SORBONNE UNIVERSITÉ

EURECOM
Sophia Antipolis

PhD Thesis

In partial fulfillment of the requirements for the degree of doctor of philosophy from Sorbonne University

Communication-efficient Decentralized Learning for Intelligent Networked Systems

Eunjeong JEONG

Scheduled for defense on September 27, 2024, before a committee composed of:

Reviewers:		
Prof.	Carlo FISCHIONE	KTH, Sweden
Prof.	Jemin LEE	Yonsei University, South Korea
Examiners:		
Prof.	Zheng CHEN	Linköping University, Sweden
Prof.	Giovanni NEGLIA	INRIA, France
Prof.	Navid NIKAEIN	EURECOM, France *(President of the Jury)*
Thesis Advisor:		
Prof.	Marios KOUNTOURIS	EURECOM, France

Thèse

Présentée pour obtenir le Grade de Docteur de Sorbonne Université

Apprentissage Décentralisé Efficace en Matière de Communication pour les Systèmes en Réseaux Intelligents

Eunjeong JEONG

Soutenance de thèse planifiée au 27 Septembre 2024, devant le jury composé de :

Rapporteurs :			
	Prof.	Carlo FISCHIONE	KTH, Suède
	Prof.	Jemin LEE	Yonsei University, Corée du Sud
Examinateurs :			
	Prof.	Zheng CHEN	Linköping University, Suède
	Prof.	Giovanni NEGLIA	INRIA, France
	Prof.	Navid NIKAEIN	EURECOM, France *(Président du Jury)*
Directeur de Thèse :			
	Prof.	Marios KOUNTOURIS	EURECOM, France

Acknowledgments

It is such an amazing thing to get a Ph.D. for doing something that I enjoy. And certainly, I would not have made it without the help of many people.

First and foremost, I sincerely send my gratitude to my thesis supervisor Marios Kountouris. I always appreciate his wide and deep insight into the newest and the most fundamental academic trends. I learned how to set my own problems, how to delve into these problems as an individual researcher, and how to clearly deliver what I have done or what I have to do in the scientific language.

I greatly thank Prof. Carlo Fischione and Prof. Jemin Lee for reviewing my thesis. I also highly appreciate Prof. Zheng Chen, Dr. Giovanni Neglia, and Prof. Navid Nikaein for participating in my defense as examiners. Thank you very much for sparing your valuable time for an important moment of my milestone.

I would like to acknowledge Huawei France for supplying support during my doctoral program.

From the bottom of my heart, I thank EURECOM friends and colleagues for being humorous, warmhearted, silly, and intellectual. I am also grateful to my good old friends in Korea for being supportive, chaotic, and joyful.

I certainly thank EURECOM staff for helping me with the most annoying and trickiest administrative procedures. I appreciate their punctuality and kindness.

Most importantly, I would like to express my deepest gratitude to my parents and my brother for supporting such a bizarre family member like me who had decided to take a long-term intercontinental travel and study further to grab another degree. Also, I thank to my grandmother for sending positive energies and cheering to me since I was very young. She would be proud of her first grandchild bringing a fancy degree.

Needless to say that I thank my dog, Sonic, who has been my best friend in my journey as a doctoral student. Sometimes he was a bad boy tearing my paper draft or smacking the keyboard to type gibberish on my Overleaf projects. Nonetheless, he has been lovely enough to make the world a lot better place to bear.

Last but not least, I thank my potential readers who, somehow, decided to turn another page of this thesis.

CONTENTS

List of Figures

List of Algorithms

List of Tables

Nomenclature

Acronyms and Abbreviations

AI	Artificial Intelligence
AirComp	Over-the-air Computation
AWGN	Additive White Gaussian Noises
CD	Co-Distillation
CNN	Convolutional Neural Network
DSGD	Decentralized Stochastic Gradient Descent
FedAvg	Federated Averaging
FL	Federated Learning
IoT	Internet of Things
KD	Knowledge Distillation
non-i.i.d.	non-independent and identically distributed
PPP	Poisson Point Process
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
SINR	Signal-to-Interference-plus-Noise Ratio
TCP	Transmission Control Protocol

Notations

a	Power scaling factor
B	Number of training batch per user
b	Index of training batch
c	Confidence coefficient
$d(i,j)$	Physical distance between device i and j
$d_w(i,j)$	Dissimilarity (statistical distance) between model i and model j
$\mathcal{D}_i$	Local training set
$\mathcal{E}$	A set of edges
$\mathbb{E}[\cdot]$	Expectation
f_i	Local loss function
$g_i(\theta)$	Local gradient oracle (in Chapter 2 and Chapter 3)
$g_i(w)$	Regularization term (in Chapter 4)
G^2	Bound for expected gradient magnitude
$\mathcal{G}$	Connectivity graph (in Chapter 2)
$h_{i,j}$	Channel coefficient (in Chapter 2) · Fading gain (in Chapter 3)
i	Index of a random user without losing generality
j	Index of a random neighbor without losing generality
$J(\cdot)$	Joint objective function (in Chapter 4)
K	Number of all events throughout the whole learning procedure
k	Index of an event
L	Lipschitz constant
N	Number of clients
N_0	Noise power density
$\mathcal{N}_i$, $\mathcal{N}(i)$	Set of neighbors of user i
n_θ	Dimension of local models
$n_\mathcal{X}$	Dimension of input features
$\mathcal{Q}$	Set of all transmission indicators, i.e., $\{q_t^{i\to j} \mid \forall i, j \in \mathcal{U}, \forall t \in [0,T)\}$
P	Unification period
P_i	Local transmission power
$\mathrm{P}[\cdot]$	Probability distribution
q	Transmission incidence indicator (in Chapter 3)
R	Number of total global communication rounds
r	Index of global round (in Chapter 2 and Chapter 4) Cell radius (in Chapter 3)

S_r	Number of slots at round r
s	Index of transmission slot
T	Total learning time
T_{ex}	Exchanging interval
t	Index of a time point
$\mathcal{U}$	Set of network users
W	Mixing matrix (in Chapter 2) · Collaboration graph (in Chapter 4) Bandwidth (in Chapter 3)
$w_{i,j}^{(r)}$	(i,j) entry of the mixing matrix $W^{(r)}$
w_i	Connectivity vector of user i (in Chapter 4)
y	Transmitted message
$\tilde{y}$	Received message
$\hat{y}$	Estimated message from the recipient
z	Noise vector (in Chapter 2)
$\mathbf{z}_i$	Logits (in Chapter 4)
α	Path loss exponent
$\beta^{(r)}$	Learning rate for renewing $w_{ij}^{(r)}$
$\Gamma_{i\rightarrow j}$	Transmission delay for a message from i to j
$\Gamma_{\max}$	Transmission delay deadline
γ	Power alignment coefficient
Δ_i	Local update
ζ	Bound for gradient divergence
$\eta,\ \eta_i$	Local learning rate
Θ	Set of local models, i.e., $\{\theta_1, \cdots, \theta_N\}$
θ	Global model
$\bar{\theta}$	Average model, i.e., $\frac{1}{N}\sum_{i=1}^{N}\theta_i$
θ_i	Local model
$\lambda,\ \lambda_i$	Exponential rate parameter (mean of local computation time)
$\mu_1,\ \mu_2$	Hyperparameters (different across chapters; refer to each chapter for definitions.)
ν_i	Probability of user i being a straggler
ξ	Step size while updating local models with stale gradients
ρ	Bound for sum of squared $q_t^{i\rightarrow j}$'s
σ^2	Bound for variance of gradients
τ	Transmission delay
Ψ	Maximum number of packets that each user allows to receive during P
$\psi_i(t_a, t_b)$	Number of packets arrived during time $[t_a, t_b)$

ABSTRACT

The burgeoning Internet of Things (IoT) and the rise of edge computing demand scalable and robust decentralized learning systems that prioritize privacy, adapt to dynamic environments, and accommodate diverse user requirements. This thesis advances the field of distributed machine learning by unifying key advancements in decentralized collaborative learning into a comprehensive framework that addresses communication challenges, computation variability, and personalization in networked systems.

