mathrm{j}+\alpha}\right) \leq \left(1+\frac{t-t_{k-1}}{t_{k-1}-1+\alpha}\right)^{\gamma\beta\lambda_+}$.

Using the results obtained for bounding (i) and (ii) back into (A.17), using the fact that $\ln(1+x) \leq 2\sqrt{x}$ for $x \geq 0$, and performing some algebraic manipulations, we get

$$(\Sigma_{+,t})^2 \leq 4(\tau-1)\left(1+\frac{\tau}{\alpha-1}\right)^2\left(1+\frac{\tau-1}{\alpha-1}\right)^{6\gamma\beta}\eta_t^2\beta^2[t_{k-1}+\alpha]. \tag{A.19}$$

On the other hand, we bound (b) in (A.16) as follows:

$$
\begin{aligned}
(t+\alpha)(\Sigma_{+,t})^2 &\leq 4\gamma^2\beta^2 \frac{(t-t_{k-1})(t+\alpha)}{t_{k-1}+\alpha+1}\left(1+\frac{t-t_{k-1}}{t_{k-1}+\alpha-1}\right)^{6\gamma\beta} \\
&\leq 4\gamma^2\beta^2(\tau-1)\left(1+\frac{\tau-2}{\alpha+1}\right)\left(1+\frac{\tau-1}{\alpha-1}\right)^{6\gamma\beta},
\end{aligned}
$$

which implies

$$
(\Sigma_{+,t})^2 \leq 4\gamma\beta(\tau-1)\left(1+\frac{\tau-2}{\alpha+1}\right)\left(1+\frac{\tau-1}{\alpha-1}\right)^{6\gamma\beta}\eta_t\beta. \tag{A.20}
$$

Substituting (A.19) and (A.20) into (A.16), we get

$$
\begin{aligned}
&\mathbb{E}[F(\hat{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)] \leq \\
&\qquad \left(1-\mu\eta_t + Z_1\omega^2\eta_t^2\beta^2\right)\frac{\nu}{t+\alpha} + \eta_t^2\beta^2 Z_2,
\end{aligned} \tag{A.21}
$$

where Z_1 and Z_2 are given in the statement of the theorem.

To complete the induction, we need to show that the right hand side of (A.21) is less than or equal to $\frac{\nu}{t+1+\alpha}$. This condition can be represented equivalently as the following inequality:

$$
\begin{aligned}
&\gamma^2\beta^2\left(-\frac{\mu}{\gamma\beta^2}(t+\alpha) + Z_1\omega^2\right)\nu + Z_2\gamma^2\beta^2(t+\alpha) \\
&\qquad + \nu(t+\alpha-1) + \frac{\nu}{t+1+\alpha} \leq 0.
\end{aligned} \tag{A.22}
$$

(A.22) needs to be satisfied $\forall t \geq 0$. Since the expression on the left hand side is convex in t, it is sufficient to satisfy this condition for $t \to \infty$ and $t = 0$. Obtaining these limits gives us the following set of conditions: $\mu\gamma - 1 > 0$, $\omega < \omega_{\max}$ and ν as in (3.25), which completes the induction, hence the proof.

B. PRELIMINARIES AND NOTATIONS USED IN THE PROOFS

In the following Appendices, in order to increase the tractability of the the expressions inside the proofs, we introduce the the following scaled parameters: (i) strong convexity denoted by $\tilde{\mu}$, normalized gradient diversity by $\tilde{\delta}$, step size by $\tilde{\eta}_t$, SGD variance $\tilde{\sigma}$, and consensus error inside the clusters $\tilde{\epsilon}_c^{(t)}$ and across the network $\tilde{\epsilon}^{(t)}$ inside the cluster as follows:

- **Strong convexity**: F is μ-strongly convex, i.e.,

$$F(\mathbf{w}_1) \geq F(\mathbf{w}_2) + \nabla F(\mathbf{w}_2)^\top (\mathbf{w}_1 - \mathbf{w}_2) + \frac{\tilde{\mu}\beta}{2}\Big\|\mathbf{w}_1 - \mathbf{w}_2\Big\|^2, \ \forall \mathbf{w}_1, \mathbf{w}_2, \tag{B.1}$$

  where as compared to Assumption 1, we considered $\tilde{\mu} = \mu/\beta \in (0,1)$.

- **Gradient diversity**: The gradient diversity across the device clusters c is measured via two non-negative constants δ, ζ that satisfy

$$\|\nabla \hat{F}_c(\mathbf{w}) - \nabla F(\mathbf{w})\| \leq \sqrt{\beta}\tilde{\delta} + 2\omega\beta\|\mathbf{w} - \mathbf{w}^*\|, \ \forall c, \mathbf{w}, \tag{B.2}$$

  where as compared to Assumption 3.1.1, we presumed $\tilde{\delta} = \delta/\sqrt{\beta}$ and $\omega = \zeta/(2\beta) \in [0,1]$.

- **Step size**: The local updates to compute *intermediate updated local model* at the devices is expressed as follows:

$$\widetilde{\mathbf{w}}_{\rm i}^{(t)} = \mathbf{w}_{\rm i}^{(t-1)} - \frac{\tilde{\eta}_{t-1}}{\beta}\widehat{\mathbf{g}}_{\rm i}^{(t-1)}, \ t \in \mathcal{T}_k, \tag{B.3}$$

  where we used the scaled in the step size, i.e., $\tilde{\eta}_{t-1} = \eta_{t-1}\beta$. Also, when we consider decreasing step size, we consider scaled parameter $\tilde{\gamma}$ in the step size as follows: $\frac{\gamma}{t+\alpha} = \frac{\tilde{\gamma}/\beta}{t+\alpha}$ indicating that $\tilde{\gamma} = \gamma\beta$.

- **Variance of the noise of the estimated gradient through SGD**: The variance on the SGD noise is bounded as:

$$\mathbb{E}[\|\mathbf{n}_{\mathrm{j}}^{(t)}\|^2] \leq \beta\tilde{\sigma}^2, \forall \mathrm{j}, t, \tag{B.4}$$

  where we consider scaled SGD noise as: $\tilde{\sigma}^2 = \sigma^2/\beta$.

- **Average of the consensus error inside cluster c and across the network**: $\epsilon_c^{(t)}$ is an upper bound on the average of the consensus error inside cluster c for time t, i.e.,

$$\frac{1}{s_c}\sum_{\mathrm{i}\in\mathcal{S}_c}\|\mathbf{e}_{\mathrm{i}}^{(t)}\|^2 \leq (\tilde{\epsilon}_c^{(t)})^2/\beta, \tag{B.5}$$

  where we use the scaled consensus error $(\tilde{\epsilon}_c^{(t)})^2 = \beta(\epsilon_c^{(t)})^2$. Also, in the proofs we use the notation ϵ to denote the average consensus error across the network defined as $(\epsilon^{(t)})^2 = \sum_{c=1}^{N}\varrho_c(\epsilon_c^{(t)})^2$. When the consensus is assumed to be decreasing over time we use the scaled coefficient $\tilde{\phi}^2 = \phi^2/\beta$, resulting in $(\epsilon^{(t)})^2 = \eta_t^2\tilde{\phi}^2\beta$.

Finally, to track the global model variations, we introduce the instantaneous global model $\hat{\mathbf{w}}^{(t)} = \sum\limits_{c=1}^{N}\varrho_c\mathbf{w}_{n_c}^{(t)}$, where n_c is a node uniformly sampled from cluster c. We note that $\hat{\mathbf{w}}^{(t)}$ is only realized at the server at the instance of the global aggregations.

C. PROOFS

C.1 Proof of Proposition C.1.1

Proposition C.1.1. *Under Assumptions 1 and 3, if $\eta_t = \frac{\gamma}{t+\alpha}$, $\epsilon^{(t)}$ is non-increasing with respect to $t \in \mathcal{T}_k$, i.e., $\epsilon^{(t+1)}/\epsilon^{(t)} \leq 1$ and $\alpha \geq \max\{\beta\gamma[\frac{\mu}{4\beta} - 1 + \sqrt{(1+\frac{\mu}{4\beta})^2 + 2\omega}], \frac{\beta^2\gamma}{\mu}\}$, using* `TT-HF` *for ML model training, the following upper bound on the expected model dispersion across the clusters holds:*

$$A^{(t)} \leq \frac{16\omega^2}{\mu}[\Sigma_{+,t}]^2[F(\bar{\mathbf{w}}(t_{k-1})) - F(\mathbf{w}^*)] + 25[\Sigma_{+,t}]^2\left[\frac{\sigma^2}{\beta^2} + \frac{\delta^2}{\beta^2} + (\epsilon^{(0)})^2\right], \quad t \in \mathcal{T}_k, \tag{C.1}$$

where

$$[\Sigma_{+,t}]^2 = \left[\sum_{\ell=t_{k-1}}^{t-1}\left(\prod_{\mathrm{j}=t_{k-1}}^{\ell-1}(1+\eta_{\mathrm{j}}\beta\lambda_+)\right)\beta\eta_\ell\left(\prod_{\mathrm{j}=\ell+1}^{t-1}(1+\eta_{\mathrm{j}}\beta)\right)\right]^2, \tag{C.2}$$

and

$$\lambda_+ = 1 - \frac{\mu}{4\beta} + \sqrt{(1+\frac{\mu}{4\beta})^2 + 2\omega}. \tag{C.3}$$

Proof. We break down the proof into 3 parts: in Part I we find the relationship between $\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\|$ and $\sum_{c=1}^{N}\varrho_c\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|$, which turns out to form a coupled dynamic system, which is solved in Part II. Finally, Part III draws the connection between $A^{(t)}$ and the solution of the coupled dynamic system and obtains the upper bound on $A^{(t)}$.

(Part I) Finding the relationship between $\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\|$ and $\sum_{c=1}^{N}\varrho_c\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|$: Using the definition of $\bar{\mathbf{w}}^{(t+1)}$ given in Definition 3.2.1, and the notations introduced in Appendix B, we have:

$$\bar{\mathbf{w}}^{(t+1)} = \bar{\mathbf{w}}^{(t)} - \frac{\tilde{\eta}_t}{\beta}\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\widehat{\mathbf{g}}_{\mathrm{j},t}, \; t \in \mathcal{T}_k. \tag{C.4}$$

Adding and subtracting terms in the above equality gives us:

$$\begin{aligned}\bar{\mathbf{w}}^{(t+1)} - \mathbf{w}^* = \bar{\mathbf{w}}^{(t)} - \mathbf{w}^* - \frac{\tilde{\eta}_t}{\beta}\nabla F(\bar{\mathbf{w}}^{(t)})\\ - \frac{\tilde{\eta}_t}{\beta}\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}[\widehat{\mathbf{g}}_{\mathrm{j},t} - \nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})]\\ - \frac{\tilde{\eta}_t}{\beta}\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}[\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) - \nabla F_{\mathrm{j}}(\bar{\mathbf{w}}_c^{(t)})]\\ - \frac{\tilde{\eta}_t}{\beta}\sum_{c=1}^{N}\varrho_c[\nabla F_c(\bar{\mathbf{w}}_c^{(t)}) - \nabla F_c(\bar{\mathbf{w}}^{(t)})].\end{aligned} \tag{C.5}$$

Taking the norm-2 from the both hand sides of the above equality and applying the triangle inequality yields:

$$\begin{aligned}\|\bar{\mathbf{w}}^{(t+1)} - \mathbf{w}^*\| \leq& \|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^* - \frac{\tilde{\eta}_t}{\beta}\nabla F(\bar{\mathbf{w}}^{(t)})\| + \frac{\tilde{\eta}_t}{\beta}\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\|\mathbf{n}_{\mathrm{j}}^{(t)}\|\\ &+ \frac{\tilde{\eta}_t}{\beta}\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\|\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) - \nabla F_{\mathrm{j}}(\bar{\mathbf{w}}_c^{(t)})\|\\ &+ \frac{\tilde{\eta}_t}{\beta}\sum_{c=1}^{N}\varrho_c\|\nabla F_c(\bar{\mathbf{w}}_c^{(t)}) - \nabla F_c(\bar{\mathbf{w}}^{(t)})\|.\end{aligned} \tag{C.6}$$

To bound the terms on the right hand side above, we first use the μ-strong convexity and β-smoothness of $F(\cdot)$, when $\eta_t \leq \frac{\mu}{\beta^2}$, to get

$$\begin{aligned}&\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^* - \frac{\tilde{\eta}_t}{\beta}\nabla F(\bar{\mathbf{w}}^{(t)})\|\\ &= \sqrt{\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\|^2 + (\frac{\tilde{\eta}_t}{\beta})^2\|\nabla F(\bar{\mathbf{w}}^{(t)})\|^2 - \frac{2\tilde{\eta}_t}{\beta}(\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*)^\top\nabla F(\bar{\mathbf{w}}^{(t)})}\\ &\overset{(a)}{\leq} \sqrt{(1 - 2\tilde{\eta}_t\tilde{\mu})\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\|^2 + (\frac{\tilde{\eta}_t}{\beta})^2\|\nabla F(\bar{\mathbf{w}}^{(t)})\|^2}\\ &\overset{(b)}{\leq} \sqrt{1 - 2\tilde{\eta}_t\tilde{\mu} + \tilde{\eta}_t^2}\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\| \overset{(c)}{\leq} (1 - \frac{\tilde{\eta}_t\tilde{\mu}}{2})\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\|,\end{aligned} \tag{C.7}$$

where (a) results from the property of a strongly convex function, i.e., $(\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*)^\top\nabla F(\bar{\mathbf{w}}^{(t)}) \geq \tilde{\mu}\beta\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\|^2$, (b) comes from the property of smooth functions, i.e., $\|\nabla F(\bar{\mathbf{w}}^{(t)})\|^2 \leq \beta^2\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\|^2$ and the last step (c) follows from the fact that $\tilde{\eta}_t \leq \tilde{\eta}_0$ and assuming $\tilde{\eta}_0 \leq \tilde{\mu}$, implying $\alpha \geq \tilde{\gamma}/\tilde{\mu}$. Also, considering the other terms on the right hand side of (C.6), using β-smoothness, we have

$$\|\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) - \nabla F_{\mathrm{j}}(\bar{\mathbf{w}}_c^{(t)})\| \leq \beta \frac{1}{s_c} \sum_{\mathrm{j} \in \mathcal{S}_c} \|\mathbf{w}_{\mathrm{j}}^{(t)} - \bar{\mathbf{w}}_c^{(t)}\|. \tag{C.8}$$

Moreover, using Condition 1, we get

$$\begin{aligned}\frac{1}{s_c} \sum_{\mathrm{j} \in \mathcal{S}_c} \|\mathbf{w}_{\mathrm{j}}^{(t)} - \bar{\mathbf{w}}_c^{(t)}\| &= \frac{1}{s_c} \sum_{\mathrm{j} \in \mathcal{S}_c} \|\mathbf{e}_{\mathrm{j}}^{(t)}\| \\ &\leq \sqrt{\frac{1}{s_c} \sum_{\mathrm{j} \in \mathcal{S}_c} \|\mathbf{e}_{\mathrm{j}}^{(t)}\|^2} \leq \tilde{\epsilon}_c^{(t)}/\sqrt{\beta}. \end{aligned} \tag{C.9}$$

Combining (C.8) and (C.9) gives us:

$$\frac{1}{s_c} \sum_{\mathrm{j} \in \mathcal{S}_c} \|\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) - \nabla F_{\mathrm{j}}(\bar{\mathbf{w}}_c^{(t)})\| \leq \sqrt{\beta} \tilde{\epsilon}_c^{(t)}. \tag{C.10}$$

Replacing the result of (C.7) and (C.10) in (C.6) yields:

$$\begin{aligned}\|\bar{\mathbf{w}}^{(t+1)} - \mathbf{w}^*\| \leq & (1 - \frac{\tilde{\eta}_t \tilde{\mu}}{2}) \|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\| + \frac{\tilde{\eta}_t}{\beta} \sum_{c=1}^{N} \varrho_c \frac{1}{s_c} \sum_{\mathrm{j} \in \mathcal{S}_c} \|\mathbf{n}_{\mathrm{j}}^{(t)}\| \\ & + \frac{\tilde{\eta}_t}{\sqrt{\beta}} \sum_{c=1}^{N} \varrho_c \tilde{\epsilon}_c^{(t)} + \tilde{\eta}_t \sum_{c=1}^{N} \varrho_c \|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|. \end{aligned} \tag{C.11}$$

Multiplying the both hand sides of the above inequality by $\sqrt{\beta}$ followed by taking square and expectation, we get

$$\begin{aligned}\mathbb{E}\left[\sqrt{\beta} \|\bar{\mathbf{w}}^{(t+1)} - \mathbf{w}^*\|\right]^2 \leq & \mathbb{E}\Bigg[\sqrt{\beta}(1 - \frac{\tilde{\eta}_t \tilde{\mu}}{2}) \|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\| + \frac{\tilde{\eta}_t}{\sqrt{\beta}} \sum_{c=1}^{N} \varrho_c \frac{1}{s_c} \sum_{\mathrm{j} \in \mathcal{S}_c} \|\mathbf{n}_{\mathrm{j}}^{(t)}\| \\ & + \tilde{\eta}_t \sum_{c=1}^{N} \varrho_c \tilde{\epsilon}_c^{(t)} + \tilde{\eta}_t \sqrt{\beta} \sum_{c=1}^{N} \varrho_c \|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\| \Bigg]^2. \end{aligned} \tag{C.12}$$

Taking the square roots from the both hand sides and using Fact 1 (See Appendix C.6) yields:

$$\begin{aligned}\sqrt{\beta\mathbb{E}[\|\bar{\mathbf{w}}^{(t+1)}-\mathbf{w}^*\|^2]} \leq& (1-\frac{\tilde{\eta}_t\tilde{\mu}}{2})\sqrt{\beta\mathbb{E}[\|\bar{\mathbf{w}}^{(t)}-\mathbf{w}^*\|^2]}+\tilde{\eta}_t\tilde{\sigma}\\ &+\tilde{\eta}_t\sum_{c=1}^{N}\varrho_c\tilde{\epsilon}_c^{(t)}+\tilde{\eta}_t\sqrt{\beta\Big(\sum_{c=1}^{N}\varrho_c\mathbb{E}\|\bar{\mathbf{w}}_c^{(t)}-\bar{\mathbf{w}}^{(t)}\|\Big)^2}.\end{aligned}\tag{C.13}$$

We compact (C.13) and represent it via the following relationship:

$$x_2^{(t+1)} \leq \left[\tilde{\eta}_t,(1-\frac{\tilde{\eta}_t\tilde{\mu}}{2})\right]\mathbf{x}^{(t)}+\tilde{\eta}_t\left(\tilde{\sigma}+\sum_{c=1}^{N}\varrho_c\tilde{\epsilon}_c^{(t)}\right),\tag{C.14}$$

where $\mathbf{x}^{(t)}=\left[x_1^{(t)},x_2^{(t)}\right]^\top$, $x_1^{(t)}=\sqrt{\beta\mathbb{E}[(\sum_{c=1}^{N}\varrho_c\|\bar{\mathbf{w}}_c^{(t)}-\bar{\mathbf{w}}^{(t)}\|)^2]}$, and $x_2^{(t)}=\sqrt{\beta\mathbb{E}[\|\bar{\mathbf{w}}^{(t)}-\mathbf{w}^*\|^2]}$.