A core objective of this research is to design a versatile decentralized learning framework that operates efficiently in environments characterized by unreliable communication, heterogeneous devices, and evolving user needs. To achieve this, we propose an asynchronous learning paradigm that decouples communication and computation timelines. This decoupling enables autonomous operation and reduces reliance on rigid synchronization protocols, thereby mitigating the impact of communication delays and straggler problems. Rigorous theoretical analysis and extensive experimental evaluations demonstrate the convergence and robustness of this asynchronous approach.

Furthermore, recognizing that uniform global models may not suffice in heterogeneous settings, the framework integrates strategies for personalization. By leveraging knowledge distillation techniques to quantify the statistical dissimilarity between local models, the framework fosters meaningful collaboration between users with similar data distributions. This enables the development of tailored models for individual participants, ensuring that the framework addresses the unique requirements of each user while maintaining the benefits of collective learning.

By combining robust asynchronous communication, dynamic adaptation to computation and network variability, and user-centric personalization, this thesis presents a unified approach to decentralized learning. The proposed framework sets the stage for a new generation of intelligent networked systems that are not only communication-efficient but also scalable, adaptive, and user-focused. These contributions aim to redefine the potential of decentralized learning, bridging critical gaps and enabling broader applications in diverse domains such as smart cities, healthcare, and autonomous systems.

RÉSUMÉ

L’essor de l’Internet des Objets (IoT) et le développement du calcul en périphérie nécessitent des systèmes d’apprentissage décentralisés évolutifs et robustes, qui mettent l’accent sur la confidentialité, s’adaptent à des environnements dynamiques et répondent aux besoins variés des utilisateurs. Cette thèse fait progresser le domaine de l’apprentissage automatique distribué en unifiant des avancées clés en apprentissage collaboratif décentralisé dans un cadre complet traitant des défis de communication, des variations de calcul et de la personnalisation dans les systèmes en réseau.

L’objectif principal de cette recherche est de concevoir un cadre d’apprentissage décentralisé polyvalent qui fonctionne efficacement dans des environnements caractérisés par des communications peu fiables, des dispositifs hétérogènes et des besoins utilisateurs évolutifs. Pour y parvenir, nous proposons un paradigme d’apprentissage asynchrone qui découple les chronologies de communication et de calcul. Ce découplage permet une opération autonome et réduit la dépendance aux protocoles de synchronisation rigides, atténuant ainsi l’impact des retards de communication et des problèmes de "traînards". Des analyses théoriques rigoureuses et des évaluations expérimentales approfondies démontrent la convergence et la robustesse de cette approche.

De plus, reconnaissant que des modèles globaux uniformes ne suffisent pas dans des environnements hétérogènes, le cadre intègre des stratégies de personnalisation. En exploitant des techniques de distillation des connaissances pour quantifier la dissimilarité statistique entre les modèles locaux, le cadre favorise une collaboration significative entre les utilisateurs ayant des distributions de données similaires. Cela permet de développer des modèles adaptés aux besoins individuels tout en préservant les avantages de l’apprentissage collectif.

En combinant une communication asynchrone robuste, une adaptation dynamique aux variations de calcul et de réseau, et une personnalisation centrée sur l’utilisateur, cette thèse propose une approche unifiée de l’apprentissage décentralisé. Le cadre ouvre la voie à une nouvelle génération de systèmes intelligents en réseau, efficaces en termes de communication, évolutifs, adaptatifs et centrés sur l’utilisateur. Ces contributions visent à redéfinir le potentiel de l’apprentissage décentralisé, comblant des lacunes cruciales et permettant des applications plus larges dans divers domaines tels que les villes intelligentes, la santé et les systèmes autonomes.

1 Introduction

The rapid growth of the Internet of Things (IoT) and edge computing has opened up new possibilities for decentralized collaborative learning. This evolution allows devices to optimize local neural networks through direct peer-to-peer communications. Shifting towards fully decentralized learning removes the need for centralized servers, offering better privacy, reduced delays, and increased resilience against network failures. However, achieving efficient and reliable decentralized learning in wireless networks faces significant challenges, including changing network structures, unreliable communication links, and varied computational abilities of devices. Addressing these challenges is crucial to realizing the full potential of decentralized learning.

Fully decentralized federated learning is not just a solution, but an essential one for tackling the pressing data privacy and security issues of our time. It ensures that raw data stays on local devices, a crucial aspect in fields where data sensitivity is critical, such as healthcare, finance, and personal devices. Its popularity also comes from its ability to use the computing power of many edge devices, reducing dependence on centralized systems and improving scalability. Nevertheless, current challenges include managing changing network structures, which can cause intermittent connectivity and unreliable communication. Additionally, the varied computational power of different devices makes it harder to coordinate efficient learning processes across the network. These limitations call for innovative solutions to make decentralized learning robust and effective, and to underline the urgency of the issue.

In this thesis, we not only address the challenges of decentralized learning but also propose innovative solutions that could potentially transform the field. We focus on asynchronous decentralized learning algorithms, which operate independently and adapt to the inherent variability of wireless networks. Our main contribution is the development of an asynchronous Decentralized Stochastic Gradient Descent (DSGD) algorithm that maintains performance despite communication and computation issues. We provide a thorough theoretical analysis and non-asymptotic convergence guarantees for our method, supported by extensive experimental evaluations. Additionally, we pioneer the personalization of decentralized networks with a personalized DSGD algorithm that uses knowledge distillation to measure and quantify statistical differences between models. These innovative solutions hold the potential for exciting breakthroughs in decentralized learning.

Our research pivots on three key aspects of decentralized collaborative learning: asynchronous communication, efficient algorithm design for continuous learning over row-stochastic networks, and personalized learning. By addressing these critical areas, we aim to push the boundaries of decentralized learning, making it a viable and effective solution for future intelligent networks.

1.1 Background

Before discussing the main contributions of the thesis, we hereby introduce the fundamental concepts such as Federated Learning (FL), decentralized (serverless) learning, and communication efficiency.

1.1.1 Federated Learning

Federated Learning (FL) [1] is a distributed machine learning paradigm where multiple devices, or clients, collaboratively train a shared global model while keeping their local data private. This approach emerged to address data privacy concerns, enabling each client to perform computations on its data locally and only share model updates with a central server. The server then aggregates these updates to refine the global model. This iterative process continues until the model converges.

Let $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \cdots \mathcal{D}_N$ be a joint training dataset and $|\mathcal{D}| = \sum_{i=1}^{N} |\mathcal{D}_i|$ the total number of training data samples. The goal of FL is to solve problems of the form

$$\min_{\theta \in \mathbb{R}^{n}\mathcal{X}} f(\theta;\mathcal{D})$$

where

$$f(\theta;\mathcal{D}) = \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \sum_{i=1}^{N} f_i(\theta;\mathcal{D}_i) \quad \text{and} \quad f_i(\theta;\mathcal{D}_i) = \sum_{x \in \mathcal{D}_i} f(\theta, x).$$

Federated Learning operates through several key stages:

1. Initialization: The central server initializes and distributes the global model to all participating clients.

2. Local Training: Each client trains the model on its local dataset, performing a specified number of iterations or epochs. This stage allows the model to learn from diverse and

potentially non-independent and identically distributed (non-i.i.d.) data distributions on different devices.

3. Model Update: After local training, each client computes updates, typically in the form of gradients or model weights, and sends these updates back to the central server. Importantly, the raw data remains on the clients, preserving privacy. received from the clients to improve the global model. Common aggregation methods include averaging the weights or gradients. This aggregated model is then redistributed to the clients for further local training.

4. Iteration: The local training and model aggregation process is repeated iteratively. The global model is continuously refined as it learns from the data across all clients. This iterative process continues until the model converges to a satisfactory level of performance.

Key concepts of FL include:

- Local Training: Each client independently trains the model on its own data, ensuring that sensitive information does not leave the device. This decentralized approach leverages the computational power and data availability of edge devices.

- Model Aggregation: The central server plays a crucial role in combining the updates from all clients. By aggregating these updates, the server creates a more accurate and generalized global model that benefits from the diverse data held by the clients.

- Privacy Preservation: A primary advantage of FL is that it significantly reduces the risk of data breaches. Since the raw data remains on the local devices and only model updates are communicated, the privacy of the individuals' data is maintained.

- Scalability: FL is highly scalable and can accommodate a large number of clients, therefore suitable for applications involving extensive networks of devices, such as IoT ecosystems, where data is generated at the edge.

- Communication Efficiency: FL aims to minimize communication overhead by reducing the frequency and size of the model updates exchanged between clients and servers. Techniques such as model compression, quantization, and selective update sharing are often employed to enhance communication efficiency.

- Heterogeneous Data: FL is designed to handle heterogeneous data distributions across clients. Unlike centralized learning, where data is assumed to be i.i.d., FL recognizes that data on different devices may vary significantly and adapts accordingly.

Algorithm 1: Federated Averaging (local SGD)

```
Input: $\theta^{(0)}$
for each round $r = 1, \cdots, R$ do
    $\mathcal{N}^{(r)} \leftarrow$ random set of users
    for $i \in \mathcal{N}^{(r)}$ do
        for local step $b = 1, \cdots, B$ do
            $\mathcal{B}_i \leftarrow$ mini-batch of $B$ steps
            $\theta_i \leftarrow \theta_i - \frac{|\mathcal{D}_i|}{B} \eta \sum_{x_i \in \mathcal{B}_i} \nabla f(\theta; x_i)$
        end for
    end for
    $\theta \leftarrow \sum_{i \in \mathcal{N}^{(r)}} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \theta_i$
end for
return $\theta^{(R)}$
```

FL is an evolving field with ongoing research focusing on improving its efficiency, robustness, and applicability. Current research areas include developing more sophisticated aggregation algorithms, enhancing security and privacy through advanced cryptographic techniques, and optimizing communication protocols to reduce overhead further.

Federated Averaging

Federated Averaging (FedAvg) [2] is one of the first approaches to implement FL that enables collaborative training of machine learning models across multiple decentralized devices while keeping data private. The process begins with each client training a local model using its own dataset and calculating updated model parameters, θ_i. These local parameters are then sent to a central server. The server aggregates these parameters by computing a weighted average based on the number of data samples, at each client, resulting in the global model update: $\theta = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \theta_i$. This global model is then redistributed to the clients for further local training, and the process repeats. (See Alg. 1) This method has been the most commonly used algorithm for training models in a federated manner since it ensures that the model benefits from diverse data across clients while preserving data privacy.

1.1.2 Decentralized (Serverless) Learning

Among federated learning schemes, we particularly delve into decentralized (serverless) learning, a distributed machine learning approach where devices collaboratively train models without relying on a central server. Learning without a central server is actively studied due to the difficulty of getting a complete entity that is always stable, trustworthy, and fair. [3]–[6] This paradigm leverages peer-to-peer communication and consensus algorithms to achieve model updates, enhancing robustness and scalability. [7]

In decentralized learning, devices communicate directly, eliminating the need for a central server and reducing single points of failure. This direct communication propagates information through various network topologies, such as fully connected, ring, or mesh networks.

Consensus algorithms, such as DSGD and the gossip protocol, play a crucial role in helping nodes converge to a common model. These algorithms iteratively update the model based on local computations and updates from neighboring nodes, ensuring global consistency. [8], [9]

Decentralized learning is inherently fault-tolerant, as the failure of individual nodes does not incapacitate the entire system. If a node fails or becomes temporarily unavailable, its neighbors can continue to exchange updates with other nodes, dynamically adapting to the failure and maintaining overall functionality.

The approach scales naturally with the number of participating nodes, making it suitable for large-scale deployments like IoT networks. [10] As more nodes join the network, they integrate into the existing peer-to-peer communication structure, contributing to and benefiting from the collective learning process without overloading a central server. [11]

Enhanced privacy and security are significant advantages of decentralized learning. Since there is no central repository of data or model updates, each node retains control over its local data. Nodes share only the necessary information for model updates, often using encrypted communications and differential privacy techniques to protect sensitive data further.

Decentralized learning can also accommodate dynamic changes in the network topology, such as nodes joining or leaving the network or varying communication links. Consensus algorithms and peer-to-peer protocols are designed to handle these changes seamlessly, ensuring the learning process continues smoothly despite network dynamics.

In summary, decentralized (serverless) learning offers a robust, scalable, and flexible approach to collaborative model training. By leveraging peer-to-peer communication [12], [13], consensus algorithms [14]–[16], and asynchronous updates [17]–[20], it overcomes many limitations of centralized systems, making it particularly well-suited for complex, large-scale, and dynamic environments like IoT networks.

1.1.3 Communication Efficiency in On-device Learning

In the literature, several efforts have been made to enhance communication efficiency in FL, collectively addressing the solution for communication bottlenecks by reducing the amount of data transmitted, optimizing the timing and frequency of communications, and leveraging novel communication techniques.

One crucial factor in achieving communication efficiency is the age of information (AoI) [21]. AoI refers to the time elapsed since a piece of information was generated. In FL, minimizing communication overhead ensures that models across devices are trained using the most recent information. This reduces the negative impact of outdated information (high AoI) on model convergence and accuracy. [22], [23]

Numerous studies have focused on minimizing the amount of data exchanged between devices to reduce latency and bandwidth usage. Techniques such as gradient compression [24], sparsification [25]–[27], and quantization [28]–[30] are used to reduce communication load. These methods are crucial in wireless networks where communication resources are limited.

Most approaches aim to reduce the number of communications or decrease the message size. To minimize communication frequency, one-shot [31]–[33] or few-shot FL [34]–[36] methods have been proposed. Beyond reduction of the number of communication rounds, optimizing the iteration cost and communication round duration are also considered. [37], [38] Alternatively, the data packet size can be downsized by lowering the precision of model parameters, thus decreasing the amount of exchanged information. Techniques such as quantization [39], sparsification [40]–[43], and pruning [44]–[46] are integrated with collaborative learning to achieve this. While compression can result in information loss or distortion, leading to a tradeoff between encoding ratio and compression error, it can significantly improve overall communication complexity [47].

On the other hand, FL has attempted conjugating with problems in traditional wireless systems, such as collision avoidance, bandwidth constraints, and latency. Collision avoidance matters as multiple devices simultaneously communicating their model updates can lead to significant collisions and data loss. [48] On the other hand, wireless networks often suffer from limited bandwidth. In FL, transmitting extensive model updates or gradients can consume significant bandwidth, leading to congestion and reduced network performance. [49], [50] Furthermore, wireless networks can have variable latency, which affects the synchronization of model updates in FL. Ensuring timely updates or delay tolerance is challenging due to the varying latency in wireless communications. [51]

1.2 Justification and Research Questions

This thesis delves into a series of novel and critical research questions in the realm of decentralized collaborative learning over wireless networks:

- *How can global consensus be achieved despite communication and computation impairments?* This question is essential for ensuring that decentralized learning systems can function effectively even when individual devices experience delays or failures in communication and computation.

- *How can decentralized users follow simple, independent instructions, and how can the procedure be continuous, allowing transmission and reception at any moment on the timeline to reduce synchronization costs?* This question focuses on making decentralized learning more practical and efficient by reducing the overhead associated with synchronization and allowing for more flexible communication protocols.
- *How can we measure stochastic dissimilarity between local models without exchanging raw data to customize model training effectively?* This question is crucial for personalizing model training in decentralized systems while preserving data privacy, as it seeks methods to tailor models to local data distributions without sharing sensitive data.