The relationship in (C.14) reveals the dependency of $x_2^{(t+1)}$ on $x_2^{(t)}$ and $x_1^{(t)}$. To bound $x_1^{(t)}$, we first use the fact that $\bar{\mathbf{w}}_c^{(t+1)}$ can be written as follows:

$$\bar{\mathbf{w}}_c^{(t+1)}=\bar{\mathbf{w}}_c^{(t)}-\frac{\tilde{\eta}_t}{\beta}\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})-\frac{\tilde{\eta}_t}{\beta}\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\mathbf{n}_{\mathrm{j}}^{(t)}.\tag{C.15}$$

Similarly, $\bar{\mathbf{w}}^{(t+1)}$ can be written as:

$$\bar{\mathbf{w}}^{(t+1)}=\bar{\mathbf{w}}^{(t)}-\frac{\tilde{\eta}_t}{\beta}\sum_{d=1}^{N}\varrho_d\frac{1}{s_d}\sum_{\mathrm{j}\in\mathcal{S}_d}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})-\frac{\tilde{\eta}_t}{\beta}\sum_{d=1}^{N}\varrho_d\frac{1}{s_d}\sum_{\mathrm{j}\in\mathcal{S}_d}\mathbf{n}_{\mathrm{j}}^{(t)}.\tag{C.16}$$

Combining (C.15) and (C.16) and performing some algebraic manipulations yields:

$$\begin{aligned}\bar{\mathbf{w}}_c^{(t+1)}-\bar{\mathbf{w}}^{(t+1)}=&\bar{\mathbf{w}}_c^{(t)}-\bar{\mathbf{w}}^{(t)}-\frac{\tilde{\eta}_t}{\beta}\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\mathbf{n}_{\mathrm{j}}^{(t)}+\frac{\tilde{\eta}_t}{\beta}\sum_{d=1}^{N}\varrho_d\frac{1}{s_d}\sum_{\mathrm{j}\in\mathcal{S}_d}\mathbf{n}_{\mathrm{j}}^{(t)}\\ &-\frac{\tilde{\eta}_t}{\beta}\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\left[\nabla F_{\mathrm{j}}(\bar{\mathbf{w}}_{\mathrm{j}}^{(t)})-\nabla F_{\mathrm{j}}(\bar{\mathbf{w}}_c^{(t)})\right]+\frac{\tilde{\eta}_t}{\beta}\sum_{d=1}^{N}\varrho_d\frac{1}{s_d}\sum_{\mathrm{j}\in\mathcal{S}_d}\left[\nabla F_{\mathrm{j}}(\bar{\mathbf{w}}_{\mathrm{j}}^{(t)})-\nabla F_{\mathrm{j}}(\bar{\mathbf{w}}_d^{(t)})\right]\\ &-\frac{\tilde{\eta}_t}{\beta}\Big[\nabla\hat{F}_c(\bar{\mathbf{w}}_c^{(t)})-\nabla\hat{F}_c(\bar{\mathbf{w}}^{(t)})\Big]+\frac{\tilde{\eta}_t}{\beta}\sum_{d=1}^{N}\varrho_d\Big[\nabla\hat{F}_d(\bar{\mathbf{w}}_d^{(t)})-\nabla\hat{F}_d(\bar{\mathbf{w}}^{(t)})\Big]\\ &-\frac{\tilde{\eta}_t}{\beta}\Big[\nabla\hat{F}_c(\bar{\mathbf{w}}^{(t)})-\nabla F(\bar{\mathbf{w}}^{(t)})\Big].\end{aligned}\tag{C.17}$$

Taking the norm-2 of the both hand sides of the above equality and applying the triangle inequality gives us

$$\begin{aligned}
&\|\bar{\mathbf{w}}_c^{(t+1)} - \bar{\mathbf{w}}^{(t+1)}\| \leq \|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\| + \frac{\tilde{\eta}_t}{\beta}\|\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\mathbf{n}_{\mathrm{j}}^{(t)}\| + \frac{\tilde{\eta}_t}{\beta}\|\sum_{d=1}^{N}\varrho_d\frac{1}{s_d}\sum_{\mathrm{j}\in\mathcal{S}_d}\mathbf{n}_{\mathrm{j}}^{(t)}\| \\
&+ \frac{\tilde{\eta}_t}{\beta}\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\|\nabla F_{\mathrm{j}}(\bar{\mathbf{w}}_{\mathrm{j}}^{(t)}) - \nabla F_{\mathrm{j}}(\bar{\mathbf{w}}_c^{(t)})\| + \frac{\tilde{\eta}_t}{\beta}\sum_{d=1}^{N}\varrho_d\frac{1}{s_d}\sum_{\mathrm{j}\in\mathcal{S}_d}\|\nabla F_{\mathrm{j}}(\bar{\mathbf{w}}_{\mathrm{j}}^{(t)}) - \nabla F_{\mathrm{j}}(\bar{\mathbf{w}}_d^{(t)})\| \\
&+ \frac{\tilde{\eta}_t}{\beta}\|\nabla \hat{F}_c(\bar{\mathbf{w}}_c^{(t)}) - \nabla \hat{F}_c(\bar{\mathbf{w}}^{(t)})\| + \frac{\tilde{\eta}_t}{\beta}\sum_{d=1}^{N}\varrho_d\|\nabla \hat{F}_d(\bar{\mathbf{w}}_d^{(t)}) - \nabla \hat{F}_d(\bar{\mathbf{w}}^{(t)})\| \\
&+ \frac{\tilde{\eta}_t}{\beta}\|\nabla \hat{F}_c(\bar{\mathbf{w}}^{(t)}) - \nabla F(\bar{\mathbf{w}}^{(t)})\|. \qquad \text{(C.18)}
\end{aligned}$$

Using β-smoothness of $F_{\mathrm{j}}(\cdot)$, $\forall$j, and $\hat{F}_c(\cdot)$, $\forall c$, Definition 3.1.1 and Condition 1, we further bound the right hand side of (C.18) to get

$$\begin{aligned}
&\|\bar{\mathbf{w}}_c^{(t+1)} - \bar{\mathbf{w}}^{(t+1)}\| \leq (1+\tilde{\eta}_t)\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\| + 2\omega\tilde{\eta}_t\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\| + \tilde{\eta}_t\sum_{d=1}^{N}\varrho_d\|\bar{\mathbf{w}}_d^{(t)} - \bar{\mathbf{w}}^{(t)}\| \\
&+ \frac{\tilde{\eta}_t}{\beta}\|\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\mathbf{n}_{\mathrm{j}}^{(t)}\| + \frac{\tilde{\eta}_t}{\beta}\|\sum_{d=1}^{N}\varrho_d\frac{1}{s_d}\sum_{\mathrm{j}\in\mathcal{S}_d}\mathbf{n}_{\mathrm{j}}^{(t)}\| \\
&+ \tilde{\eta}_t\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\|\bar{\mathbf{w}}_{\mathrm{j}}^{(t)} - \bar{\mathbf{w}}_c^{(t)}\| + \tilde{\eta}_t\sum_{d=1}^{N}\varrho_d\frac{1}{s_d}\sum_{\mathrm{j}\in\mathcal{S}_d}\|\bar{\mathbf{w}}_{\mathrm{j}}^{(t)} - \bar{\mathbf{w}}_d^{(t)}\| + \frac{\tilde{\eta}_t}{\sqrt{\beta}}\tilde{\delta}. \qquad \text{(C.19)}
\end{aligned}$$

Using (C.9) we have $\frac{1}{s_d}\sum_{\mathrm{j}\in\mathcal{S}_d}\|\bar{\mathbf{w}}_{\mathrm{j}}^{(t)} - \bar{\mathbf{w}}_d^{(t)}\| \leq \frac{\tilde{\epsilon}_d^{(t)}}{\sqrt{\beta}}$, and thus (C.19) can be written as

$$\begin{aligned}
&\|\bar{\mathbf{w}}_c^{(t+1)} - \bar{\mathbf{w}}^{(t+1)}\| \leq (1+\tilde{\eta}_t)\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\| + 2\omega\tilde{\eta}_t\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\| + \tilde{\eta}_t\sum_{d=1}^{N}\varrho_d\|\bar{\mathbf{w}}_d^{(t)} - \bar{\mathbf{w}}^{(t)}\| \\
&+ \frac{\tilde{\eta}_t}{\beta}\|\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\mathbf{n}_{\mathrm{j}}^{(t)}\| + \frac{\tilde{\eta}_t}{\beta}\|\sum_{d=1}^{N}\varrho_d\frac{1}{s_d}\sum_{\mathrm{j}\in\mathcal{S}_d}\mathbf{n}_{\mathrm{j}}^{(t)}\| + \frac{\tilde{\eta}_t}{\sqrt{\beta}}\left(\tilde{\epsilon}_c^{(t)} + \sum_{d=1}^{N}\varrho_d\tilde{\epsilon}_d^{(t)} + \tilde{\delta}\right). \qquad \text{(C.20)}
\end{aligned}$$

Taking the weighted sum $\sum_{c=1}^{N} \varrho_c$ from the both hand sides of the above inequality gives us

$$\sum_{c=1}^{N} \varrho_c \|\bar{\mathbf{w}}_c^{(t+1)} - \bar{\mathbf{w}}^{(t+1)}\| \leq (1+2\tilde{\eta}_t) \sum_{c=1}^{N} \varrho_c \|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\| + 2\omega\tilde{\eta}_t \|\bar{\mathbf{w}}^{(t)} - \mathbf{w}^*\| \\ + \frac{\tilde{\eta}_t}{\beta} \sum_{c=1}^{N} \varrho_c \|\frac{1}{s_c} \sum_{\mathrm{j}\in\mathcal{S}_c} \mathbf{n}_{\mathrm{j}}^{(t)}\| + \frac{\tilde{\eta}_t}{\beta} \|\sum_{d=1}^{N} \varrho_d \frac{1}{s_d} \sum_{\mathrm{j}\in\mathcal{S}_d} \mathbf{n}_{\mathrm{j}}^{(t)}\| + \frac{\tilde{\eta}_t}{\sqrt{\beta}} \left(2\sum_{d=1}^{N} \varrho_d \tilde{\epsilon}_d^{(t)} + \tilde{\delta} \right). \tag{C.21}$$

Multiplying the both hand side of the above inequality by $\sqrt{\beta}$, followed by taking square and expectation, using a similar procedure used to obtain (C.13), we get the bound on $x_1^{(t+1)}$ as follows:

$$x_1^{(t+1)} \leq [(1+2\tilde{\eta}_t), 2\omega\tilde{\eta}_t]\mathbf{x}^{(t)} + \tilde{\eta}_t \left(2\sum_{d=1}^{N} \varrho_d \tilde{\epsilon}_d^{(t)} + \tilde{\delta} + 2\tilde{\sigma} \right). \tag{C.22}$$

(Part II) Solving the coupled dynamic system: To bound $\mathbf{x}^{(t)}$, we need to bound $x_1^{(t)}$ and $x_2^{(t)}$, where $x_2^{(t)}$ is given by (C.14), which is dependent on $\mathbf{x}^{(t-1)}$. Also, $x_1^{(t)}$ is given in (C.22) which is dependent on $\mathbf{x}^{(t-1)}$. This leads to a *coupled dynamic system* where $\mathbf{x}^{(t)}$ can be expressed in a compact form as follows:

$$\mathbf{x}^{(t+1)} \leq [\mathbf{I} + \tilde{\eta}_t \mathbf{B}]\mathbf{x}^{(t)} + \tilde{\eta}_t \mathbf{z}, \tag{C.23}$$

where $\mathbf{x}^{(t_{k-1})} = \mathbf{e}_2\sqrt{\beta}\|\bar{\mathbf{w}}^{(t_{k-1})} - \mathbf{w}^*\|$, $\mathbf{z} = [2,1]^\top[\tilde{\sigma} + \sum_{d=1}^{N} \varrho_d \tilde{\epsilon}_d^{(0)}] + \mathbf{e}_1\tilde{\delta}$, $\mathbf{B} = \begin{bmatrix} 2 & 2\omega \\ 1 & -\tilde{\mu}/2 \end{bmatrix}$, $\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. We aim to characterize an upper bound on $\mathbf{x}^{(t)}$ denoted by $\bar{\mathbf{x}}^{(t)}$, where

$$\bar{\mathbf{x}}^{(t+1)} = [\mathbf{I} + \tilde{\eta}_t \mathbf{B}]\bar{\mathbf{x}}^{(t)} + \tilde{\eta}_t \mathbf{z}. \tag{C.24}$$

To solve the coupled dynamic, we use the eigen-decomposition on $\mathbf{B}$: $\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{U}^{-1}$, where

$$\mathbf{D} = \begin{bmatrix} \lambda_+ & 0 \\ 0 & \lambda_- \end{bmatrix}, \ \mathbf{U} = \begin{bmatrix} \omega & \omega \\ \frac{\lambda_+}{2} - 1 & \frac{\lambda_-}{2} - 1 \end{bmatrix}, \ \mathbf{U}^{-1} = \frac{1}{\omega\sqrt{(1+\tilde{\mu}/4)^2 + 2\omega}} \begin{bmatrix} 1 - \frac{\lambda_-}{2} & \omega \\ \frac{\lambda_+}{2} - 1 & -\omega \end{bmatrix}$$

and the eigenvalues in $\mathbf{D}$ are given by

$$\lambda_+ = 1 - \tilde{\mu}/4 + \sqrt{(1+\tilde{\mu}/4)^2 + 2\omega} > 0 \tag{C.25}$$

and

$$\lambda_- = 1 - \tilde{\mu}/4 - \sqrt{(1+\tilde{\mu}/4)^2 + 2\omega} = -\frac{\tilde{\mu}+2\omega}{\lambda_+} < 0 \tag{C.26}$$

To further compact the relationship in (C.24), we introduce a variable $\mathbf{f}^{(t)}$, where $\mathbf{f}^{(t)} = \mathbf{U}^{-1}\bar{\mathbf{x}}^{(t)} + \mathbf{U}^{-1}\mathbf{B}^{-1}\mathbf{z}$, satisfying the following recursive expression:

$$\mathbf{f}^{(t+1)} = [\mathbf{I} + \tilde{\eta}_t\mathbf{D}]\mathbf{f}^{(t)}. \tag{C.27}$$

Recursive expansion of the right hand side of the above equality yields:

$$\mathbf{f}^{(t)} = \prod_{\ell=t_{k-1}}^{t-1} [\mathbf{I} + \tilde{\eta}_\ell\mathbf{D}]\mathbf{f}^{(t_{k-1})}. \tag{C.28}$$

Using the fact that $\bar{\mathbf{x}}^{(t)} = \mathbf{U}\mathbf{f}^{(t)} - \mathbf{B}^{-1}\mathbf{z}$, we obtain the following expression for $\bar{\mathbf{x}}^{(t)}$:

$$\bar{\mathbf{x}}^{(t)} = \mathbf{U}\prod_{\ell=t_{k-1}}^{t-1}(\mathbf{I}+\tilde{\eta}_\ell\mathbf{D})\mathbf{U}^{-1}\mathbf{e}_2\|\bar{\mathbf{w}}^{(t_{k-1})} - \mathbf{w}^*\| + \mathbf{U}\left[\prod_{\ell=t_{k-1}}^{t-1}(\mathbf{I}+\tilde{\eta}_\ell\mathbf{D}) - \mathbf{I}\right]\mathbf{U}^{-1}\mathbf{B}^{-1}\mathbf{z}. \tag{C.29}$$

(Part III) Finding the connection between $A^{(t)}$ and $\mathbf{x}^{(t)}$ and the expression for $A^{(t)}$: To bound the model dispersion across the clusters, we revisit (C.20), where we multiply its both hand side by $\sqrt{\beta}$, followed by taking square and expectation and follow a similar procedure used to obtain (C.13) to get:

$$\begin{aligned}\sqrt{\beta\mathbb{E}[\|\bar{\mathbf{w}}_c^{(t+1)} - \bar{\mathbf{w}}^{(t+1)}\|^2]} \leq & (1+\tilde{\eta}_t)\sqrt{\beta\mathbb{E}[\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2]} + \tilde{\eta}_t y^{(t)} \\ & + \tilde{\eta}_t[\tilde{\epsilon}_c^{(t)} + \sum_{d=1}^{N}\varrho_d\tilde{\epsilon}_d^{(t)} + \tilde{\delta} + 2\tilde{\sigma}],\end{aligned} \tag{C.30}$$

where $y^{(t)} = [1, 2\omega]\, \mathbf{x}^{(t)}$. Recursive expansion of (C.30) yields:

$$\sqrt{\beta \mathbb{E}[\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2]} \leq \sum_{\ell=t_{k-1}}^{t-1} \tilde{\eta}_\ell \prod_{\mathrm{j}=\ell+1}^{t-1} (1+\tilde{\eta}_\mathrm{j}) y^{(\ell)}$$
$$+ \sum_{\ell=t_{k-1}}^{t-1} \tilde{\eta}_\ell \prod_{\mathrm{j}=\ell+1}^{t-1} (1+\tilde{\eta}_\mathrm{j}) [\tilde{\epsilon}_c^{(0)} + \sum_{d=1}^{N} \varrho_d \tilde{\epsilon}_d^{(0)} + \tilde{\delta} + 2\tilde{\sigma}]. \tag{C.31}$$

The expression in (C.31) reveals the dependency of the difference between the model in one cluster c and the global average of models, i.e., the left hand side, on $y^{(t)}$ which by itself depends on $\mathbf{x}^{(t)}$. Considering the fact that $A^{(t)} \triangleq \mathbb{E}\left[\sum_{c=1}^{N} \varrho_c \left\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\right\|^2\right]$, the aforementioned dependency implies the dependency of $A^{(t)}$ on $\mathbf{x}^{(t)}$.