The overarching goal throughout this thesis is to develop robust solutions that address the inherent limitations of edge devices, enabling them to train collaboratively in a federated manner. These limitations include heterogeneous time consumption for data transmission and local model training (covered in Chapters 2 and 3), high synchronization costs (Chapter 3), and highly non-i.i.d. local data distributions (Chapter 4).

The common theme across each chapter is FL, where participants actively collaborate and exchange information to train a global model without directly sharing raw data. Each agent aims to train a model to a level that would be unattainable for a single user alone.

Additionally, this thesis explores decentralized user networks, where the roles of gathering information, broadcasting model updates, and renewing the model from collected data are not fixed to a specific entity. Instead, different participants can perform these tasks independently, promoting flexibility and resilience in the network.

The practical implications of this research are far-reaching. This work paves the way for more robust and efficient IoT applications by addressing the challenges of decentralized collaborative learning. The proposed solutions can be applied in various real-world scenarios, such as smart cities, autonomous vehicles, and personalized healthcare, offering significant benefits to end-users and service providers. This research not only advances decentralized learning but also contributes to the broader vision of intelligent, interconnected systems capable of adapting to dynamic environments and user needs.

1.3 Thesis Outline

This thesis addresses various communication challenges in learning systems, with each chapter delving deeper into specific learning-related issues than the previous one. Chapter 2 explores asynchronous DSGD in the context of unstable user experiences, focusing on the challenges of communication and computation delays. Chapter 3 proposes a framework for asynchronous decentralized users, which enhances the autonomy of each participant, allowing for more flexible and efficient learning processes. Chapter 4 examines the impact of personalization in cooperative learning across heterogeneous edge devices, introducing strategies to tailor learning processes to individual user needs and data distributions. Finally, Chapter 5 provides an overall summary of the thesis and discusses insightful future research directions that build on the findings presented in the preceding chapters.

The research conducted during the course of this thesis has resulted in the following publications:

- E. Jeong, M. Zecchin, and M. Kountouris, “Asynchronous decentralized learning over unreliable wireless networks”, in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 607–612. DOI: 10.1109/ICC45855.2022.9838891 [52].
- E. Jeong and M. Kountouris, “DRACO: a framework for decentralized asynchronous learning over continuous row-stochastic networks”, under review in *IEEE Open Journal of the Communications Society (OJCOMS)*, 2024 [53].
- E. Jeong and M. Kountouris, “Personalized decentralized federated learning with knowledge distillation”, in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 1982–1987. DOI: 10.1109/ICC45041.2023.10279714 [54].

2 Asynchronous Decentralized Learning over Unreliable Wireless Networks

Decentralized learning allows users at the network edge to collaboratively train models by sharing information through device-to-device communications. However, previous studies have primarily focused on wireless networks with static topologies and dependable participants. This research introduces an asynchronous DSGD algorithm designed to withstand the typical computational and communication failures at the wireless network edge. We provide a theoretical analysis of its performance and offer a non-asymptotic convergence guarantee. Our experimental findings support this analysis, showing the advantages of using asynchronous methods and the reuse of outdated gradient information in decentralized learning across unreliable wireless networks.

2.1 Introduction

Distributed learning algorithms enable devices within wireless networks to collaboratively refine model parameters by alternating between local optimization and communication phases. By harnessing the combined computational resources at the edge of wireless networks in a manner that is both communication-efficient [2] and privacy-preserving [55], distributed learning emerges as a critical technological driver for the intelligent networks of the future. A notable approach within this field is decentralized learning [56], which facilitates collaborative training among edge devices without needing a central server, employing a peer-to-peer communication style. Unlike federated learning, decentralized learning does not rely on a central parameter server or a star network topology, offering greater flexibility in terms of connectivity [5]. This characteristic makes decentralized learning especially suitable for future wireless networks that utilize device-to-device communications. Various decentralized learning approaches for wireless networks have been proposed and scrutinized [57]–[59], with particular emphasis on the use of Over-the-air Computation (AirComp) [60] to facilitate low-latency training at the network edge. Previous studies typically focus on wireless networks composed of reliable workers

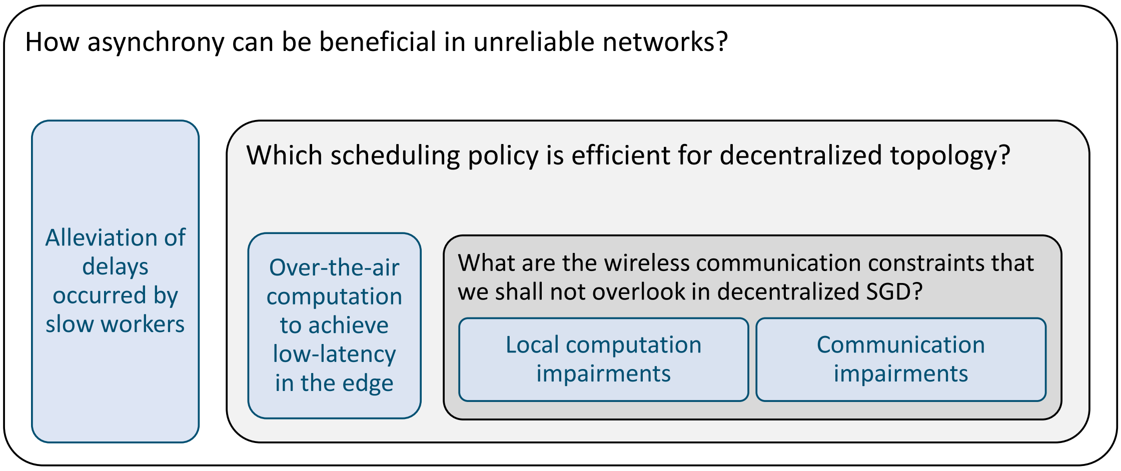

How asynchrony can be beneficial in unreliable networks?
Alleviation of delays occurred by slow workers
Which scheduling policy is efficient for decentralized topology?
Over-the-air computation to achieve low-latency in the edge
What are the wireless communication constraints that we shall not overlook in decentralized SGD?
Local computation impairments
Communication impairments

Figure 2.1. Problem settings in Asynchronous DSGD

with static topologies throughout training. However, such conditions are rarely met in real-world scenarios where communication links may be unstable or obstructed, and devices might become intermittently inactive due to computational constraints or energy conservation needs. Particularly, asynchronous DSGD faces significant challenges in real-world settings due to communication and computation impairments. These impairments arise from unstable wireless networks, varying device capabilities, and resource constraints. Communication issues include network delays, packet loss, and dynamic network topologies, while computation impairments stem from heterogeneous devices, resource limitations, and system failures. These impairments disrupt the smooth flow of updates between devices, leading to desynchronization, stale gradients, unbalanced contributions from devices, and increased iteration costs. This not only slows down the learning process but also hinders the scalability of DSGD in large-scale networks.

To address these issues, robust algorithms have been developed that can effectively handle stale gradients, adapt to dynamic environments, and manage communication efficiently despite impairments. Distributed training has shown promise in mitigating the impact of stragglers (slower workers) [61]–[63] by leveraging the personalization of each client. However, fully harnessing the benefits of asynchronism in decentralized learning across wireless networks remains a significant and ongoing challenge, especially for networks that aim for global consensus among unreliable users.

This chapter introduces an asynchronous version of DSGD tailored to tackle the inherent communication and computation challenges in heterogeneous wireless networks. We explore decentralized learning across networks with randomly fluctuating, time-varying communication topologies and unreliable devices that may become stragglers at any stage of the process. To manage communication disruptions, we employ a consensus approach utilizing time-dependent mixing matrices that reflect the current state of the network. Concurrently, we calibrate the learning rates for devices at the network edge to maintain the stationary point of the overarching network objective, despite varied computational capabilities. We also provide a non-asymptotic convergence guarantee for our algorithm, affirming that decentralized learning can be effective even when using outdated information from slower devices to locally train models. Our experimental findings validate this approach and indicate that reusing stale gradient information can accelerate the convergence of asynchronous DSGD.

2.2 System Model

Consider a group of N wireless edge devices in which $f_i : \mathbb{R}^{n_\theta} \to \mathbb{R}$ represents a local loss function endowed in each node i. The network targets to minimize the averaged loss subject to a consensus constraint

$$\begin{aligned} \underset{\theta_1,\ldots,\theta_N}{\text{minimize}}\ f(\theta_1,\ldots,\theta_m) &:= \frac{1}{N}\sum_{i=1}^{N} f_i(\theta_i)\,, \\ \text{s.t.}\quad \theta_1 = \theta_2 &= \cdots = \theta_N. \end{aligned} \tag{2.1}$$

where $\theta_i \in \mathbb{R}^{n_\theta}$ indicates local parameter estimation. This represents the distributed empirical risk minimization problem when f_i is a loss function over a local dataset. In this context, $f(\theta)$ denotes the network objective $f(\theta_1,\ldots,\theta_N)\big|_{\theta_1=\cdots=\theta_N=\theta}$, and $\bar{\theta} = \frac{1}{N}\sum_{i=1}^{N}\theta_i$. To solve equation (2.1), we utilize a DSGD algorithm where devices alternate between local optimization based on gradient information (i.e., a computation phase) and a communication phase.

2.2.1 Computation model

To locally optimize the model estimation θ_i, we assume that each device can query a stochastic oracle satisfying the following properties.

Assumption 2.1. *At each node i, the gradient oracle $g_i(\theta)$ satisfies the following properties for all $\theta \in \mathbb{R}^{n_\theta}$*

- $\mathbb{E}[g_i(\theta)] = \nabla_\theta f_i(\theta)$ *(unbiasedness)*
- $\mathbb{E}\|g_i(\theta) - \nabla_\theta f_i(\theta)\|^2 \le \sigma^2$ *(bounded variance)*
- $\mathbb{E}\|g_i(\theta)\| \le G^2$ *(bounded magnitude).*

We recognize that certain nodes, termed stragglers, may become inactive or delay their local optimization procedures, potentially due to computational difficulties or energy limitations.

Consequently, these devices might enter the communication phase with a model that either incorporates gradient information from outdated model estimations or remains unchanged from previous iterations. Specifically, at each optimization round r, the local update mechanism is governed by the following rule:

$$\theta_i^{(r+\frac{1}{2})} = \begin{cases} \theta_i^{(r)}, & \text{if device } i \text{ is a straggler at round } r \\ \theta_i^{(r)} - \eta_i^t g_i(\theta^{(r-\tau_i)}), & \text{otherwise} \end{cases} \tag{2.2}$$

Here, η_i^r represents the local learning rate, and $\tau_i \geq 0$ measures the delay, accounting for the staleness of the gradient information at device i.

2.2.2 Communication model

The communication link between any two devices, i and j, is modeled using Rayleigh fading. During each communication iteration r, devices exchange information based on a connectivity graph $\mathcal{G}^{(r)} = (\mathcal{U}, \mathcal{E}^{(r)})$, where $\mathcal{U} = \{1, 2, \ldots, N\}$ represents the network nodes, and $(i, j) \in \mathcal{E}^{(r)}$ signifies that devices i and j can communicate in round r. These communication links are symmetric, making the graph undirected. While the connectivity graph remains static within a single optimization iteration, it can vary between iterations due to factors such as deep fading, physical obstructions, or synchronization issues.

2.3 Asynchronous Decentralized SGD

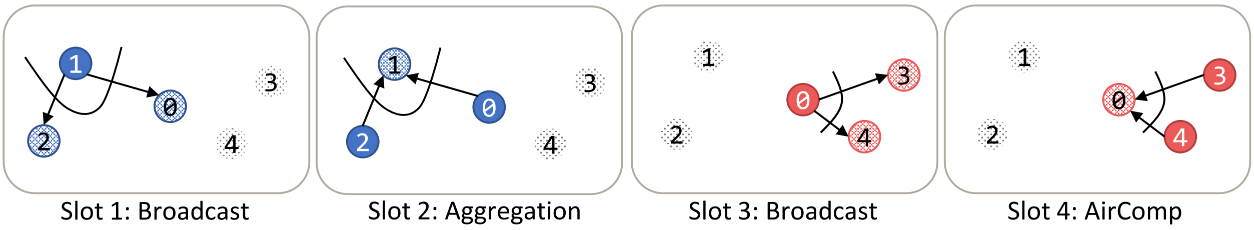


Figure 2.2. An example of the timeline for one training iteration composed of alternate Broadcast and AirComp slots.

The proposed asynchronous DSGD procedure, which accounts for both computation and communication failures, is detailed in Algorithm 2. At the start of each training iteration r, non-straggling devices update their local estimation $\theta_i^{(r)}$ according to the updating policy described in eq. (2.2), potentially using outdated gradient information. Following this, based on the current connectivity graph $\mathcal{G}^{(r)} = (\mathcal{U}, \mathcal{E}^{(r)})$, devices establish a symmetric and doubly stochastic

Algorithm 2: Asynchronous Decentralized SGD

```
Input: $\theta_i^{(0)} = \mathbf{0} \in \mathbb{R}^d$
Output: $\bar{\theta}^{(R)}$
for $r$ in $[0,R]$ do
    for each non straggling devices do
        update local model as (2.2)
    end for
    Determine matrix $W^{(r)}$ based on $\mathcal{G}^{(r)}$
    for $s$ in $[1,S_t]$ do
        if $s \equiv 0 \pmod 2$ then
            # Broadcast phase
            for each device $i$ scheduled in slot $s$ do
                Device $i$ transmits (2.6)
                Each device $j \in \mathcal{N}_i^{(r)}$ receives (2.7)
                Each device $j \in \mathcal{N}_i^{(r)}$ estimates (2.8)
            end for
        else
            # AirComp Phase
            for each star center $i$ scheduled in slot $s$ do
                Each device $j \in \mathcal{N}_i^{(r)}$ transmits (2.6)
                Device $i$ receives (2.4)
                Device $i$ estimates (2.5)
            end for
        end if
    end for
    for each device do
        model consensus as in (2.9)
    end for
end for
```

mixing matrix $W^{(r)}$ using the Metropolis-Hastings weighting scheme [64]. These weights are straightforward to compute and suitable for distributed implementation, as each device only needs to know the degrees of its neighbors to determine the weights on its adjacent edges.