So, the key to obtain $A^{(t)}$ is to bound $y^{(t)}$, which can be expressed as follows:

$$\begin{aligned} y^{(t)} &= [1, 2\omega]\mathbf{x}^{(t)} \leq [1, 2\omega]\bar{\mathbf{x}}^{(t)} \\ &= [g_1 \Pi_{+,t} + g_2 \Pi_{-,t}] \sqrt{\beta} \|\bar{\mathbf{w}}(t_{k-1}) - \mathbf{w}^*\| \\ &+ [g_3(\Pi_{+,t} - \Pi_{0,t}) + g_4(\Pi_{-,t} - \Pi_{0,t})][\tilde{\sigma} + \sum_{d=1}^{N} \varrho_d \tilde{\epsilon}_d^{(0)}] \\ &+ [g_5(\Pi_{+,t} - \Pi_{0,t}) + g_6(\Pi_{-,t} - \Pi_{0,t})]\tilde{\delta}, \end{aligned} \tag{C.32}$$

where we define $\Pi_{\{+,-,0\},t} = \prod_{\ell=t_{k-1}}^{t-1} [1 + \tilde{\eta}_\ell \lambda_{\{+,-,0\}}]$, with λ_+ given by (C.25) and λ_- given by (C.26) and $\lambda_0 = 0$. Also, the constants g_1, g_2, g_3, g_4, g_5, and g_6 are given by:

$$g_1 \triangleq [1, 2\omega]\mathbf{U}\mathbf{e}_1\mathbf{e}_1^\top \mathbf{U}^{-1}\mathbf{e}_2 = \omega \left[1 - \frac{\tilde{\mu}/4}{\sqrt{(1+\tilde{\mu}/4)^2 + 2\omega}}\right] > 0,$$

$$g_2 \triangleq [1, 2\omega]\mathbf{U}\mathbf{e}_2\mathbf{e}_2^\top \mathbf{U}^{-1}\mathbf{e}_2 = \omega \left[1 + \frac{\tilde{\mu}/4}{\sqrt{(1+\tilde{\mu}/4)^2 + 2\omega}}\right] = g_2 = 2\omega - g_1 > 0,$$

$$g_3 \triangleq [1, 2\omega]\mathbf{U}\mathbf{e}_1\mathbf{e}_1^\top \mathbf{U}^{-1}\mathbf{B}^{-1}[2, 1]^\top = \frac{1}{2} + \frac{1 + \tilde{\mu}/4 + 2\omega}{2\sqrt{(1+\tilde{\mu}/4)^2 + 2\omega}} = g_3 > 1,$$

$$g_4 \triangleq [1, 2\omega]\mathbf{U}\mathbf{e}_2\mathbf{e}_2^\top \mathbf{U}^{-1}\mathbf{B}^{-1}[2, 1]^\top = \frac{1}{2} - \frac{1 + \tilde{\mu}/4 + 2\omega}{2\sqrt{(1+\tilde{\mu}/4)^2 + 2\omega}},$$

$$= -\omega \frac{1+2\omega+\tilde{\mu}/2}{\sqrt{(1+\tilde{\mu}/4)^2+2\omega}} \frac{1}{\sqrt{(1+\tilde{\mu}/4)^2+2\omega}+[1+\tilde{\mu}/4+2\omega]} = 1-g_3 < 0,$$

$$g_5 \triangleq [1,2\omega]\mathbf{U}\mathbf{e}_1\mathbf{e}_1^\top\mathbf{U}^{-1}\mathbf{B}^{-1}\mathbf{e}_1 = \frac{1}{[\tilde{\mu}+2\omega]\sqrt{(1+\tilde{\mu}/4)^2+2\omega}}$$

$$\cdot \frac{\frac{\tilde{\mu}}{2}(1+\tilde{\mu}/4)^2+\omega[1+\frac{5\tilde{\mu}}{4}+\tilde{\mu}^2/8]+2\omega^2+\sqrt{(1+\tilde{\mu}/4)^2+2\omega}[\frac{\tilde{\mu}}{2}(1+\tilde{\mu}/4)+\omega[1+\tilde{\mu}/2]]}{1+\tilde{\mu}/4+\sqrt{(1+\tilde{\mu}/4)^2+2\omega}} > 0,$$

$$g_6 \triangleq [1,2\omega]\mathbf{U}\mathbf{e}_2\mathbf{e}_2^\top\mathbf{U}^{-1}\mathbf{B}^{-1}\mathbf{e}_1$$

$$= \frac{\omega}{[\tilde{\mu}+2\omega]\sqrt{(1+\tilde{\mu}/4)^2+2\omega}} \frac{1+\frac{3\tilde{\mu}}{4}+2\omega+\sqrt{(1+\tilde{\mu}/4)^2+2\omega}}{1+\tilde{\mu}/4+\sqrt{(1+\tilde{\mu}/4)^2+2\omega}} = \frac{\tilde{\mu}/2+2\omega}{\tilde{\mu}+2\omega} - g_5 > 0.$$

Revisiting (C.30) with the result of (C.32) gives us:

$$\begin{aligned}\sqrt{\beta\mathbb{E}[\|\bar{\mathbf{w}}_c^{(t)}-\bar{\mathbf{w}}^{(t)}\|^2]} &\leq 2\omega\frac{g_1\Sigma_{+,t}+g_2\Sigma_{-,t}}{g_1+g_2}\sqrt{\beta}\|\bar{\mathbf{w}}(t_{k-1})-\mathbf{w}^*\| \\ &+ [\Sigma_{+,t}+(g_3-1)(\Sigma_{+,t}-\Sigma_{-,t})][\tilde{\sigma}+\sum_{d=1}^{N}\varrho_d\tilde{\epsilon}_d^{(0)}] \\ &+ \frac{\tilde{\mu}/2}{\tilde{\mu}+2\omega}[\frac{g_5}{g_5+g_6}\Sigma_{+,t}+\frac{g_6}{g_5+g_6}\Sigma_{-,t}+\Sigma_{0,t}]\tilde{\delta}+\Sigma_{0,t}[\tilde{\epsilon}_c^{(0)}+\tilde{\sigma}],\end{aligned} \tag{C.33}$$

where we used the facts that $g_3+g_4=1$, $g_5+g_6=\frac{\tilde{\mu}/2+2\omega}{\tilde{\mu}+2\omega}$, $g_1+g_2=2\omega$, and $g_3>1$, and defined $\Sigma_{+,t}$, $\Sigma_{-,t}$, and $\Sigma_{0,t}$ as follows:

$$\Sigma_{\{+,-,0\},t} = \sum_{\ell=t_{k-1}}^{t-1}\tilde{\eta}_\ell\prod_{\mathrm{j}=\ell+1}^{t-1}(1+\tilde{\eta}_\mathrm{j})\Pi_{\{+,-,0\},\ell} = \sum_{\ell=t_{k-1}}^{t-1}\left[\prod_{\mathrm{j}=t_{k-1}}^{\ell-1}(1+\tilde{\eta}_\mathrm{j}\lambda_{\{+,-,0\}})\right]\tilde{\eta}_\ell\left[\prod_{\mathrm{j}=\ell+1}^{t-1}(1+\tilde{\eta}_\mathrm{j})\right].$$

We now demonstrate that: (i) $\Sigma_{-,t}\leq\Sigma_{+,t}$, (ii) $\Sigma_{0,t}\leq\Sigma_{+,t}$, and (iii) $\Sigma_{-,t}\geq 0$.

To prove $\Sigma_{-,t}\leq\Sigma_{+,t}$, we upper bound $\Sigma_{-,t}$ as follows:

$$\begin{aligned}\Sigma_{-,t} &\leq \sum_{\ell=t_{k-1}}^{t-1}\left[\prod_{\mathrm{j}=t_{k-1}}^{\ell-1}|1+\tilde{\eta}_\mathrm{j}\lambda_-|\right]\tilde{\eta}_\ell\left[\prod_{\mathrm{j}=\ell+1}^{t-1}(1+\tilde{\eta}_\mathrm{j})\right] \\ &\leq \sum_{\ell=t_{k-1}}^{t-1}\left[\prod_{\mathrm{j}=t_{k-1}}^{\ell-1}(1+\tilde{\eta}_\mathrm{j}\lambda_+)\right]\tilde{\eta}_\ell\left[\prod_{\mathrm{j}=\ell+1}^{t-1}(1+\tilde{\eta}_\mathrm{j})\right] = \Sigma_{+,t}.\end{aligned} \tag{C.34}$$

Similarly it can be shown that $\Sigma_{0,t} \leq \Sigma_{+,t}$ since $\lambda_+ > 1$.

To prove $\Sigma_{-,t} \geq 0$, it is sufficient to impose the condition $(1 + \tilde{\eta}_{\mathrm{j}}\lambda_-) \geq 0, \forall \mathrm{j}$, i.e. $(1 + \tilde{\eta}_0\lambda_-) \geq 0$, which implies $\alpha \geq \tilde{\gamma}[\tilde{\mu}/4 - 1 + \sqrt{(1+\tilde{\mu}/4)^2 + 2\omega}]$.

Considering (C.33) with the above mentioned properties for $\Sigma_{-,t}$, $\Sigma_{+,t}$, and $\Sigma_{0,t}$, we get:

$$\begin{aligned}\sqrt{\beta \mathbb{E}[\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2]} \leq & 2\omega\Sigma_{+,t}\sqrt{\beta}\|\bar{\mathbf{w}}^{(t_{k-1})} - \mathbf{w}^*\| \\ & + g_3\Sigma_{+,t}[\tilde{\sigma} + \sum_{d=1}^{N} \varrho_d \tilde{\epsilon}_d^{(0)}] \\ & + \frac{\tilde{\mu}}{\tilde{\mu}+2\omega}\Sigma_{+,t}\tilde{\delta} + \Sigma_{+,t}[\tilde{\sigma} + \tilde{\epsilon}_c^{(0)}]. \end{aligned} \tag{C.35}$$

Moreover, since $\frac{\tilde{\mu}}{\tilde{\mu}+2\omega} \leq 1$, $\sum\limits_{d=1}^{N} \varrho_d \tilde{\epsilon}_d^{(0)} = \tilde{\epsilon}^{(0)}$ and $g_3 \leq \frac{1+\sqrt{3}}{2}$ (since g_3 is increasing with respect to ω and decreasing with respect to $\tilde{\mu}$), from (C.35) we obtain

$$\sqrt{\beta \mathbb{E}[\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2]} \leq 2\omega\Sigma_{+,t}\sqrt{\beta}\|\bar{\mathbf{w}}^{(t_{k-1})} - \mathbf{w}^*\| + \Sigma_{+,t}\left[\frac{3+\sqrt{3}}{2}\tilde{\sigma} + \frac{1+\sqrt{3}}{2}\tilde{\epsilon}^{(0)} + \tilde{\epsilon}_c^{(0)} + \tilde{\delta}\right]. \tag{C.36}$$

Taking the square of the both hand sides followed by taking the weighted sum $\sum_{c=1}^{N} \varrho_c$, we get:

$$\begin{aligned} & \beta A^{(t)} = \beta\mathbb{E}\left[\sum_{c=1}^{N} \varrho_c \|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2\right] \leq 8\omega^2[\Sigma_{+,t}]^2\beta\|\bar{\mathbf{w}}^{(t_{k-1})} - \mathbf{w}^*\|^2 \\ & + 2[\Sigma_{+,t}]^2 \sum_{c=1}^{N} \varrho_c \left[\frac{3+\sqrt{3}}{2}\tilde{\sigma} + \frac{1+\sqrt{3}}{2}\tilde{\epsilon}^{(0)} + \tilde{\epsilon}_c^{(0)} + \tilde{\delta}\right]^2 \\ & \leq 8\omega^2[\Sigma_{+,t}]^2\beta\|\bar{\mathbf{w}}^{(t_{k-1})} - \mathbf{w}^*\|^2 + 2[\Sigma_{+,t}]^2\left[\frac{3+\sqrt{3}}{2}\tilde{\sigma} + \frac{3+\sqrt{3}}{2}\tilde{\epsilon}^{(0)} + \tilde{\delta}\right]^2 \\ & \leq 8\omega^2[\Sigma_{+,t}]^2\beta\|\bar{\mathbf{w}}^{(t_{k-1})} - \mathbf{w}^*\|^2 + 25[\Sigma_{+,t}]^2\left[\tilde{\sigma}^2 + \tilde{\delta}^2 + (\tilde{\epsilon}^{(0)})^2\right]. \end{aligned} \tag{C.37}$$

Using the strong convexity of $F(.)$, we have $\|\bar{\mathbf{w}}(t_{k-1}) - \mathbf{w}^*\|^2 \leq \frac{2}{\tilde{\mu}\beta}[F(\bar{\mathbf{w}}(t_{k-1})) - F(\mathbf{w}^*)]$, using which in (C.37) yields:

$$\begin{aligned}\beta A^{(t)} &\leq \frac{16\omega^2}{\tilde{\mu}}[\Sigma_{+,t}]^2[F(\bar{\mathbf{w}}(t_{k-1})) - F(\mathbf{w}^*)] + 25[\Sigma_{+,t}]^2\left[\tilde{\sigma}^2 + (\tilde{\epsilon}^{(0)})^2 + \tilde{\delta}^2\right] \\ &= \frac{16\omega^2\beta}{\mu}[\Sigma_{+,t}]^2[F(\bar{\mathbf{w}}(t_{k-1})) - F(\mathbf{w}^*)] + 25[\Sigma_{+,t}]^2\left[\frac{\sigma^2}{\beta} + \frac{\delta^2}{\beta} + \beta(\epsilon^{(0)})^2\right]. \end{aligned} \tag{C.38}$$

This concludes the proofs.

□

C.2 Proof of Theorem C.2.1

Theorem C.2.1. *Under Assumptions 1, 2, and 3, upon using* `TT-HF` *for ML model training, if $\eta_t \leq 1/\beta$, $\forall t$, the one-step behavior of $\hat{\mathbf{w}}^{(t)}$ can be described as follows:*

$$\begin{aligned}\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)\right] \leq &(1-\mu\eta_t)\mathbb{E}[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)] \\ &+ \frac{\eta_t\beta^2}{2}A^{(t)} + \frac{1}{2}[\eta_t\beta^2(\epsilon^{(t)})^2 + \eta_t^2\beta\sigma^2 + \beta(\epsilon^{(t+1)})^2],\ t \in \mathcal{T}_k,\end{aligned}$$

where

$$A^{(t)} \triangleq \mathbb{E}\left[\sum_{c=1}^{N}\varrho_c\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|_2^2\right]. \tag{C.39}$$

Proof. Considering $t \in \mathcal{T}_k$, using (3.10), (3.13), the definition of $\bar{\mathbf{w}}$ given in Definition 3.2.1, and the fact that $\sum_{\mathrm{i}\in\mathcal{S}_c}\mathbf{e}_{\mathrm{i}}^{(t)} = 0$, $\forall t$, under Assumption 2, the global average of the local models follows the following dynamics:

$$\bar{\mathbf{w}}^{(t+1)} = \bar{\mathbf{w}}^{(t)} - \frac{\tilde{\eta}_t}{\beta}\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) - \frac{\tilde{\eta}_t}{\beta}\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\mathbf{n}_{\mathrm{j}}^{(t)}, \tag{C.40}$$

where $\mathbf{n}_{\mathrm{j}}^{(t)} = \widehat{\mathbf{g}}_{\mathrm{j}}^{(t)} - \nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})$. On the other hand, the β-smoothness of the global function F implies

$$F(\bar{\mathbf{w}}^{(t+1)}) \leq F(\bar{\mathbf{w}}^{(t)}) + \nabla F(\bar{\mathbf{w}}^{(t)})^\top (\bar{\mathbf{w}}^{(t+1)} - \bar{\mathbf{w}}^{(t)}) + \frac{\beta}{2}\left\|\bar{\mathbf{w}}^{(t+1)} - \bar{\mathbf{w}}^{(t)}\right\|^2. \tag{C.41}$$

Replacing the result of (C.40) in the above inequality, taking the conditional expectation (conditioned on the knowledge of $\bar{\mathbf{w}}^{(t)}$) of the both hand sides, and using the fact that $\mathbb{E}_t[\mathbf{n}_{\mathrm{j}}^{(t)}] = \mathbf{0}$ yields:

$$\begin{aligned}\mathbb{E}_t\left[F(\bar{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)\right] &\leq F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*) - \frac{\tilde{\eta}_t}{\beta}\nabla F(\bar{\mathbf{w}}^{(t)})^\top \sum_{c=1}^{N} \varrho_c \frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c} \nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) \\ &+ \frac{\tilde{\eta}_t^2}{2\beta}\left\|\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})\right\|^2 + \frac{\tilde{\eta}_t^2}{2\beta}\mathbb{E}_t\left[\left\|\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\mathbf{n}_{\mathrm{j}}^{(t)}\right\|^2\right].\end{aligned} \tag{C.42}$$

Since $\mathbb{E}_t[\|\mathbf{n}_{\mathrm{i}}^{(t)}\|_2^2] \leq \beta\tilde{\sigma}^2$, $\forall$i, we get

$$\begin{aligned}\mathbb{E}_t\left[F(\bar{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)\right] \leq & F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*) \\ & - \frac{\tilde{\eta}_t}{\beta}\nabla F(\bar{\mathbf{w}}^{(t)})^\top \sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) \\ & + \frac{\tilde{\eta}_t^2}{2\beta}\left\|\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})\right\|^2 + \frac{\tilde{\eta}_t^2\tilde{\sigma}^2}{2}.\end{aligned} \tag{C.43}$$

Using Lemma 3 (see Appendix C.5), we further bound (C.43) as follows:

$$\begin{aligned}&\mathbb{E}_t\left[F(\bar{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)\right] \leq (1-\tilde{\mu}\tilde{\eta}_t)(F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)) \\ &- \frac{\tilde{\eta}_t}{2\beta}(1-\tilde{\eta}_t)\left\|\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})\right\|^2 + \frac{\tilde{\eta}_t^2\tilde{\sigma}^2}{2} + \frac{\tilde{\eta}_t\beta}{2}\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\left\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}_{\mathrm{j}}^{(t)}\right\|^2 \\ &\leq (1-\tilde{\mu}\tilde{\eta}_t)(F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)) + \frac{\tilde{\eta}_t\beta}{2}\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\left\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}_{\mathrm{j}}^{(t)}\right\|^2 + \frac{\tilde{\eta}_t^2\tilde{\sigma}^2}{2},\end{aligned} \tag{C.44}$$

where the last step follows from $\tilde{\eta}_t \leq 1$. To further bound the terms on the right hand side of (C.44), we use the fact that

$$\|\mathbf{w}_{\mathrm{i}}^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2 = \|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2 + \|\mathbf{e}_{\mathrm{i}}^{(t)}\|^2 + 2[\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}]^\top \mathbf{e}_{\mathrm{i}}^{(t)}, \tag{C.45}$$

which results in

$$\frac{1}{s_c} \sum_{\mathrm{i} \in \mathcal{S}_c} \|\mathbf{w}_{\mathrm{i}}^{(t)} - \bar{\mathbf{w}}^{(t)}\|_2^2 \leq \|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|_2^2 + \frac{(\tilde{\epsilon}_c^{(t)})^2}{\beta}. \tag{C.46}$$