Subsequently, the communication phase begins, during which devices exchange their updated estimations and utilize a gossip scheme based on $W^{(r)}$. To exploit the capabilities of AirComp, devices use analog transmission in conjunction with the scheduling scheme proposed in [57]. Consequently, the communication phase is divided into multiple pairs of communication slots, each consisting of an *AirComp slot* and a *broadcast slot*, as illustrated in Fig. 2.2. During the AirComp slot s, the star center i receives the superposition of signals transmitted by its neighboring devices $\mathcal{N}^{(r)}(i) = \{j \in \mathcal{U} : (i,j) \in \mathcal{E}^{(r)}\}$. Specifically, each scheduled node

$j \in \mathcal{N}^{(r)}(i)$ transmits to the star center i as follows:

$$y_j^{(s,r)} = \frac{\sqrt{c_i^{(s,r)}}}{h_{i,j}^{(s,r)}} w_{i,j}^{(r)} \theta_j^{(r+\frac{1}{2})} \tag{2.3}$$

where $h_{i,j}^{(s,r)} \in \mathbb{C}^{n_{\mathbb{x}}}$ is the channel coefficient between user i and j during slot s, $c_i^{(s,r)} \in \mathbb{R}$ is a power alignment coefficient, and $w_{i,j}^{(r)}$ is the (i,j) entry of the mixing matrix W. The star center i receives the aggregated signal

$$\tilde{y}_i^{(s,r)} = \sum_{j \in \mathcal{N}(i)} h_{i,j}^{(s,r)} y_j^{(s,r)} + z_i^{(s,r)} \tag{2.4}$$

where $z_i^{(s,r)} \sim \mathcal{N}(0, \sigma_w \mathbb{1}_d)$ is a noise vector, and estimates the aggregated model as

$$\hat{y}_i^{(s,r)} = \frac{\tilde{y}_i^{(s,r)}}{\sqrt{c_i^{(s,r)}}} = \sum_{j \in \mathcal{N}(i)} w_{i,j}^{(r)} \theta_j^{(r+\frac{1}{2})} + \frac{z_i^{(s,r)}}{\sqrt{c_i^{(s,r)}}}. \tag{2.5}$$

During a broadcast slot s, the scheduled node i transmits the signal

$$y_i^{(s,r)} = \sqrt{a_i^{(s,r)}} \theta_i^{(r+\frac{1}{2})} \tag{2.6}$$

using a power scaling factor $a_i^{(s,r)}$, and all neighboring devices $j \in \mathcal{N}^{(r)}(i)$ receive

$$\tilde{y}_j^{(s,r)} = h_{j,i}^{(s,r)} y_i^{(s,r)} + z_j^{(s,r)} \tag{2.7}$$

and estimate the updated model as

$$\hat{y}_j^{(s,r)} = w_{j,i}^{(r)} \frac{\tilde{y}_j^{(s,r)}}{\sqrt{a_i^{(s,r)}} h_{j,i}^{(s,r)}} = w_{j,i}^{(r)} \left(\theta_i^{(r+\frac{1}{2})} + \frac{z_j^{(s,r)}}{\sqrt{a_i} h_{j,i}} \right). \tag{2.8}$$

At the end of the communication phase, each node i obtains the new estimation $\theta_i^{(r+1)}$ by combining all received signals and using a consensus with step size $\xi \in (0,1]$:

$$\theta_i^{(r+1)} = (1-\xi)\theta_i^{(r+\frac{1}{2})} + \xi \left\{ \sum_{j=1}^{N} w_{i,j}^{(r)} \theta_j^{(r+\frac{1}{2})} + \tilde{n}_i^{(r)} \right\} \tag{2.9}$$

where $\tilde{n}_i^{(r)} \sim \mathcal{N}(0, \tilde{\sigma}_{w,i}^{(r)} \mathbb{1}_{n_{\mathbb{x}}})$ is a noise vector term accounting for the aggregation of noise components during AirComp and broadcast transmissions at device i during communication phase r.

2.4 Convergence Analysis

In this section, we study the effect of communication and computation failures on the asynchronous DSGD procedure and prove its convergence.

2.4.1 Effect of Communication Failures

Communication disruptions lead to a randomly varying connectivity graph, where the set of edges changes with each optimization iteration. From an algorithmic standpoint, these random communication impairments manifest in the DSGD algorithm through stochastic mixing matrices. A notable category within these stochastic mixing matrices are those that adhere to the expected consensus property.

Definition 2.1 (Expected Consensus Rate [5]). *A random matrix $W \in \mathbb{R}^{N\times N}$ is said to satisfy the expected consensus with rate p if for any $X \in \mathbb{R}^{d\times N}$*

$$\mathbb{E}_W\left[\left\|WX-\bar{X}\right\|_F^2\right] \le (1-p)\left\|X-\bar{X}\right\|_F^2$$

where $\bar{X} = X\frac{\mathbf{11}^R}{N}$ and the expectation is with respect to the random matrix W.

Lemma 2.1. *If the event that the connectivity graph $\mathcal{G}^{(r)}$ is connected at round r has a probability $q > 0$ and the Metropolis-Hastings weighting is used to generated the mixing $W^{(r)}$, the expected consensus rate is satisfied with rate $p = q\delta > 0$, with δ being the expected consensus rate in case of a connected topology.*

Proof. Define the event $E^{(r)} := \{\mathcal{G}^{(r)}$ is connected$\}$ and its complementary event $\bar{E}^{(r)}$. Whenever the Metropolis-Hasting weights are obtained from a connected graph, the resulting mixing matrix $W^{(r)}$ has a consensus rate greater than zero. Therefore, there exists $\delta > 0$ such that

$$\mathbb{E}_{W^{(r)}|E^{(r)}}\left\|W^{(r)}X-\bar{X}\right\|_F^2 \le (1-\delta)\left\|W^{(r)}X-\bar{X}\right\|_F^2$$

It follows that, for any $X \in \mathbb{R}^{d\times m}$

$$\begin{aligned}\mathbb{E}_{W^{(r)}}\left\|W^{(r)}X-\bar{X}\right\|_F^2 &= q\mathbb{E}_{W^{(r)}|E^{(r)}}\left\|W^{(r)}X-\bar{X}\right\|_F^2 + (1-q)\mathbb{E}_{W^{(r)}|\bar{E}^{(r)}}\left\|X-\bar{X}\right\|_F^2 \\ &\le q(1-\delta)\left\|W^{(r)}X-\bar{X}\right\|_F^2 + (1-q)\left\|X-\bar{X}\right\|_F^2\end{aligned}$$

where we have lower bounded the consensus rate by zero in case of disconnected topologies. Grouping terms and having assumed $q > 0$, we obtain that the expected consensus is satisfied with rate $(1-q\delta) > 0$. □

If the expected consensus is satisfied, it is then possible to establish a convergent behavior for the estimations generated by the proposed algorithm.

Lemma 2.2 (Consensus inequality)**.** *Under Assumption 2.1, after R iterations, DSGD with a constant learning rate η and consensus step size ξ satisfies*

$$\sum_{i=1}^{N}\left\|\theta_i^{(R)}-\bar{\theta}^{(R)}\right\|_2 \leq \eta^2\frac{12mG^2}{(p\xi)^2}+\xi\frac{2}{p}\sum_{i=1}^{m}\sigma_{w,i}^2$$

where $\sigma_{w,i}^2=\max_{r=0}^{R}\mathbb{E}\left[\left\|\tilde{n}_i^{(r)}\right\|^2\right]$.

Proof. Similarly to [57], [65] we establish the following recursive inequality

$$\sum_{i=1}^{m}\mathbb{E}\left\|\theta^{(r)}-\bar{\theta}^{(r)}\right\|^2\leq\left(1-\frac{p\zeta}{2}\right)\sum_{i=1}^{m}\mathbb{E}\left\|\theta^{(t-1)}-\bar{\theta}^{(t-1)}\right\|^2+\frac{\eta^2}{p\zeta}\left(6mG^2\right)+\zeta^2\sum_{i=1}^{N}\mathbb{E}\left\|\tilde{n}_i^{(r)}\right\|^2.$$

Defining $\sigma_{w,i}^2=\max_{t=0}^{T}\mathbb{E}\left\|\tilde{n}_i^{(r)}\right\|^2$ and then solving the recursion we obtain the final expression. □

Overall, communication failures amount to a reduced expected consensus rate compared to the scenario with perfect communication. At the same time, dropping users that are delayed and are unable to synchronize and perform AirComp, renders the communication protocol more flexible. For instance, in Fig. 2.3, we consider a network of nine nodes organized according to different topologies and show the evolution of the average spectral gap of the mixing matrix with Metropolis-Hastings weights, whenever devices not satisfying a certain delay constraint are dropped. As expected, stricter delay requirements result in sparser effective communication graphs and mixing matrices with smaller spectral gaps.

2.4.2 Effect of Computation Failures

Random computation impairments make the group of devices that effectively update the model parameter vary over time. To account for this in the analysis, we introduce a virtual learning rate that is zero in case of failed computation. Namely, the learning rate at device i during computation round r becomes

$$\tilde{\eta}_i^{(r)}=\begin{cases}0, & \text{if } i \text{ is straggler at round } r\\ \eta_i^{(r)}, & \text{otherwise}\end{cases}$$

where $\eta_i^{(r)}$ is a specified learning rate value in case of successful computation. Furthermore, to ensure that the procedure converges to stationary points of the network objective even when

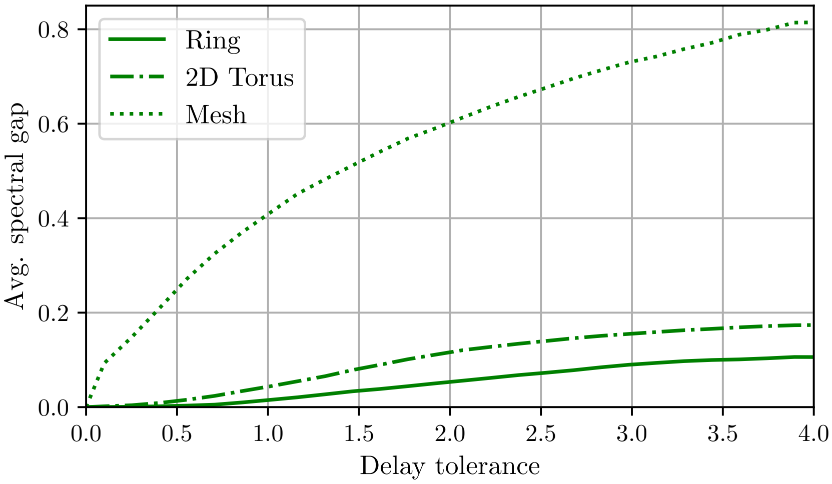


Figure 2.3. Average spectral gap under different delay constraints for mesh, ring, and two-dimensional torus topologies with 9 nodes. Each link is associated to a completion time $\sim Exp(1)$ and is dropped if it exceeds the delay tolerance value.

edge devices have different computing capabilities, the expected learning rates have to be equalized. In particular, if $\mathbb{E}[\eta_i^{(r)}] = \eta, \ \forall i$, we have that stationary points are maintained in expectation, namely

$$\sum_{i=1}^{N} \mathbb{E}[\tilde{\eta}_i^{(r)}]\nabla f_i(\theta) = 0 \implies \nabla f(\theta) = 0.$$

Finally, the existence of straggling devices introduces asynchronicity in the decentralized optimization procedure. In particular, a device i that fails at completing the gradient computation at a given optimization iteration is allowed to apply the result in a later one, without discarding the computation results. While we do not specify the delay distribution, we rather introduce the following assumption regarding the staleness of gradients.

Assumption 2.2. *For all iteration r, there exists a constant $c \leq 1$ such that*

$$\mathbb{E}\left\|\nabla f(\bar{\theta}^{(r)}) - \frac{\sum_{i=1}^{N} \nabla f_i(\theta_i^{(r-\tau_i)})}{N}\right\|^2 \leq c\mathbb{E}\left\|\nabla f(\bar{\theta}^{(r)})\right\|^2 + L^2 \frac{\sum_{i=1}^{N} \mathbb{E}\left\|\theta_i^{(r)} - \bar{\theta}^{(r)}\right\|^2}{N}.$$

The above assumption is similar to the one in [66] with an additional consensus error term. Note that the value of c is proportional to the staleness of the gradients and in case of perfect synchronization ($c = 0$) the bound amounts to a standard consensus error term.

2.4.3 Convergence Guarantee

In this subsection, we demonstrate the convergence of the decentralized optimization procedure to a stationary point of the problem (2.1).

Theorem 2.1. *Consider a network of unreliable communicating devices in which the expected consensus rate is satisfied with constant* p *and each device can be a straggler with probability* $\nu_i < 1$. *If Assumptions 2.1 and 2.2 are satisfied, asynchronous DSGD with constant learning rate* $\eta_i = \min_j(1-\nu_j)/(\sqrt{4LR}(1-\nu_i))$ *and consensus rate* $\xi = 1/R^{3/8}$ *satisfies the following stationary condition*

$$\frac{1}{R}\sum_{r=1}^{R}\left\|\nabla f(\bar{\theta}^{(r)})\right\|^2 \leq \frac{8\sqrt{L}(f(\bar{\theta}^{(R)})-f^*)}{c'\nu_{min}\sqrt{R}} + \frac{3G^2L}{R^{1/4}p^2c'} + \sqrt{\frac{L}{4R}}\frac{\sigma^2}{Nc'\min_j(1-\nu_j)}$$
$$+\sum_{i=1}^{N}\frac{\sigma^2_{w,i}}{Nc'}\left(\frac{2L^2c}{pR^{3/8}} + \frac{4L\sqrt{L}}{NR^{1/4}\nu_{min}}\right)$$

where $c' = 1-c$, $\nu_{min} = \min_j(1-\nu_j)$ *and* $f^* = \min_{\theta\in\mathbb{R}^{n_{\mathbb{X}}}} f(\theta)$.

Proof. See Appendix A.1. □

The above theorem establishes a vanishing bound on the stationarity of the returned solution, which involves quantities related to both communication and computation impairments. In particular, the constant of the slowest vanishing terms $R^{-1/4}$ contains the term p related to random connectivity, as well as c' and ν_{min} due to stragglers.

2.5 Numerical Results

The effectiveness of the proposed asynchronous DSGD scheme is assessed using a network of $N = 15$ devices that collaboratively optimize the parameters of a Convolutional Neural Network (CNN) for image classification with Fashion-MNIST [67]. Gradients are calculated using batches of 16 data samples and the performance is evaluated using a test set of 500 images. We model the channel gain between each device pair as Rayleigh fading and we assume a shifted exponential computation time at each device, i.e., $T_{comp} = T_{min} + Exp(\lambda)$ with $T_{min} = 0.25$s and $\lambda = 1$. In Fig. 2.4, nodes communicate only when the channel is in favorable conditions, i.e., when the channel gain exceeds a certain minimum threshold h_{min}. This allows to save energy; however, while higher threshold values result into lower average energy consumption, they also produce mixing matrices with smaller consensus rate, thus increasing the convergence time.

To study the effect of computation impairments, our proposed asynchronous learning algorithm is compared with: (i) *synchronous DSGD*, which waits for all devices to finish their compu-

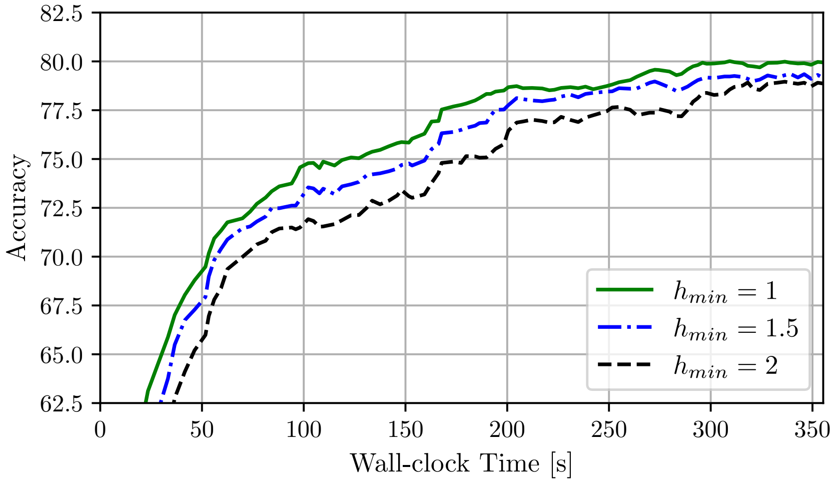


Figure 2.4. Test accuracy versus time under different channel gain thresholds. Smaller thresholds result in larger average consensus rates and therefore in faster convergence.

tations; and (ii) *synchronous DSGD with a delay barrier* Γ_{max}, which discards computation from users that violate the maximum computing time. Compared to the latter, our asynchronous procedure allows for slow devices to reuse stale gradient computations during later iterations. In Fig. 2.5, we plot the evolution of the test accuracy of the aforementioned algorithms under two different values of $\Gamma_{\max}$. For a moderate delay constraint $\Gamma_{\max} = \mathbb{E}[T_{comp}]$, asynchronous DSGD and synchronous DSGD with delay barrier perform similarly as the fraction of slow users is modest. Nonetheless, imposing a delay constraint and discarding slow devices greatly reduces the training time compared to the synchronous DSGD case. On the other hand, for a stringent delay requirement, $\Gamma_{\max} = \frac{4}{5}\mathbb{E}[T_{comp}]$, reusing stale gradients turns out to be beneficial and the proposed asynchronous DSGD attains higher accuracy faster compared to the synchronous DSGD with a delay barrier.