Replacing (C.46) in (C.44) and taking the unconditional expectation from the both hand sides of the resulting expression gives us

$$\begin{aligned}
&\mathbb{E}\left[F(\bar{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)\right] \leq (1 - \tilde{\mu}\tilde{\eta}_t)\mathbb{E}[F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)] \\
&+ \frac{\tilde{\eta}_t \beta}{2} \sum_{c=1}^{N} \varrho_c \left(\|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|_2^2 + \frac{(\tilde{\epsilon}_c^{(t)})^2}{\beta} \right) + \frac{\tilde{\eta}_t^2 \tilde{\sigma}^2}{2} \\
&= (1 - \tilde{\mu}\tilde{\eta}_t)\mathbb{E}[F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)] + \frac{\tilde{\eta}_t \beta}{2} A^{(t)} + \frac{1}{2}[\tilde{\eta}_t (\tilde{\epsilon}^{(t)})^2 + \tilde{\eta}_t^2 \tilde{\sigma}^2],
\end{aligned} \tag{C.47}$$

where

$$A^{(t)} \triangleq \mathbb{E}\left[\sum_{c=1}^{N} \varrho_c \|\bar{\mathbf{w}}_c^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2\right]. \tag{C.48}$$

By β-smoothness of $F(\cdot)$, we have

$$\begin{aligned}
&F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*) \leq F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*) + \nabla F(\bar{\mathbf{w}}^{(t)})^\top \left(\hat{\mathbf{w}}^{(t)} - \bar{\mathbf{w}}^{(t)}\right) + \frac{\beta}{2}\|\hat{\mathbf{w}}^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2 \\
&\leq F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*) + \nabla F(\bar{\mathbf{w}}^{(t)})^\top \sum_{c=1}^{N} \varrho_c \mathbf{e}_{s_c}^{(t)} + \frac{\beta}{2} \sum_{c=1}^{N} \varrho_c \|\mathbf{e}_{s_c}^{(t)}\|^2.
\end{aligned} \tag{C.49}$$

Taking the expectation with respect to the device sampling from both hand sides of (C.49), since the sampling is conducted uniformly at random, we obtain

$$\mathbb{E}_t\left[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)\right] \leq F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*) + \nabla F(\bar{\mathbf{w}}^{(t)})^\top \sum_{c=1}^{N} \varrho_c \underbrace{\mathbb{E}_t\left[\mathbf{e}_{s_c}^{(t)}\right]}_{=0} + \frac{\beta}{2}\sum_{c=1}^{N} \varrho_c \mathbb{E}_t\left[\|\mathbf{e}_{s_c}^{(t)}\|^2\right]. \tag{C.50}$$

Taking the total expectation from both hand sides of the above inequality yields:

$$\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)\right] \leq \mathbb{E}\left[F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)\right] + \frac{(\tilde{\epsilon}^{(t)})^2}{2}. \tag{C.51}$$

Replace (C.47) into (C.51), we have

$$\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)\right] \leq (1 - \tilde{\mu}\tilde{\eta}_t)\mathbb{E}[F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)] + \frac{\tilde{\eta}_t\beta}{2}A^{(t)} + \frac{1}{2}[\tilde{\eta}_t(\tilde{\epsilon}^{(t)})^2 + \tilde{\eta}_t^2\tilde{\sigma}^2 + (\tilde{\epsilon}^{(t+1)})^2]. \tag{C.52}$$

On the other hands, using the strong convexity of $F(\cdot)$, we have

$$\begin{aligned} F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*) &\geq F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*) + \nabla F(\bar{\mathbf{w}}^{(t)})^\top\left(\hat{\mathbf{w}}^{(t)} - \bar{\mathbf{w}}^{(t)}\right) + \frac{\mu}{2}\|\hat{\mathbf{w}}^{(t)} - \bar{\mathbf{w}}^{(t)}\|^2 \\ &\geq F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*) + \nabla F(\bar{\mathbf{w}}^{(t)})^\top \sum_{c=1}^{N} \varrho_c \mathbf{e}_{s_c}^{(t)}. \end{aligned} \tag{C.53}$$

Taking the expectation with respect to the device sampling from the both hand sides of (C.53), since the sampling is conducted uniformly at random, we obtain

$$\mathbb{E}_t\left[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)\right] \geq F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*) + \nabla F(\bar{\mathbf{w}}^{(t)})^\top \sum_{c=1}^{N} \varrho_c \underbrace{\mathbb{E}_t\left[\mathbf{e}_{s_c}^{(t)}\right]}_{=0}. \tag{C.54}$$

Taking the total expectation from both hand sides of the above inequality yields:

$$\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)\right] \geq \mathbb{E}\left[F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)\right]. \tag{C.55}$$

Finally, replacing (C.55) into (C.52), we obtain

$$
\begin{aligned}
&\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)\right] \leq (1-\tilde{\mu}\tilde{\eta}_t)\mathbb{E}[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)] \\
&+ \frac{\tilde{\eta}_t\beta}{2}A^{(t)} + \frac{1}{2}[\tilde{\eta}_t(\tilde{\epsilon}^{(t)})^2 + \tilde{\eta}_t^2\tilde{\sigma}^2 + (\tilde{\epsilon}^{(t+1)})^2] \\
&= (1-\mu\eta_t)\mathbb{E}[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)] + \frac{\eta_t\beta^2}{2}A^{(t)} + \frac{1}{2}[\eta_t\beta^2(\epsilon^{(t)})^2 + \eta_t^2\beta\sigma^2 + \beta(\epsilon^{(t+1)})^2]. \qquad \text{(C.56)}
\end{aligned}
$$

This concludes the proof.

□

C.3 Proof of Theorem C.3.1

Theorem C.3.1. *Define* $Z_1 \triangleq \frac{32\beta^2\gamma}{\mu}(\tau-1)\left(1+\frac{\tau}{\alpha-1}\right)^2\left(1+\frac{\tau-1}{\alpha-1}\right)^{6\beta\gamma}$, $Z_2 \triangleq \frac{1}{2}[\frac{\sigma^2}{\beta}+\frac{2\phi^2}{\beta}] + 50\beta\gamma(\tau-1)\left(1+\frac{\tau-2}{\alpha+1}\right)\left(1+\frac{\tau-1}{\alpha-1}\right)^{6\beta\gamma}\left[\frac{\sigma^2}{\beta}+\frac{\phi^2}{\beta}+\frac{\delta^2}{\beta}\right]$. *Also, assume* $\gamma > 1/\mu$, $\alpha \geq \max\{\beta\gamma[\frac{\vartheta}{4} - 1 + \sqrt{(1+\frac{\vartheta}{4})^2+2\omega}], \frac{\beta^2\gamma}{\mu}\}$ *and* $\omega < \frac{1}{\beta\gamma}\sqrt{\alpha\frac{\mu\gamma-1+\frac{1}{1+\alpha}}{Z_1}} \triangleq \omega_{\max}$. *Upon using* `TT-HF` *for ML model training under Assumptions 1, 2, and 3, if* $\eta_t = \frac{\gamma}{t+\alpha}$ *and* $\epsilon^{(t)} = \eta_t\phi$, $\forall t$, *we have:*

$$
\mathbb{E}\left[(F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*))\right] \leq \frac{\nu}{t+\alpha}, \quad \forall t, \qquad \text{(C.57)}
$$

where $\nu \triangleq Z_2 \max\left\{\frac{\beta^2\gamma^2}{\mu\gamma-1}, \frac{\alpha}{Z_1(\omega_{\max}^2-\omega^2)}, \frac{\alpha}{Z_2}\left[F(\hat{\mathbf{w}}^{(0)}) - F(\mathbf{w}^*)\right]\right\}$.

Proof. We carry out the proof by induction. We start by considering the first global aggregation, i.e., $k=1$. Note that the condition in (C.57) trivially holds at the beginning of this global aggregation $t = t_0 = 0$ since $\nu \geq \alpha\left[F(\hat{\mathbf{w}}^{(0)}) - F(\mathbf{w}^*)\right]$. Now, assume that

$$
\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t_{k-1})}) - F(\mathbf{w}^*)\right] \leq \frac{\nu}{t_{k-1}+\alpha} \qquad \text{(C.58)}
$$

for some $k \geq 1$. We prove that this implies

$$
\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)\right] \leq \frac{\nu}{t+\alpha}, \ \forall t \in \mathcal{T}_k, \qquad \text{(C.59)}
$$

and as a result $\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t_k)}) - F(\mathbf{w}^*)\right] \leq \frac{\nu}{t_k+\alpha}$. To prove (C.59), we use induction over $t \in \{t_{k-1}+1, \dots, t_k\}$. Clearly, the condition holds for $t = t_{k-1}$ from the induction hypothesis.

Now, we assume that it also holds for some $t \in \{t_{k-1}, \ldots, t_k - 1\}$, and aim to show that it holds at $t+1$.

From the result of Theorem C.2.1, considering $\tilde{\epsilon}^{(t)} = \tilde{\eta}_t \tilde{\phi}$, we get

$$\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)\right] \leq (1 - \tilde{\mu}\tilde{\eta}_t)\frac{\nu}{t+\alpha} + \frac{\tilde{\eta}_t \beta}{2} A^{(t)} + \frac{1}{2}[\tilde{\eta}_t^3 \tilde{\phi}^2 + \tilde{\eta}_t^2 \tilde{\sigma}^2 + \tilde{\eta}_{t+1}^2 \tilde{\phi}^2]. \tag{C.60}$$

Using the induction hypothesis and the bound on $A^{(t)}$, we can further upper bound (C.60) as

$$\begin{aligned}\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)\right] &\leq (1 - \tilde{\mu}\tilde{\eta}_t)\,\frac{\nu}{t+\alpha} + \frac{8\tilde{\eta}_t \omega^2}{\tilde{\mu}}[\Sigma_{+,t}]^2 \frac{\nu}{t_{k-1}+\alpha} \\ &+ \frac{25}{2}\tilde{\eta}_t [\Sigma_{+,t}]^2 \left[\tilde{\sigma}^2 + (\tilde{\epsilon}^{(0)})^2 + \tilde{\delta}^2\right] + \frac{1}{2}[\tilde{\eta}_t^3 \tilde{\phi}^2 + \tilde{\eta}_t^2 \tilde{\sigma}^2 + \tilde{\eta}_{t+1}^2 \tilde{\phi}^2].\end{aligned} \tag{C.61}$$

Since $\tilde{\eta}_{t+1} \leq \tilde{\eta}_t$, $\tilde{\eta}_t \leq \tilde{\eta}_0 \leq \tilde{\mu} \leq 1$ and $\tilde{\epsilon}^{(0)} = \tilde{\eta}_0 \tilde{\phi} \leq \tilde{\phi}$, we further upper bound (C.61) as

$$\begin{aligned}\mathbb{E}\left[F(\hat{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)\right] &\leq (1 - \tilde{\mu}\tilde{\eta}_t)\,\frac{\nu}{t+\alpha} + \frac{8\tilde{\eta}_t \omega^2}{\tilde{\mu}} \underbrace{[\Sigma_{+,t}]^2}_{(a)} \frac{\nu}{t_{k-1}+\alpha} \\ &+ \frac{25}{2}\tilde{\eta}_t \underbrace{[\Sigma_{+,t}]^2}_{(b)} \left[\tilde{\sigma}^2 + \tilde{\phi}^2 + \tilde{\delta}^2\right] + \frac{\tilde{\eta}_t^2}{2}[\tilde{\sigma}^2 + 2\tilde{\phi}^2].\end{aligned} \tag{C.62}$$

To get a tight upper bound for (C.62), we bound the two instances of $[\Sigma_{+,t}]^2$ appearing in (a) and (b) differently. In particular, for (a), we first use the fact that

$$\lambda_+ = 1 - \tilde{\mu}/4 + \sqrt{(1 + \tilde{\mu}/4)^2 + 2\omega} \in [2, 1 + \sqrt{3}],$$

which implies that

$$\begin{aligned}\Sigma_{+,t} &= \sum_{\ell=t_{k-1}}^{t-1} \left[\prod_{\mathrm{j}=t_{k-1}}^{\ell-1} (1 + \tilde{\eta}_{\mathrm{j}} \lambda_+)\right] \eta_\ell \left[\prod_{\mathrm{j}=\ell+1}^{t-1} (1 + \tilde{\eta}_{\mathrm{j}})\right] \\ &\leq \sum_{\ell=t_{k-1}}^{t-1} \left[\prod_{\mathrm{j}=t_{k-1}}^{\ell-1} (1 + \tilde{\eta}_{\mathrm{j}} \lambda_+)\right] \eta_\ell \left[\prod_{\mathrm{j}=\ell+1}^{t-1} (1 + \tilde{\eta}_{\mathrm{j}} \lambda_+)\right] \\ &\leq \left[\prod_{\mathrm{j}=t_{k-1}}^{t-1} (1 + \tilde{\eta}_{\mathrm{j}} \lambda_+)\right] \sum_{\ell=t_{k-1}}^{t-1} \frac{\tilde{\eta}_\ell}{1 + \tilde{\eta}_\ell \lambda_+}.\end{aligned} \tag{C.63}$$

Also, with the choice of step size $\tilde{\eta}_\ell = \frac{\tilde{\gamma}}{\ell+\alpha}$, we get

$$\Sigma_{+,t} \leq \tilde{\gamma} \underbrace{\left[\prod_{\mathrm{j}=t_{k-1}}^{t-1} \left(1 + \frac{\tilde{\gamma}\lambda_+}{\mathrm{j}+\alpha}\right)\right]}_{\text{(i)}} \underbrace{\sum_{\ell=t_{k-1}}^{t-1} \frac{1}{\ell+\alpha+\tilde{\gamma}\lambda_+}}_{\text{(ii)}} . \tag{C.64}$$

To bound (ii), since $\frac{1}{\ell+\alpha+\tilde{\gamma}\lambda_+}$ is a decreasing function with respect to ℓ, we have

$$\sum_{\ell=t_{k-1}}^{t-1} \frac{1}{\ell+\alpha+\tilde{\gamma}\lambda_+} \leq \int_{t_{k-1}-1}^{t-1} \frac{1}{\ell+\alpha+\tilde{\gamma}\lambda_+} \mathrm{d}\ell = \ln\left(1 + \frac{t-t_{k-1}}{t_{k-1}-1+\alpha+\tilde{\gamma}\lambda_+}\right), \tag{C.65}$$

where we used the fact that $\alpha > 1 - \tilde{\gamma}\lambda_+$ (implied by $\alpha > 1$).

To bound (i), we first rewrite it as follows:

$$\prod_{\mathrm{j}=t_{k-1}}^{t-1} \left(1 + \frac{\tilde{\gamma}\lambda_+}{\mathrm{j}+\alpha}\right) = \mathrm{e}^{\sum_{\mathrm{j}=t_{k-1}}^{t-1} \ln\left(1+\frac{\tilde{\gamma}\lambda_+}{\mathrm{j}+\alpha}\right)} \tag{C.66}$$

To bound (C.66), we use the fact that $\ln(1+\frac{\tilde{\gamma}\lambda_+}{\mathrm{j}+\alpha})$ is a decreasing function with respect to j, and $\alpha > 1$, to get

$$\begin{aligned}\sum_{\mathrm{j}=t_{k-1}}^{t-1} \ln(1+\frac{\tilde{\gamma}\lambda_+}{\mathrm{j}+\alpha}) &\leq \int_{t_{k-1}-1}^{t-1} \ln(1+\frac{\tilde{\gamma}\lambda_+}{\mathrm{j}+\alpha})\mathrm{dj} \\ &\leq \tilde{\gamma}\lambda_+ \int_{t_{k-1}-1}^{t-1} \frac{1}{\mathrm{j}+\alpha}\mathrm{dj} = \tilde{\gamma}\lambda_+ \ln\left(1 + \frac{t-t_{k-1}}{t_{k-1}-1+\alpha}\right).\end{aligned} \tag{C.67}$$

Considering (C.66) and (C.67) together, we bound (i) as follows:

$$\prod_{\mathrm{j}=t_{k-1}}^{t-1} \left(1 + \frac{\tilde{\gamma}\lambda_+}{\mathrm{j}+\alpha}\right) \leq \left(1 + \frac{t-t_{k-1}}{t_{k-1}-1+\alpha}\right)^{\tilde{\gamma}\lambda_+}. \tag{C.68}$$

Using the results obtained for bounding (i) and (ii) back in (C.64), we get:

$$\Sigma_{+,t} \leq \tilde{\gamma} \ln\left(1 + \frac{t-t_{k-1}}{t_{k-1}-1+\alpha+\tilde{\gamma}\lambda_+}\right)\left(1 + \frac{t-t_{k-1}}{t_{k-1}-1+\alpha}\right)^{\tilde{\gamma}\lambda_+}. \tag{C.69}$$

Since $\ln(1+x) \leq \ln(1+x+2\sqrt{x}) = \ln((1+\sqrt{x})^2) = 2\ln(1+\sqrt{x}) \leq 2\sqrt{x}$ for $x \geq 0$, we can further bound (C.69) as follows:

$$\begin{aligned}\Sigma_{+,t} &\leq 2\tilde{\gamma}\sqrt{\frac{t-t_{k-1}}{t_{k-1}-1+\alpha+\tilde{\gamma}\lambda_+}}\left(1+\frac{t-t_{k-1}}{t_{k-1}+\alpha-1}\right)^{\tilde{\gamma}\lambda_+} \\ &\leq 2\tilde{\gamma}\sqrt{\frac{t-t_{k-1}}{t_{k-1}+\alpha+1}}\left(1+\frac{t-t_{k-1}}{t_{k-1}+\alpha-1}\right)^{3\tilde{\gamma}},\end{aligned} \tag{C.70}$$

where in the last inequality we used $2 \leq \lambda_+ < 3$ and $\tilde{\gamma} \geq \frac{\tilde{\mu}}{\beta}\tilde{\gamma} > 1$. Taking the square from the both hand sides of (C.70) followed by multiplying the both hand sides with $\frac{[t+\alpha]^2}{\tilde{\mu}\tilde{\gamma}[t_{k-1}+\alpha]}$ gives us:

$$\begin{aligned}&[\Sigma_{+,t}]^2\frac{[t+\alpha]^2}{\tilde{\mu}\tilde{\gamma}[t_{k-1}+\alpha]} \leq \frac{4\tilde{\gamma}}{\tilde{\mu}}\frac{[t-t_{k-1}][t+\alpha]^2}{[t_{k-1}+\alpha+1][t_{k-1}+\alpha]}\left(1+\frac{t-t_{k-1}}{t_{k-1}+\alpha-1}\right)^{6\tilde{\gamma}} \\ &\leq \frac{4\tilde{\gamma}}{\tilde{\mu}}\frac{[t-t_{k-1}][t+\alpha]^2}{[t_{k-1}+\alpha-1]^2}\frac{[t_{k-1}+\alpha-1]^2}{[t_{k-1}+\alpha+1][t_{k-1}+\alpha]}\left(1+\frac{\tau-1}{t_{k-1}+\alpha-1}\right)^{-2}\left(1+\frac{\tau-1}{t_{k-1}+\alpha-1}\right)^{6\tilde{\gamma}+2} \\ &\overset{(a)}{\leq} \frac{4\tilde{\gamma}}{\tilde{\mu}}\frac{[\tau-1][t_{k-1}+\tau-1+\alpha]^2}{[t_{k-1}+\alpha-1]^2}\left(\frac{t_{k-1}+\alpha+\tau-2}{t_{k-1}+\alpha-1}\right)^{-2} \\ &\qquad \frac{[t_{k-1}+\alpha-1]^2}{[t_{k-1}+\alpha+1][t_{k-1}+\alpha]}\left(1+\frac{\tau-1}{t_{k-1}+\alpha-1}\right)^{6\tilde{\gamma}+2} \\ &\leq \frac{4\tilde{\gamma}}{\tilde{\mu}}(\tau-1)\left(1+\frac{1}{\tau+t_{k-1}+\alpha-2}\right)^2\frac{[t_{k-1}+\alpha-1]^2}{[t_{k-1}+\alpha+1][t_{k-1}+\alpha]}\left(1+\frac{\tau-1}{t_{k-1}+\alpha-1}\right)^{6\tilde{\gamma}+2},\end{aligned} \tag{C.71}$$

where (a) comes from the fact that $t \leq t_{k-1}+\tau_k-1 \leq t_{k-1}+\tau-1$. To bound (C.71), we use the facts that

$$1+\frac{1}{\tau+t_{k-1}+\alpha-2} \leq 1+\frac{1}{\tau+\alpha-2},\ 1+\frac{\tau-1}{t_{k-1}+\alpha-1} \leq 1+\frac{\tau-1}{\alpha-1}, \tag{C.72}$$

and

$$\frac{[t_{k-1}+\alpha-1]^2}{[t_{k-1}+\alpha+1][t_{k-1}+\alpha]} \leq 1, \tag{C.73}$$

which yield

$$[\Sigma_{+,t}]^2 \frac{[t+\alpha]^2}{\tilde{\mu}\tilde{\gamma}[t_{k-1}+\alpha]} \leq \frac{4\tilde{\gamma}}{\tilde{\mu}}(\tau-1)\left(1+\frac{\tau}{\alpha-1}\right)^2\left(1+\frac{\tau-1}{\alpha-1}\right)^{6\tilde{\gamma}}. \tag{C.74}$$

Consequently, we have

$$[\Sigma_{+,t}]^2 \leq 4(\tau-1)\left(1+\frac{\tau}{\alpha-1}\right)^2\left(1+\frac{\tau-1}{\alpha-1}\right)^{6\tilde{\gamma}} \tilde{\eta}_t^2[t_{k-1}+\alpha]. \tag{C.75}$$

On the other hand, we bound the second instance of $[\Sigma_{+,t}]^2$, i.e., (b) in (C.62), as follows:

$$\begin{aligned}
[t+\alpha][\Sigma_{+,t}]^2 &\leq 4\tilde{\gamma}^2\frac{[t-t_{k-1}][t+\alpha]}{t_{k-1}+\alpha+1}\left(1+\frac{t-t_{k-1}}{t_{k-1}+\alpha-1}\right)^{6\tilde{\gamma}} \\
&\leq 4\tilde{\gamma}^2(\tau-1)\left(1+\frac{\tau-2}{t_{k-1}+\alpha+1}\right)\left(1+\frac{\tau-1}{t_{k-1}+\alpha-1}\right)^{6\tilde{\gamma}} \\
&\leq 4\tilde{\gamma}^2(\tau-1)\left(1+\frac{\tau-2}{\alpha+1}\right)\left(1+\frac{\tau-1}{\alpha-1}\right)^{6\tilde{\gamma}},
\end{aligned} \tag{C.76}$$

which implies

$$[\Sigma_{+,t}]^2 \leq 4\tilde{\gamma}(\tau-1)\left(1+\frac{\tau-2}{\alpha+1}\right)\left(1+\frac{\tau-1}{\alpha-1}\right)^{6\tilde{\gamma}} \tilde{\eta}_t. \tag{C.77}$$

Replacing (C.75) and (C.77) into (C.62), we get

$$\mathbb{E}[F(\hat{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)] \leq \left(1-\tilde{\mu}\tilde{\eta}_t + Z_1\omega^2\tilde{\eta}_t^2\right)\frac{\nu}{t+\alpha} + \tilde{\eta}_t^2 Z_2, \tag{C.78}$$

where we have defined

$$Z_1 \triangleq \frac{32\tilde{\gamma}}{\tilde{\mu}}(\tau-1)\left(1+\frac{\tau}{\alpha-1}\right)^2\left(1+\frac{\tau-1}{\alpha-1}\right)^{6\tilde{\gamma}}, \tag{C.79}$$

and

$$Z_2 \triangleq \frac{1}{2}[\tilde{\sigma}^2 + 2\tilde{\phi}^2] + 50\tilde{\gamma}(\tau-1)\left(1+\frac{\tau-2}{\alpha+1}\right)\left(1+\frac{\tau-1}{\alpha-1}\right)^{6\tilde{\gamma}}\left[\tilde{\sigma}^2+\tilde{\phi}^2+\tilde{\delta}^2\right]. \tag{C.80}$$

Now, from (C.78), to complete the induction, we aim to show that

$$\mathbb{E}[F(\hat{\mathbf{w}}^{(t+1)}) - F(\mathbf{w}^*)] \leq \left(1 - \tilde{\mu}\tilde{\eta}_t + Z_1\omega^2\tilde{\eta}_t^2\right)\frac{\nu}{t+\alpha} + \tilde{\eta}_t^2 Z_2 \leq \frac{\nu}{t+1+\alpha}. \tag{C.81}$$

We transform the condition in (C.81) through the set of following algebraic steps to an inequality condition on a convex function:

$$\begin{aligned}
&\left(-\frac{\tilde{\mu}}{\tilde{\eta}_t^2} + \frac{Z_1\omega^2}{\tilde{\eta}_t}\right)\frac{\nu}{t+\alpha} + \frac{Z_2}{\tilde{\eta}_t} + \frac{\nu}{t+\alpha}\frac{1}{\tilde{\eta}_t{}^3} \leq \frac{\nu}{t+1+\alpha}\frac{1}{\tilde{\eta}_t{}^3}\\
&\Rightarrow \left(-\frac{\tilde{\mu}}{\tilde{\eta}_t} + Z_1\omega^2\right)\frac{\nu}{t+\alpha}\frac{1}{\tilde{\eta}_t} + \frac{Z_2}{\tilde{\eta}_t} + \left(\frac{\nu}{t+\alpha} - \frac{\nu}{t+1+\alpha}\right)\frac{1}{\tilde{\eta}_t{}^3} \leq 0\\
&\Rightarrow \left(-\frac{\tilde{\mu}}{\tilde{\eta}_t} + Z_1\omega^2\right)\frac{\nu}{\tilde{\gamma}} + \frac{Z_2}{\tilde{\eta}_t} + \left(\frac{\nu}{t+\alpha} - \frac{\nu}{t+1+\alpha}\right)\frac{(t+\alpha)^3}{\tilde{\gamma}^3} \leq 0\\
&\Rightarrow \left(-\frac{\tilde{\mu}}{\tilde{\eta}_t} + Z_1\omega^2\right)\frac{\nu}{\tilde{\gamma}} + \frac{Z_2}{\tilde{\eta}_t} + \frac{\nu}{(t+\alpha)(t+\alpha+1)}\frac{(t+\alpha)^3}{\tilde{\gamma}^3} \leq 0\\
&\Rightarrow \left(-\frac{\tilde{\mu}}{\tilde{\eta}_t} + Z_1\omega^2\right)\frac{\nu}{\tilde{\gamma}} + \frac{Z_2}{\tilde{\eta}_t} + \frac{\nu}{t+\alpha+1}\frac{(t+\alpha)^2}{\tilde{\gamma}^3} \leq 0\\
&\Rightarrow \tilde{\gamma}^2\left(-\frac{\tilde{\mu}}{\tilde{\eta}_t} + Z_1\omega^2\right)\nu + \frac{Z_2}{\tilde{\eta}_t}\tilde{\gamma}^3 + \frac{(t+\alpha)^2}{t+\alpha+1}\nu \leq 0\\
&\Rightarrow \tilde{\gamma}^2\left(-\frac{\tilde{\mu}}{\tilde{\eta}_t} + Z_1\omega^2\right)\nu + \frac{Z_2}{\tilde{\eta}_t}\tilde{\gamma}^3 + \left(\frac{(t+\alpha+1)(t+\alpha-1)}{t+\alpha+1}\nu + \frac{\nu}{t+\alpha+1}\right) \leq 0,
\end{aligned} \tag{C.82}$$

where the last condition in (C.82) can be written as:

$$\tilde{\gamma}^2\left(-\frac{\tilde{\mu}}{\tilde{\eta}_t} + Z_1\omega^2\right)\nu + \frac{Z_2}{\tilde{\eta}_t}\tilde{\gamma}^3 + \nu[t+\alpha-1] + \frac{\nu}{t+1+\alpha} \leq 0. \tag{C.83}$$

Since the above condition needs to be satisfied $\forall t \geq 0$ and the expression on the left hand side of the inequality is a convex function with respect to t ($1/\eta_t$ is linear in t and $\frac{1}{t+1+\alpha}$ is convex), it is sufficient to satisfy this condition for $t \to \infty$ and $t = 0$. To obtain these limits, we first express (C.83) as follows:

$$\tilde{\gamma}^2\left(-\frac{\tilde{\mu}}{\tilde{\gamma}}(t+\alpha) + Z_1\omega^2\right)\nu + Z_2\tilde{\gamma}^2(t+\alpha) + \nu[t+\alpha-1] + \frac{\nu}{t+1+\alpha} \leq 0. \tag{C.84}$$

Upon $t \to \infty$ considering the dominant terms yields

$$\begin{aligned} & -\tilde{\gamma}\tilde{\mu}\nu t + Z_2\tilde{\gamma}^2 t + \nu t \leq 0 \\ & \Rightarrow \left[1 - \tilde{\gamma}\tilde{\mu}\right]\nu t + Z_2\tilde{\gamma}^2 t \leq 0. \end{aligned} \tag{C.85}$$

To satisfy (C.85), the necessary condition is given by:

$$\tilde{\mu}\tilde{\gamma} - 1 > 0, \tag{C.86}$$

$$\nu \geq \frac{\tilde{\gamma}^2 Z_2}{\tilde{\mu}\tilde{\gamma} - 1}. \tag{C.87}$$

Also, upon $t \to 0$, from (C.84) we have

$$\begin{aligned} & \left(-\tilde{\mu}\tilde{\gamma}\alpha + Z_1\omega^2\tilde{\gamma}^2\right)\nu + Z_2\tilde{\gamma}^2\alpha + \nu[\alpha - 1] + \frac{\nu}{1+\alpha} \leq 0 \\ & \Rightarrow \nu\left(\alpha(\tilde{\mu}\tilde{\gamma} - 1) + \frac{\alpha}{1+\alpha} - Z_1\omega^2\tilde{\gamma}^2\right) \geq \tilde{\gamma}^2 Z_2\alpha, \end{aligned} \tag{C.88}$$

which implies the following conditions

$$\omega < \frac{1}{\tilde{\gamma}}\sqrt{\alpha\frac{\tilde{\mu}\tilde{\gamma} - 1 + \frac{1}{1+\alpha}}{Z_1}}, \tag{C.89}$$

and

$$\nu \geq \frac{Z_2\alpha}{Z_1\left(\omega_{\max}^2 - \omega^2\right)}. \tag{C.90}$$

Combining (C.87) and (C.90), when ω satisfies (C.89) and

$$\nu \geq Z_2 \max\{\frac{\beta^2\gamma^2}{\mu\gamma - 1}, \frac{\alpha}{Z_1\left(\omega_{\max}^2 - \omega^2\right)}\}, \tag{C.91}$$

completes the induction and thus the proof.

□

C.4 Proof of Lemma 2

Lemma 2. *After performing $\Gamma_c^{(t)}$ rounds of consensus in cluster $\mathcal{S}_c$ with the consensus matrix $\mathbf{V}_c$, the consensus error $\mathbf{e}_{\mathrm{i}}^{(t)}$ satisfies*

$$\|\mathbf{e}_{\mathrm{i}}^{(t)}\| \leq (\lambda_c)^{\Gamma_c^{(t)}} \sqrt{s_c}\, \underbrace{\max_{\mathrm{j},\mathrm{j}' \in \mathcal{S}_c} \|\tilde{\mathbf{w}}_{\mathrm{j}}^{(t)} - \tilde{\mathbf{w}}_{\mathrm{j}'}^{(t)}\|}_{\triangleq \Upsilon_c^{(t)}}, \;\; \forall \mathrm{i} \in \mathcal{S}_c, \tag{C.92}$$

where $\lambda_c = \rho\Big(\mathbf{V}_c - \frac{\mathbf{\mathit{1}}\mathbf{\mathit{1}}^\top}{s_c}\Big)$ and $\|\mathbf{a}\|_\infty = \max_z |\mathbf{a}_z|$ denotes the ℓ_∞ norm.

Proof. The evolution of the devices' parameters can be described by (3.12) as:

$$\mathbf{W}_c^{(t)} = (\mathbf{V}_c)^{\Gamma_c^{(t)}} \widetilde{\mathbf{W}}_c^{(t)}, \; t \in \mathcal{T}_k, \tag{C.93}$$

where

$$\mathbf{W}_c^{(t)} = \Big[\mathbf{w}_{c_1}^{(t)}, \mathbf{w}_{c_2}^{(t)}, \ldots, \mathbf{w}_{s_c}^{(t)}\Big]^\top \tag{C.94}$$

and

$$\widetilde{\mathbf{W}}_c^{(t)} = \Big[\tilde{\mathbf{w}}_{c_1}^{(t)}, \tilde{\mathbf{w}}_{c_2}^{(t)}, \ldots, \tilde{\mathbf{w}}_{s_c}^{(t)}\Big]^\top. \tag{C.95}$$

Let matrix $\overline{\mathbf{W}}_c^{(t)}$ denote be the matrix with rows given by the average model parameters across the cluster, it can be represented as:

$$\overline{\mathbf{W}}_c^{(t)} = \frac{\mathbf{1}_{s_c}\mathbf{1}_{s_c}^\top \widetilde{\mathbf{W}}_c^{(t)}}{s_c}. \tag{C.96}$$

We then define $\mathbf{E}_c^{(t)}$ as

$$\mathbf{E}_c^{(t)} = \mathbf{W}_c^{(t)} - \overline{\mathbf{W}}_c^{(t)} = [\,(\mathbf{V}_c)^{\Gamma_c^{(t)}} - \mathbf{1}^\top \mathbf{1}/s_c][\widetilde{\mathbf{W}}_c^{(t)} - \overline{\mathbf{W}}_c^{(t)}], \tag{C.97}$$

so that $[\mathbf{E}_c^{(t)}]_{\mathrm{i},:} = \mathbf{e}_{\mathrm{i}}^{(t)}$, where $[\mathbf{E}_c^{(t)}]_{\mathrm{i},:}$ is the ith column of $\mathbf{E}_c^{(t)}$.

Therefore, using Assumption 2, we can bound the consensus error as

$$
\begin{aligned}
&\|\mathbf{e}_{\mathrm{i}}^{(t)}\|^2 \leq \mathrm{trace}((\mathbf{E}_c^{(t)})^\top \mathbf{E}_c^{(t)}) \qquad\qquad \text{(C.98)}\\
&= \mathrm{trace}\Big([\widetilde{\mathbf{W}}_c^{(t)} - \overline{\mathbf{W}}_c^{(t)}]^\top [\,(\mathbf{V}_c)^{\Gamma_c^{(t)}} - \mathbf{1}^\top \mathbf{1}/s_c]^2 [\widetilde{\mathbf{W}}_c^{(t)} - \overline{\mathbf{W}}_c^{(t)}]\Big)\\
&\leq (\lambda_c)^{2\Gamma_c^{(t)}} \sum_{\mathrm{j}=1}^{s_c} \|\tilde{\mathbf{w}}_{\mathrm{j}}^{(t)} - \bar{\mathbf{w}}_c^{(t)}\|^2\\
&\leq (\lambda_c)^{2\Gamma_c^{(t)}} \frac{1}{s_c} \sum_{\mathrm{j},\mathrm{j}'=1}^{s_c} \|\tilde{\mathbf{w}}_{\mathrm{j}}^{(t)} - \tilde{\mathbf{w}}_{\mathrm{j}'}^{(t)}\|^2\\
&\leq (\lambda_c)^{2\Gamma_c^{(t)}} s_c \max_{\mathrm{j},\mathrm{j}' \in \mathcal{S}_c} \|\tilde{\mathbf{w}}_{\mathrm{j}}^{(t)} - \tilde{\mathbf{w}}_{\mathrm{j}'}^{(t)}\|^2.
\end{aligned}
$$

The result of the Lemma directly follows. □

C.5 Proof of Lemma 3

Lemma 3. *Under Assumption 1, we have*

$$
\begin{aligned}
&-\frac{\tilde{\eta}_t}{\beta} \nabla F(\bar{\mathbf{w}}^{(t)})^\top \sum_{c=1}^{N} \varrho_c \frac{1}{s_c} \sum_{\mathrm{j}\in\mathcal{S}_c} \nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) \leq -\tilde{\mu}\tilde{\eta}_t (F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*))\\
&-\frac{\tilde{\eta}_t}{2\beta} \Big\| \sum_{c=1}^{N} \varrho_c \frac{1}{s_c} \sum_{\mathrm{j}\in\mathcal{S}_c} \nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) \Big\|^2 + \frac{\tilde{\eta}_t \beta}{2} \sum_{c=1}^{N} \varrho_c \frac{1}{s_c} \sum_{\mathrm{j}\in\mathcal{S}_c} \Big\| \bar{\mathbf{w}}^{(t)} - \mathbf{w}_{\mathrm{j}}^{(t)} \Big\|^2.
\end{aligned}
$$

Proof. Since $-2\mathbf{a}^\top\mathbf{b} = -\|\mathbf{a}\|^2 - \|\mathbf{b}\|^2 + \|\mathbf{a}-\mathbf{b}\|^2$ holds for any two vectors $\mathbf{a}$ and $\mathbf{b}$ with real elements, we have

$$
\begin{aligned}
&-\frac{\tilde{\eta}_t}{\beta} \nabla F(\bar{\mathbf{w}}^{(t)})^\top \sum_{c=1}^{N} \varrho_c \frac{1}{s_c} \sum_{\mathrm{j}\in\mathcal{S}_c} \nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})\\
&= \frac{\tilde{\eta}_t}{2\beta} \Bigg[- \Big\| \nabla F(\bar{\mathbf{w}}^{(t)}) \Big\|^2 - \Big\| \sum_{c=1}^{N} \varrho_c \frac{1}{s_c} \sum_{\mathrm{j}\in\mathcal{S}_c} \nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) \Big\|^2\\
&+ \Big\| \nabla F(\bar{\mathbf{w}}^{(t)}) - \sum_{c=1}^{N} \varrho_c \frac{1}{s_c} \sum_{\mathrm{j}\in\mathcal{S}_c} \nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) \Big\|^2 \Bigg]. \qquad\qquad \text{(C.99)}
\end{aligned}
$$

Since $\|\cdot\|^2$ is a convex function, using Jenson's inequality, we get: $\|\sum_{\mathrm{i}=1}^{\mathrm{j}} c_{\mathrm{j}}\mathbf{a}_{\mathrm{j}}\|^2 \leq \sum_{\mathrm{i}=1}^{\mathrm{j}} c_{\mathrm{j}}\|\mathbf{a}_{\mathrm{j}}\|^2$, where $\sum_{\mathrm{i}=1}^{\mathrm{j}} c_{\mathrm{j}} = 1$. Using this fact in (C.99) yields

$$\begin{aligned}
&-\frac{\tilde{\eta}_t}{\beta}\nabla F(\bar{\mathbf{w}}^{(t)})^\top \sum_{c=1}^{N} \varrho_c \frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c} \nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) \\
&\leq \frac{\tilde{\eta}_t}{2\beta}\Bigg[-\left\|\nabla F(\bar{\mathbf{w}}^{(t)})\right\|^2 - \left\|\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})\right\|^2 \\
&+\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\left\|\nabla F_{\mathrm{j}}(\bar{\mathbf{w}}^{(t)}) - \nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})\right\|^2\Bigg]. \qquad \text{(C.100)}
\end{aligned}$$

Using μ-strong convexity of $F(.)$, we get: $\left\|\nabla F(\bar{\mathbf{w}}^{(t)})\right\|^2 \geq 2\tilde{\mu}\beta(F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*))$. Also, using β-smoothness of $F_{\mathrm{j}}(\cdot)$ we get $\left\|\nabla F_{\mathrm{j}}(\bar{\mathbf{w}}^{(t)}) - \nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})\right\|^2 \leq \beta^2\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}_{\mathrm{j}}^{(t)}\Big\|^2$, $\forall$j. Using these facts in (C.100) yields:

$$\begin{aligned}
&-\frac{\tilde{\eta}_t}{\beta}\nabla F(\bar{\mathbf{w}}^{(t)})^\top \sum_{c=1}^{N} \varrho_c \frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c} \nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)}) \leq -\tilde{\mu}\tilde{\eta}_t(F(\bar{\mathbf{w}}^{(t)}) - F(\mathbf{w}^*)) \\
&-\frac{\tilde{\eta}_t}{2\beta}\left\|\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\nabla F_{\mathrm{j}}(\mathbf{w}_{\mathrm{j}}^{(t)})\right\|^2 + \frac{\tilde{\eta}_t\beta}{2}\sum_{c=1}^{N}\varrho_c\frac{1}{s_c}\sum_{\mathrm{j}\in\mathcal{S}_c}\left\|\bar{\mathbf{w}}^{(t)} - \mathbf{w}_{\mathrm{j}}^{(t)}\right\|^2, \qquad \text{(C.101)}
\end{aligned}$$

which concludes the proof. □

C.6 Proof of Fact 1

Fact 1. *For an arbitrary set of n random variables $x_1, \cdots, x_n$, we have:*

$$\sqrt{\mathbb{E}\left[\left(\sum_{\mathrm{i}=1}^{n} x_{\mathrm{i}}\right)^2\right]} \leq \sum_{\mathrm{i}=1}^{n}\sqrt{\mathbb{E}[x_{\mathrm{i}}^2]}. \qquad \text{(C.102)}$$

Proof. The proof can be carried out through the following set of algebraic manipulations:

$$\begin{aligned}
&\sqrt{\mathbb{E}\left[\left(\sum_{\mathrm{i}=1}^{n} x_{\mathrm{i}}\right)^2\right]} = \sqrt{\sum_{\mathrm{i}=1}^{n}\mathbb{E}[x_{\mathrm{i}}^2] + \sum_{\mathrm{i}=1}^{n}\sum_{\mathrm{j}=1,\mathrm{j}\neq\mathrm{i}}^{n}\mathbb{E}[x_{\mathrm{i}}x_{\mathrm{j}}]} \\
&\overset{(a)}{\leq} \sqrt{\sum_{\mathrm{i}=1}^{n}\mathbb{E}[x_{\mathrm{i}}^2] + \sum_{\mathrm{i}=1}^{n}\sum_{\mathrm{j}=1,\mathrm{j}\neq\mathrm{i}}^{n}\sqrt{\mathbb{E}[x_{\mathrm{i}}^2]\mathbb{E}[x_{\mathrm{j}}^2]]}} = \sqrt{\left(\sum_{\mathrm{i}=1}^{n}\sqrt{\mathbb{E}[x_{\mathrm{i}}^2]}\right)^2} = \sum_{\mathrm{i}=1}^{n}\sqrt{\mathbb{E}[x_{\mathrm{i}}^2]}, \qquad \text{(C.103)}
\end{aligned}$$

where (a) is due to the fact that $\mathbb{E}[XY] \leq \sqrt{\mathbb{E}[X^2]\mathbb{E}[Y^2]}$ resulted from Cauchy-Schwarz inequality. $\square$

D. ADDITIONAL EXPERIMENTAL RESULTS

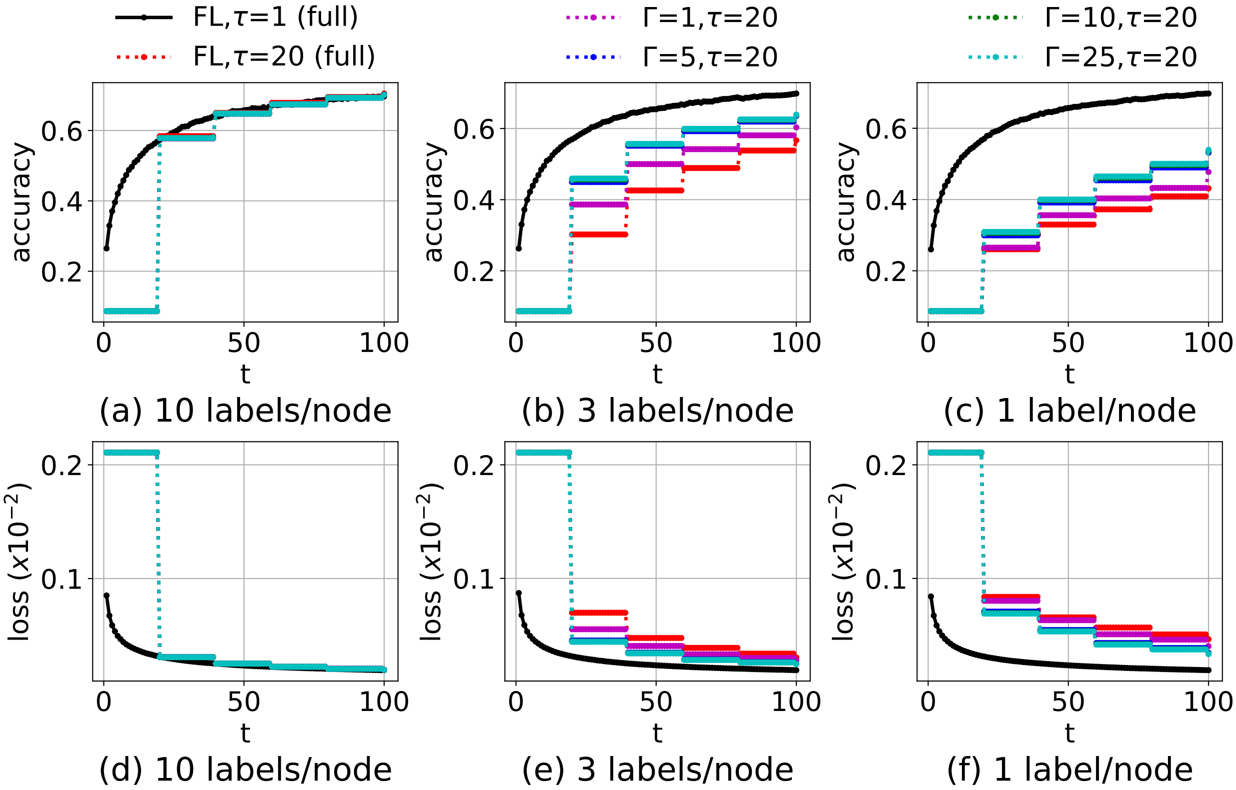
D.1 Complimentary Experiments of the Main Text

This section presents the plots from complimentary experiments mentioned in Chapter 3.4. We use an additional dataset, Fashion-MNIST (FMNIST), and fully connected neural networks (FCN) for these additional experiments. FMNIST (https://github.com/zalandoresearch/fashion-mnist) is a dataset of clothing articles consisting of a training set of 60,000 samples and a test set of 10,000 samples. Each sample is a 28x28 grayscale image, associated with a label from 10 classes.

In the following, we explain the relationship between the figures presented in this appendix and the results presented in the main text. Overall, we find that the results are qualitatively similar to what was observed for the SVM and MNIST cases:

- Fig. 3.3 from main text is repeated in Fig. D.1 for FMNIST dataset using SVM, Fig. D.6 for MNIST dataset using FCN, and Fig. D.10 for FMNIST dataset using FCN.
- Fig. 3.2 from main text is repeated in Fig. D.2 for FMNIST dataset using SVM, Fig. D.7 for MNIST dataset using FCN, and Fig. D.11 for FMNIST dataset using FCN.
- Fig. 3.4 from main text is repeated in Fig. D.3 for FMNIST dataset using SVM, Fig. D.8 for MNIST dataset using FCN, and Fig. D.12 for FMNIST dataset using FCN.
- Fig. 3.5 from main text is repeated in Fig. D.4 for FMNIST dataset using SVM, Fig. D.9 for MNIST dataset using FCN, and Fig. D.13 for FMNIST dataset using FCN.
- Fig. 3.6 from main text is repeated in Fig. D.5 for FMNIST dataset using SVM.

Since FCN has a non-convex loss function, Algorithm 4 is not directly applicable for the experiments in Fig. D.9&D.13. As a result, in these cases, we skip the control steps in line 24-25. We instead use a fixed step size, using a constant ϕ value to calculate the Γ's using (3.31). We are still able to obtain comparable reductions in total cost compared with the federated learning baselines.



(a) 10 labels/node

(b) 3 labels/node

(c) 1 label/node

(d) 10 labels/node

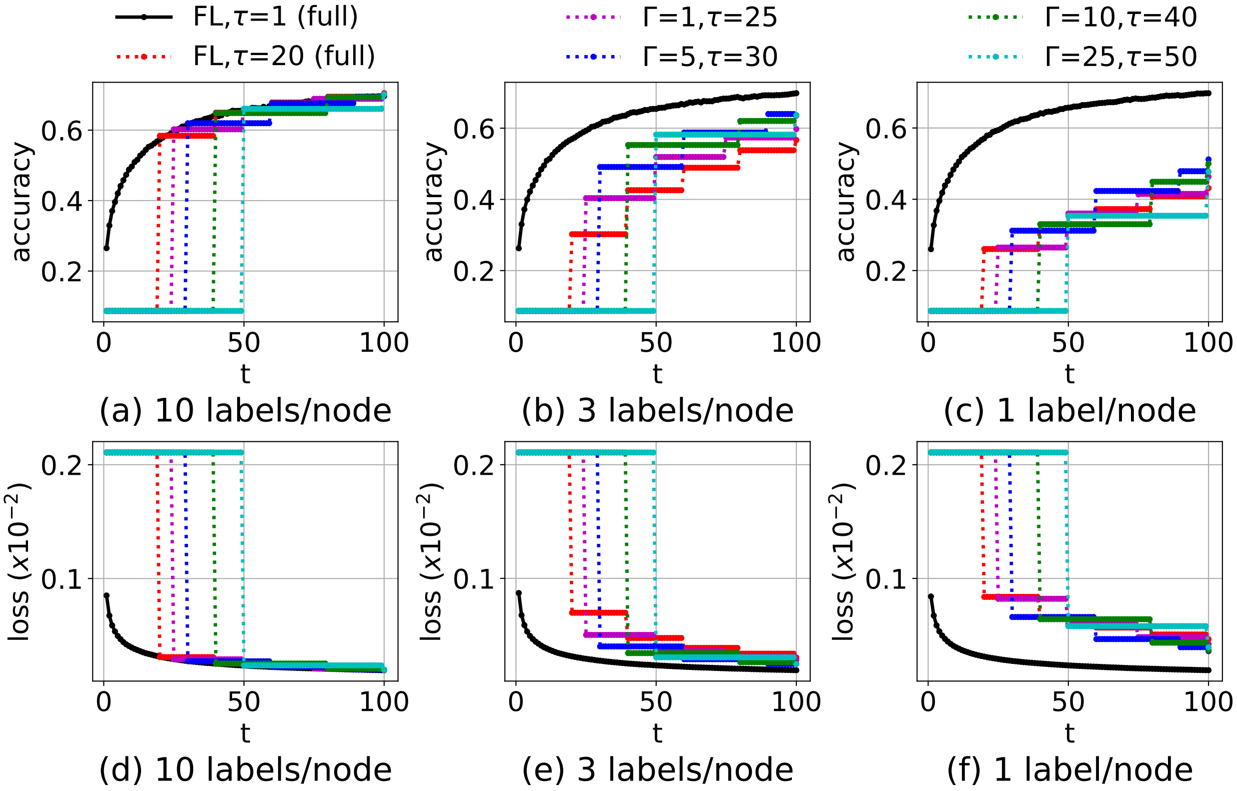
(e) 3 labels/node

(f) 1 label/node

Figure D.1. Performance comparison between `TT-HF` and baseline methods when varying the number of D2D consensus rounds (Γ). Under the same period of local model training (τ), increasing Γ results in a considerable improvement in the model accuracy/loss over time as compared to the current art [24], [116] when data is non-i.i.d. (FMNIST, SVM)

D.2 Extension to Other Federated Learning Methods

Although we develop our algorithm based on federated learning with vanilla SGD local optimizer, our method can benefit other counterparts in literature. In particular, we perform some numerical experiments on FedProx [122] to demonstrate the superiority of our semi-decentralized architecture. The performance improvement is due to the fact that, intuitively, conducting D2D communications via the method proposed by us reduces the local bias of the nodes' models to their local datasets. This benefits the convergence of federated learning methods via counteracting the effect of data heterogeneity across the nodes. The simualtion results are provided in Fig. D.14 and D.15



(a) 10 labels/node

(b) 3 labels/node

(c) 1 label/node

(d) 10 labels/node

(e) 3 labels/node

(f) 1 label/node

Figure D.2. Performance comparison between TT-HF and baseline methods when varying the local model training interval (τ) and the number of D2D consensus rounds (Γ). With a larger τ, TT-HF can still outperform the baseline federated learning [24], [116] if Γ is increased, i.e., local D2D communications can be used to offset the frequency of global aggregations. (FMNIST, SVM)

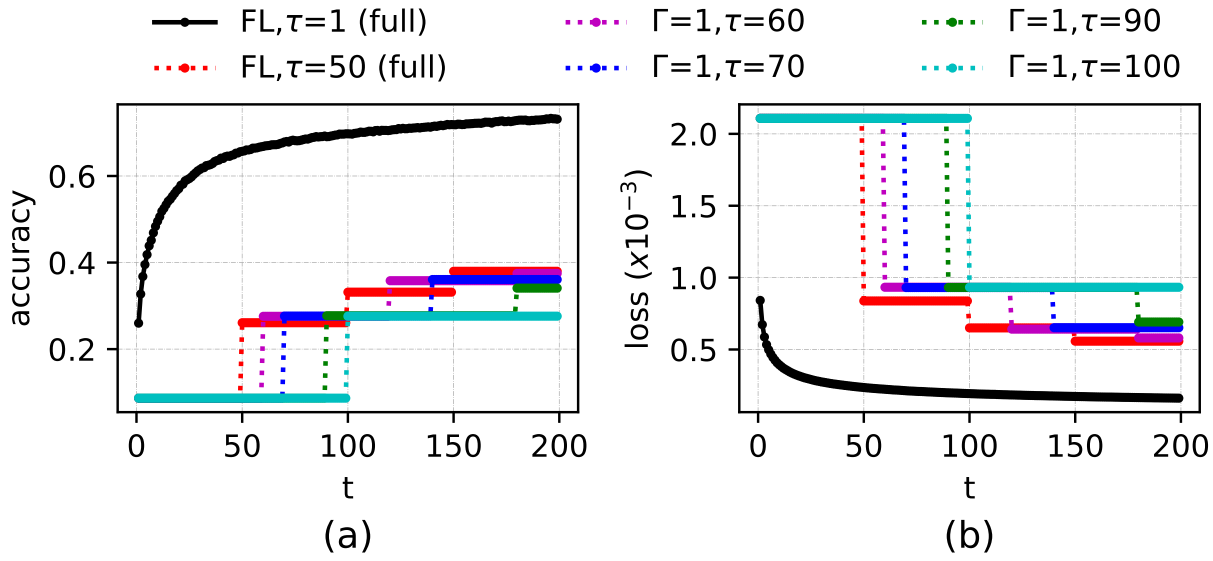


Figure D.3. Performance of TT-HF in the extreme non-i.i.d. case for the setting in Fig. 3.2 when Γ is small and the local model training interval length is increased substantially. TT-HF exhibits poor convergence behavior when τ exceeds a certain value, due to model dispersion. (FMNIST, SVM)

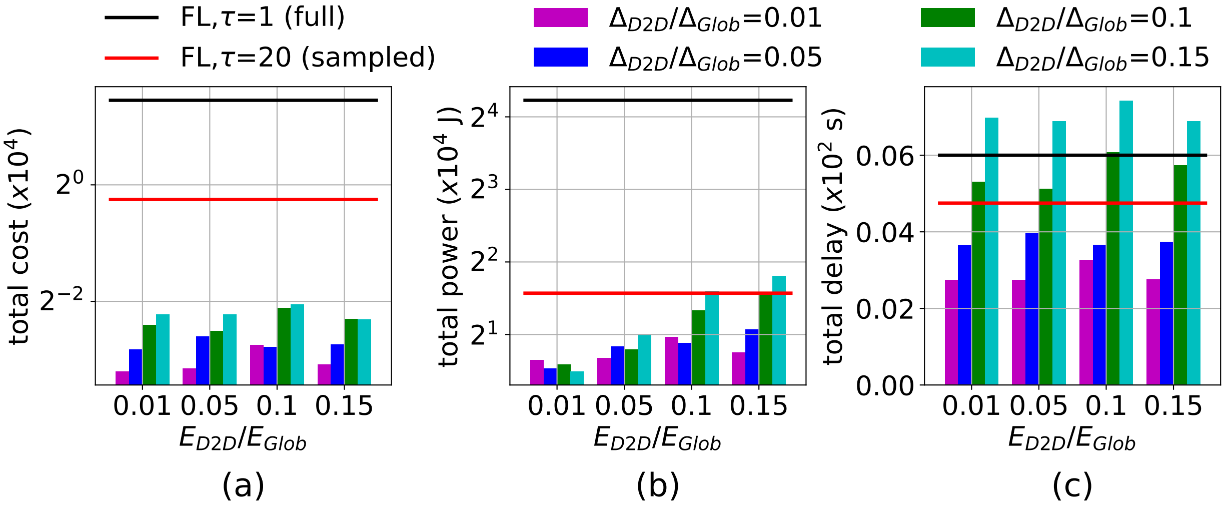


Figure D.4. Comparing total (a) cost, (b) power, and (c) delay metrics from the optimization objective in ($\boldsymbol{\mathcal{P}}$) achieved by TT-HF versus baselines upon reaching 75% of peak accuracy, for different configurations of delay and energy consumption. TT-HF obtains a significantly lower total cost in (a). (b) and (c) demonstrate the region under which TT-HF attains energy savings and delay gains. (FMNIST, SVM)

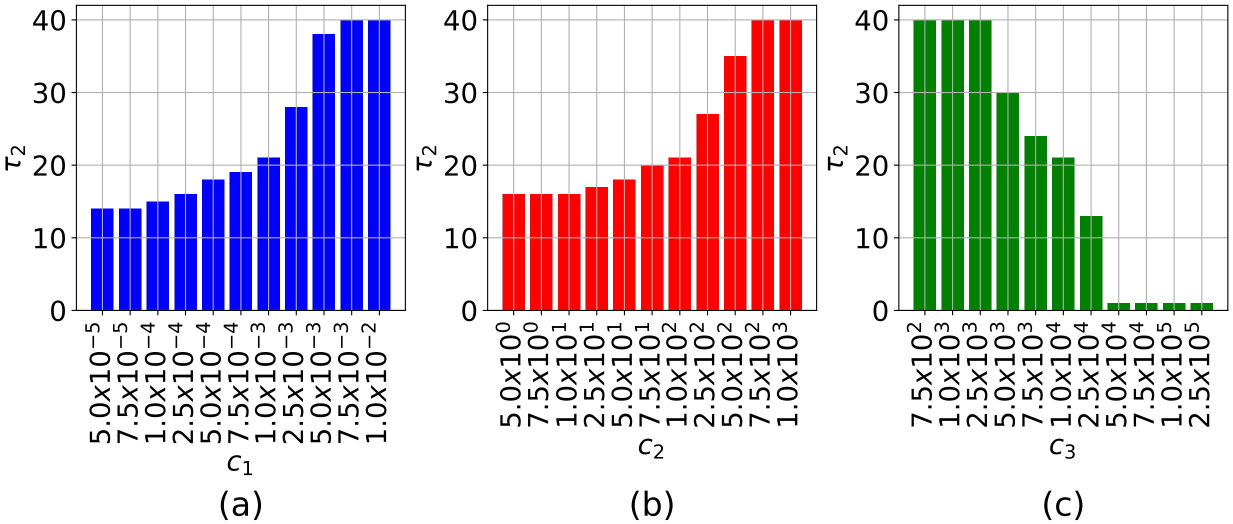
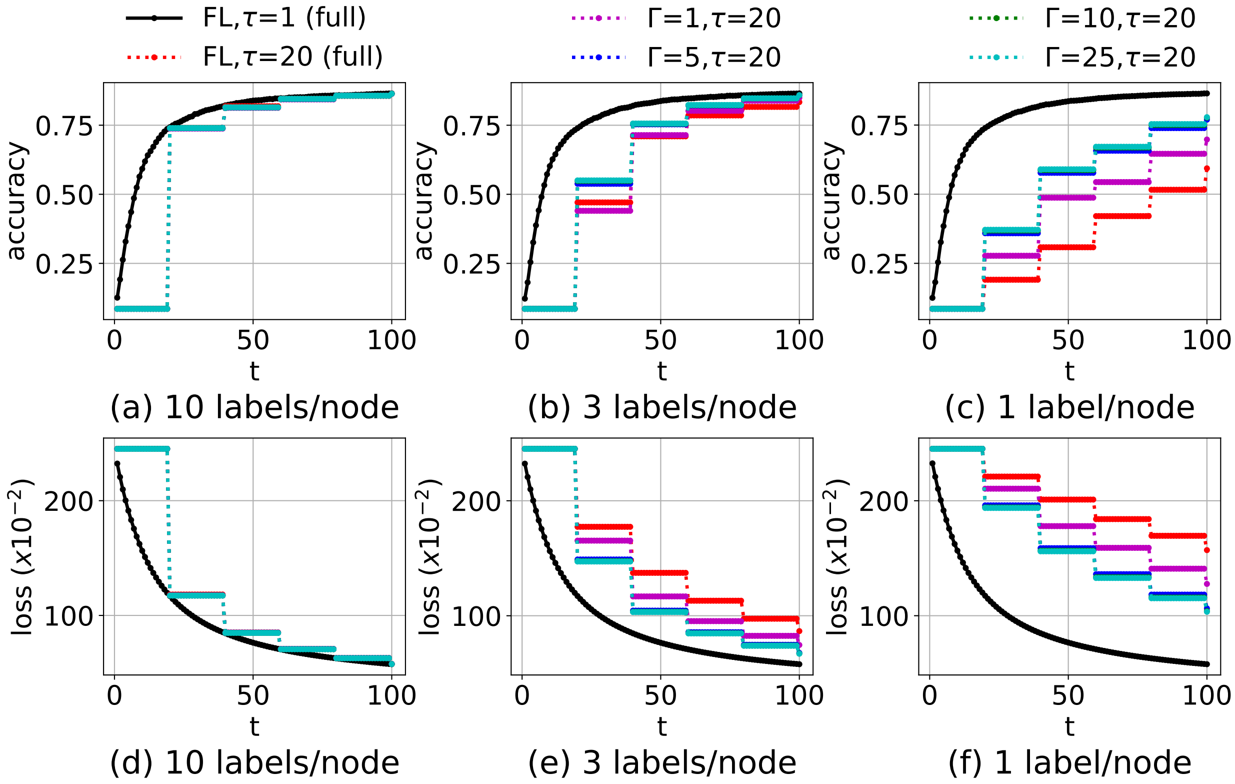


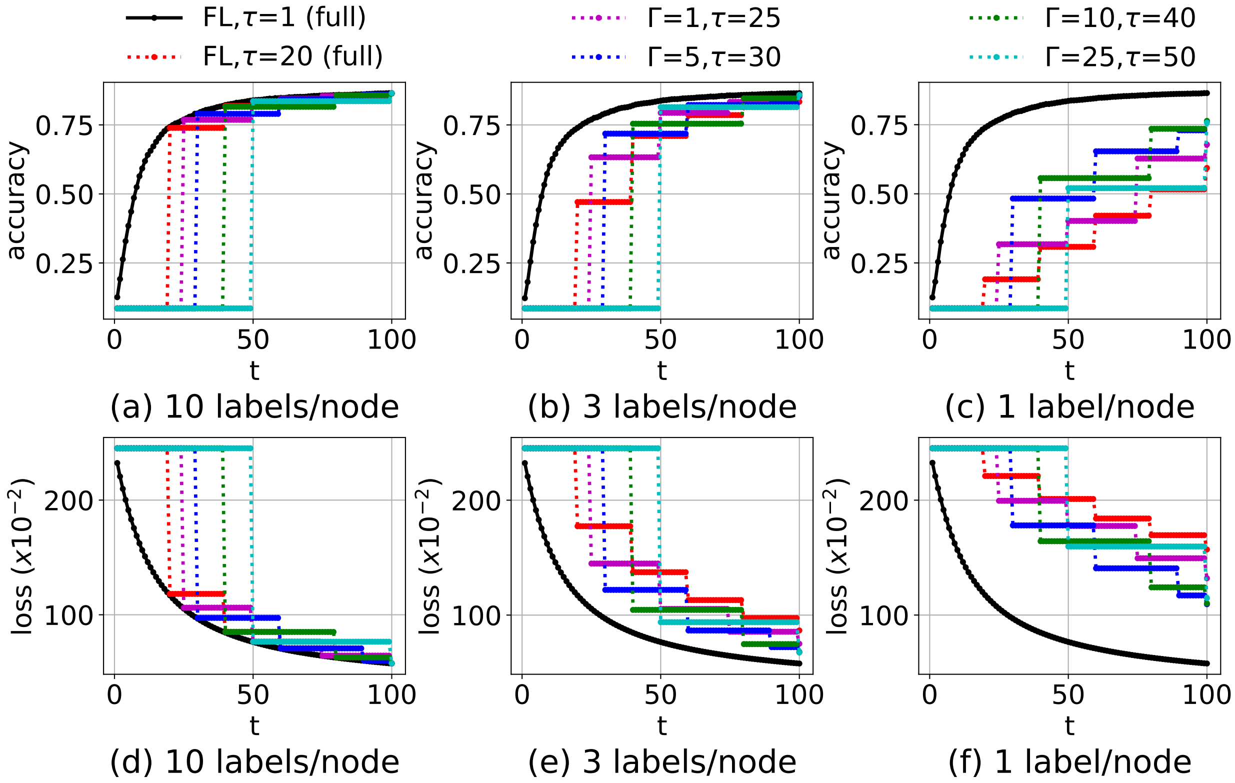
Figure D.5. Value of the second local model training interval obtained through ($\boldsymbol{\mathcal{P}}$) for different configurations of weighing coefficients c_1, c_2, c_3 (default $c_1 = 10^{-3}, c_2 = 10^2, c_3 = 10^4$). Higher weight on energy and delay (larger c_1 and c_2) prolongs the local training period, while higher weight on the global model loss (larger c_3) decreases the length, resulting in more rapid global aggregations. (FMNIST, SVM)



(a) 10 labels/node (b) 3 labels/node (c) 1 label/node

(d) 10 labels/node (e) 3 labels/node (f) 1 label/node

Figure D.6. Performance comparison between `TT-HF` and baseline methods when varying the number of D2D consensus rounds (Γ). Under the same period of local model training (τ), increasing Γ results in a considerable improvement in the model accuracy/loss over time as compared to the current art [24], [116] when data is non-i.i.d. (MNIST, Neural Network)



(a) 10 labels/node (b) 3 labels/node (c) 1 label/node

(d) 10 labels/node (e) 3 labels/node (f) 1 label/node

Figure D.7. Performance comparison between `TT-HF` and baseline methods when varying the local model training interval (τ) and the number of D2D consensus rounds (Γ). With a larger τ, `TT-HF` can still outperform the baseline federated learning [24], [116] if Γ is increased, i.e., local D2D communications can be used to offset the frequency of global aggregations. (MNIST, Neural Network)

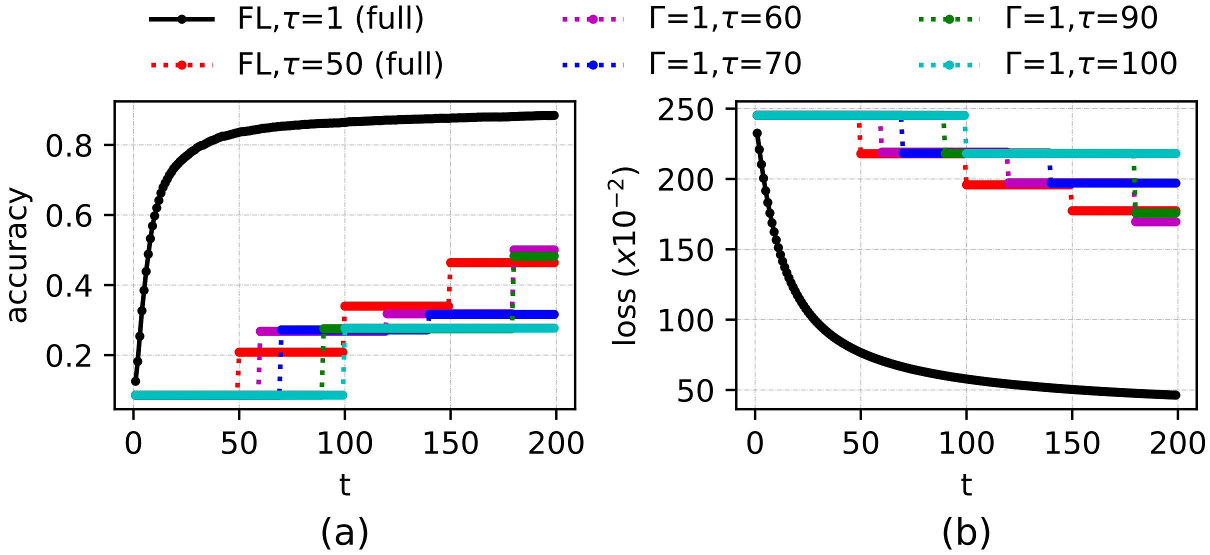


Figure D.8. Performance of TT-HF in the extreme non-i.i.d. case for the setting in Fig. 3.2 when Γ is small and the local model training interval length is increased substantially. TT-HF exhibits poor convergence behavior when τ exceeds a certain value, due to model dispersion. (MNIST, Neural Network)

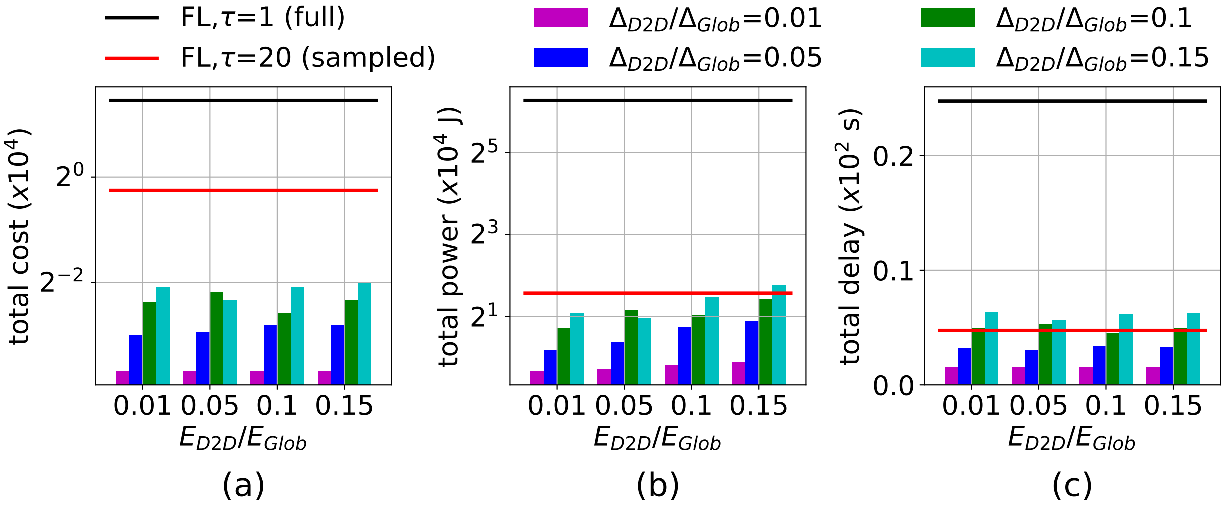
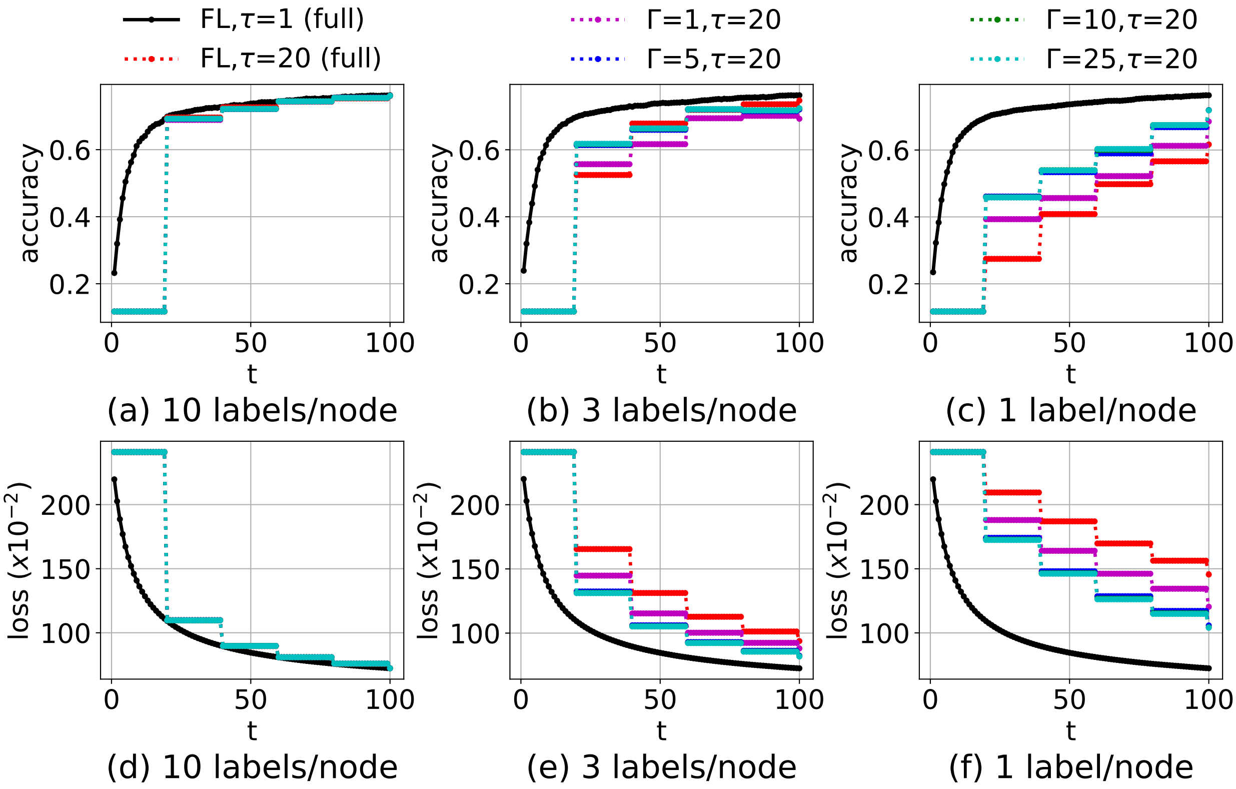


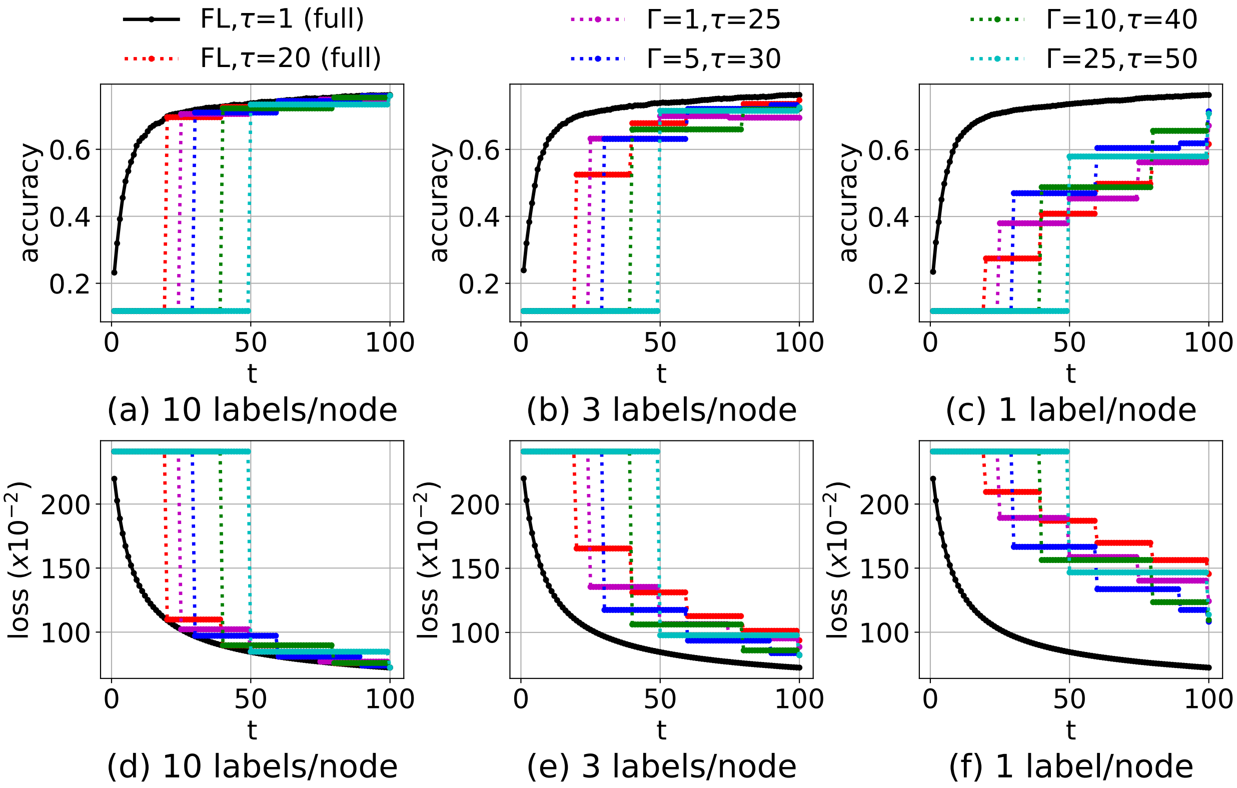
Figure D.9. Comparing total (a) cost, (b) power, and (c) delay metrics from the optimization objective in ($\mathcal{P}$) achieved by TT-HF versus baselines upon reaching 75% of peak accuracy, for different configurations of delay and energy consumption. TT-HF obtains a significantly lower total cost in (a). (b) and (c) demonstrate the region under which TT-HF attains energy savings and delay gains. (FMNIST, SVM)



(a) 10 labels/node (b) 3 labels/node (c) 1 label/node

(d) 10 labels/node (e) 3 labels/node (f) 1 label/node

Figure D.10. Performance comparison between `TT-HF` and baseline methods when varying the number of D2D consensus rounds (Γ). Under the same period of local model training (τ), increasing Γ results in a considerable improvement in the model accuracy/loss over time as compared to the current art [24], [116] when data is non-i.i.d. (FMNIST, Neural Network)



(a) 10 labels/node (b) 3 labels/node (c) 1 label/node

(d) 10 labels/node (e) 3 labels/node (f) 1 label/node

Figure D.11. Performance comparison between `TT-HF` and baseline methods when varying the local model training interval (τ) and the number of D2D consensus rounds (Γ). With a larger τ, `TT-HF` can still outperform the baseline federated learning [24], [116] if Γ is increased, i.e., local D2D communications can be used to offset the frequency of global aggregations. (FMNIST, Neural Network)

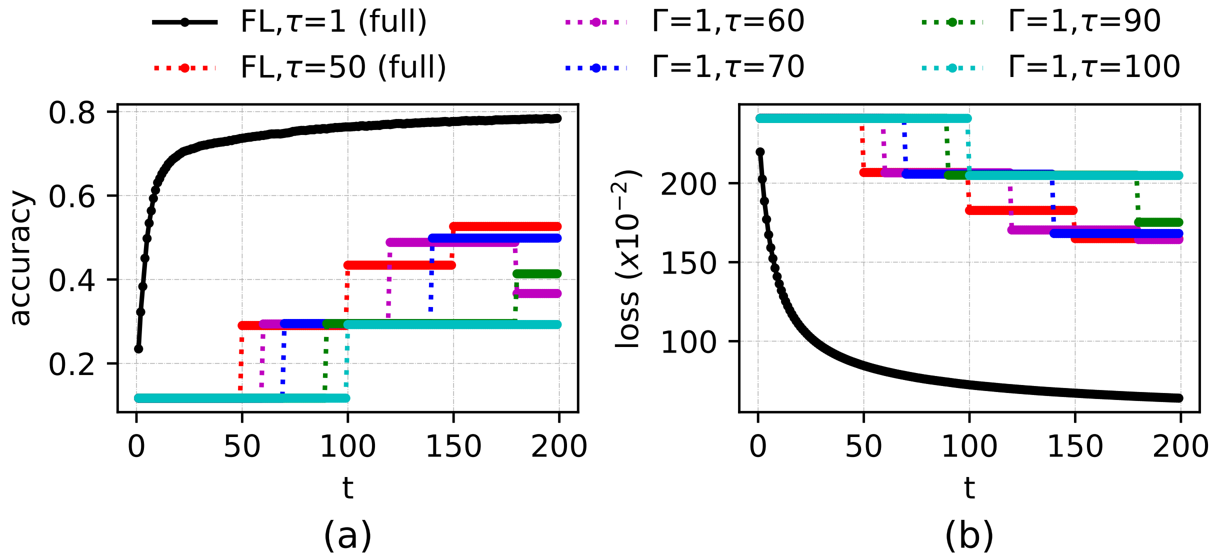


Figure D.12. Performance of `TT-HF` in the extreme non-i.i.d. case for the setting in Fig. 3.2 when Γ is small and the local model training interval length is increased substantially. `TT-HF` exhibits poor convergence behavior when τ exceeds a certain value, due to model dispersion. (FMNIST, Neural Network)

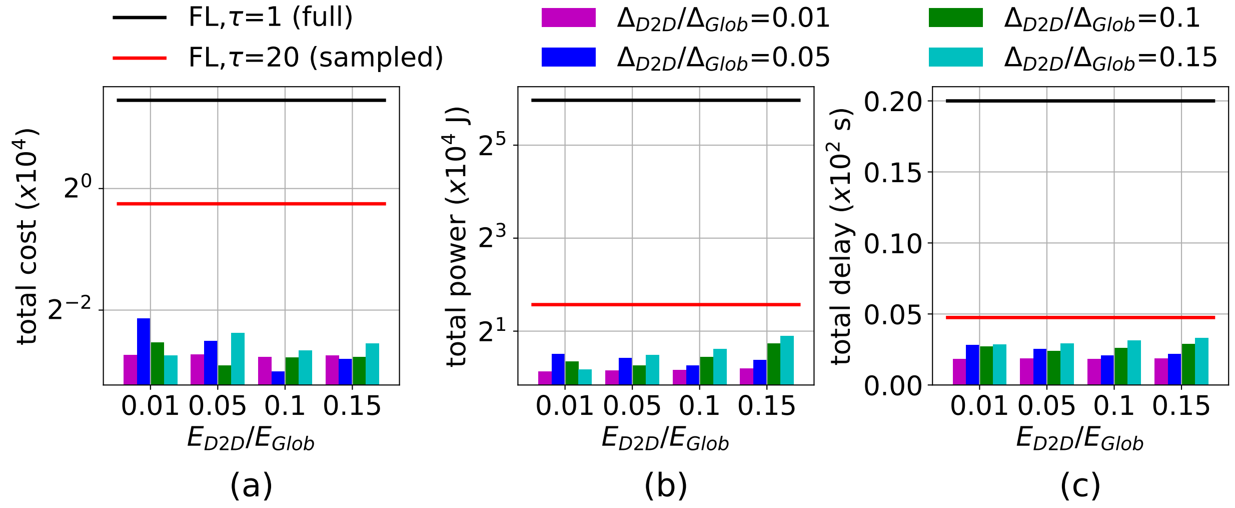
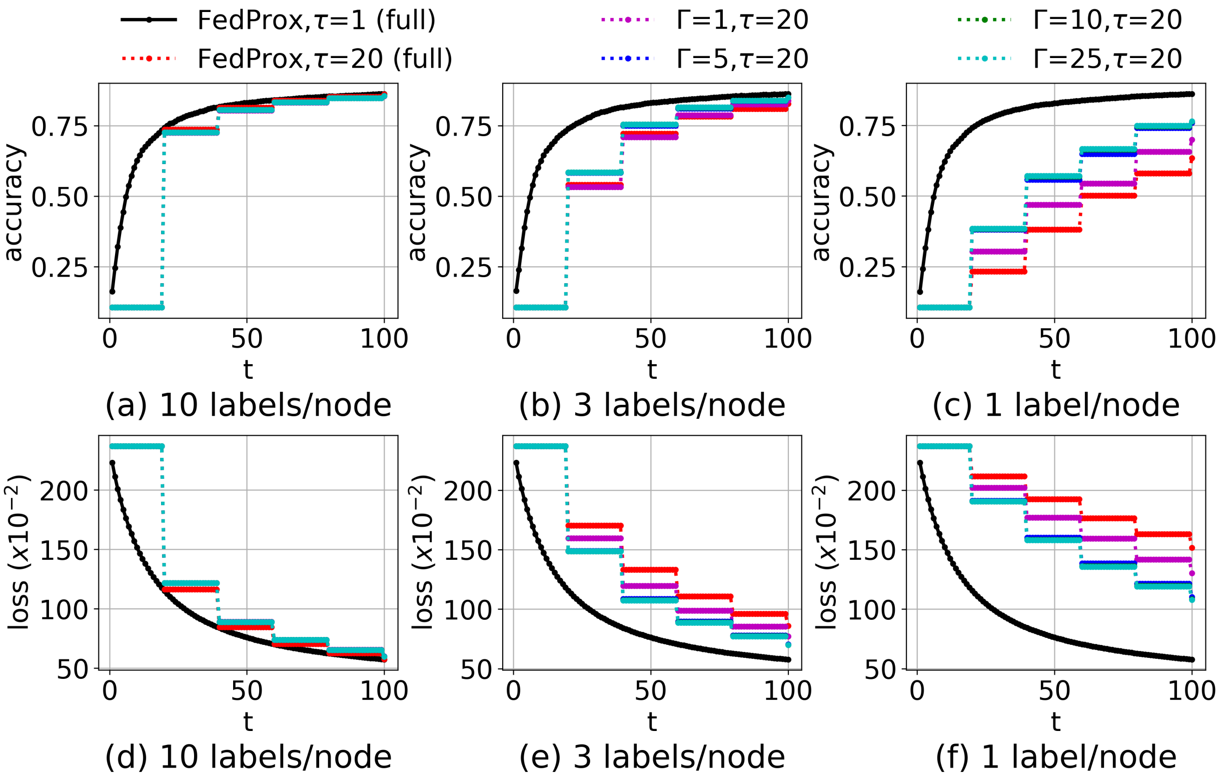


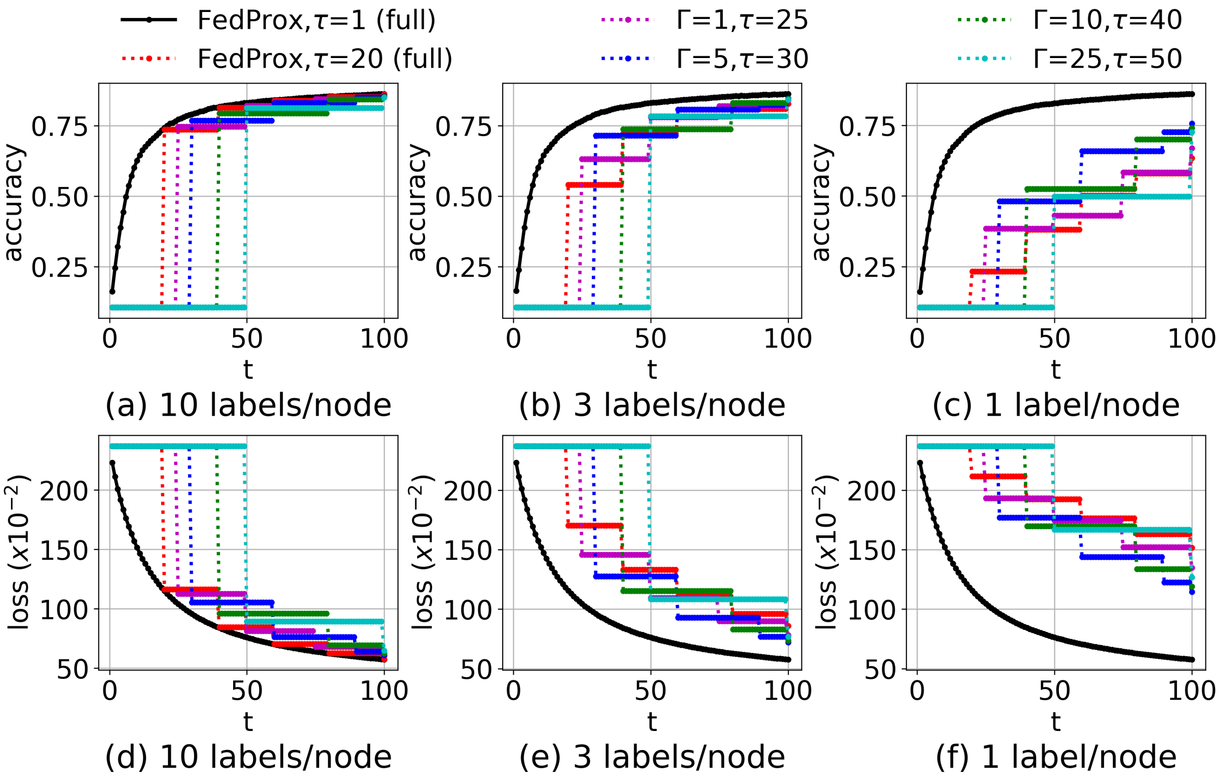
Figure D.13. Comparing total (a) cost, (b) power, and (c) delay metrics from the optimization objective in ($\boldsymbol{\mathcal{P}}$) achieved by `TT-HF` versus baselines upon reaching 75% of peak accuracy, for different configurations of delay and energy consumption. `TT-HF` obtains a significantly lower total cost in (a). (b) and (c) demonstrate the region under which `TT-HF` attains energy savings and delay gains. (FMNIST, Neural Network)



(a) 10 labels/node (b) 3 labels/node (c) 1 label/node

(d) 10 labels/node (e) 3 labels/node (f) 1 label/node

Figure D.14. Performance comparison between TT-HF and FedProx [122] when varying the number of D2D consensus rounds (Γ). Under the same period of local model training (τ), increasing Γ results in a considerable improvement in the model accuracy/loss over time as compared to the baseline when data is non-i.i.d. (MNIST, Neural Network)



(a) 10 labels/node (b) 3 labels/node (c) 1 label/node

(d) 10 labels/node (e) 3 labels/node (f) 1 label/node

Figure D.15. Performance comparison between TT-HF and FedProx [122] when varying the local model training interval (τ) and the number of D2D consensus rounds (Γ). With a larger τ, TT-HF can still outperform the baseline method if Γ is increased, i.e., local D2D communications can be used to offset the frequency of global aggregations. (MNIST, Neural Network)