2.6 Conclusion

In this chapter, we have proposed and analyzed an asynchronous implementation of DSGD, enabling decentralized optimization over realistic wireless networks characterized by unreliable communication and heterogeneous computational capabilities. We examined the effects of communication and computation failures on training performance and provided non-asymptotic convergence guarantees for the proposed algorithm. A key finding is that reusing outdated gradient information from slower devices proves beneficial in asynchronous decentralized learning.

However, the asynchronous learning scheme presented in this chapter has limitations due to analog transmission. Specifically, the star nodes must aggregate received updates at spe-

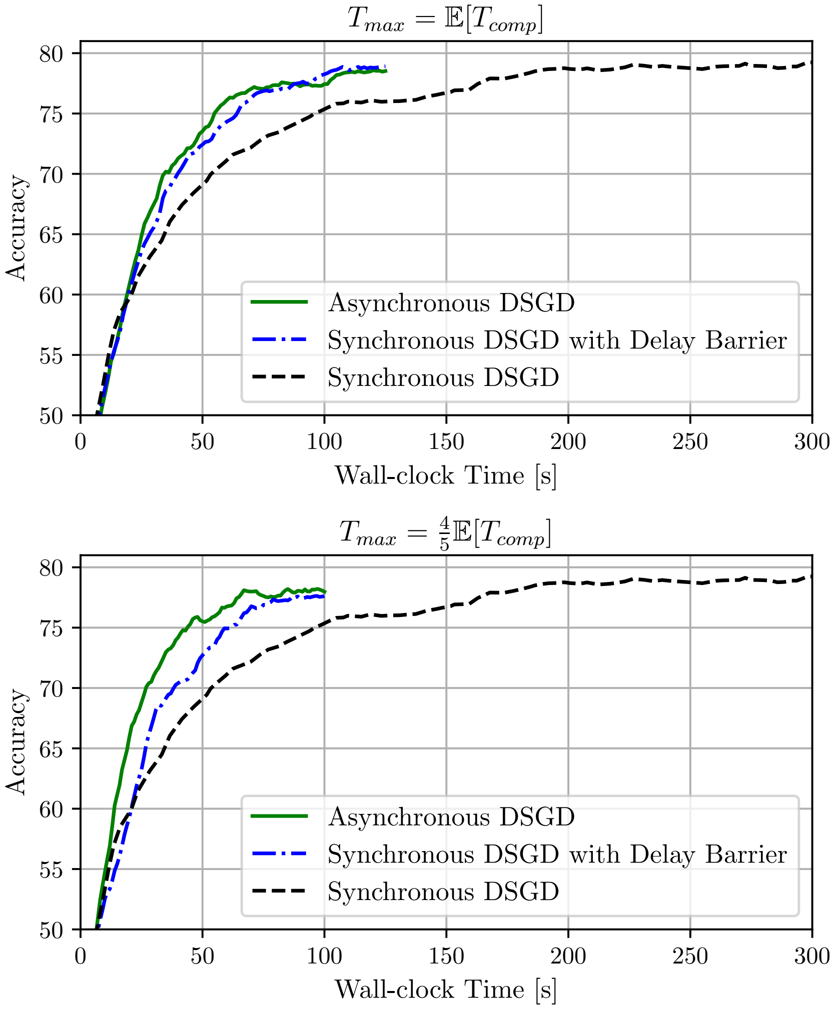


Figure 2.5. Test accuracy for the asynchronous, synchronous with delay barrier, and synchronous schemes under two different values of $\Gamma_{\max}$.

cific intervals, multiples of a unit time slot. If the delay τ is not a multiple of s, the received messages may not align with the current model, necessitating additional techniques to correct these misalignments. Furthermore, the protocol requires only one node to be active during each broadcasting step, which can slow the learning process, especially for large networks.

Beyond these limitations, the next chapter introduces an advanced framework for asynchronous and decentralized learning, termed DRACO, which aims to solve broader and deeper issues related to asynchronous decentralized learning.

3 DRACO: Decentralized Asynchronous Federated Learning over Continuous Row-Stochastic Network Matrices

Recent developments and emerging use cases, such as smart IoT and Edge Artificial Intelligence (AI), have sparked considerable interest in the training of neural networks over fully decentralized (serverless) networks. One of the major challenges of decentralized learning is to ensure stable convergence without resorting to strong assumptions applied for each agent regarding data distributions or updating policies. To address these issues, we propose DRACO, a novel method for decentralized asynchronous Stochastic Gradient Descent (SGD) over row-stochastic gossip wireless networks by leveraging continuous communication. Our approach enables edge devices within decentralized networks to perform local training and model exchanging along a continuous timeline, thereby eliminating the necessity for synchronized timing. The algorithm also features a specific technique of decoupling communication and computation schedules, which empowers complete autonomy for all users and manageable instructions for stragglers. We highlight the advantages of asynchronous and autonomous participation in decentralized optimization through a comprehensive convergence analysis. Our numerical experiments corroborate the efficacy of the proposed technique.

3.1 Introduction

Recent advancements in machine learning, networked intelligent systems, and wireless connectivity has paved the way for various innovative applications and use cases across various sectors, including the IoT, consumer robotics, autonomous transportation, and edge computing. These systems increasingly rely on decentralized learning architectures for processing data where generated, minimizing latency and bandwidth usage while enhancing privacy. However, these benefits come with significant challenges, particularly in terms of ensuring efficient and reliable communication and processing within inherently unstable and diverse network environments. Addressing these challenges requires novel approaches that adapt to the unique demands of decentralized architectures, fostering robust and expandable solutions for real-time data processing and learning.

In this work, we consider the problem of communication efficiency in FL [2] and in particular in serverless (fully decentralized) learning settings that operate without a central coordinating server [13], [68]–[71]. Asynchronous learning, empowering each participant to conduct local training and data transmission at their own pace, is a standard and relevant design choice in decentralized network schemes [72]–[77]. Asynchronous and decentralized learning have an advantage when used separately from each other, manifesting as adaptability to limited resources and downsized communication overhead. Yet, unfortunately, when these two paradigms are combined, their integration poses a greater challenge in achieving a unanimous global consensus, as required, for instance, in the development of sophisticated navigation algorithms [78].

Decentralized optimization studies in the literature often involve high "synchronization costs" due to the complexity of ensuring consensus. In other words, the majority of asynchronous learning schemes are executable only if all participants have a common sense of the global communication rounds, which have to be, in a way, synchronously counted. This paradoxical agreement in synchronized clocks takes an additional cost to bear while carrying out related techniques over wireless networks with message losses or delays due to unstable channel conditions. As a result, the focus has shifted towards analyzing decentralized learning using asynchronous gossip protocols [52], [79]–[82]. The introduction of gossiping, leveraging random or probabilistic communication, has rendered the proposed algorithms more compelling than those relying on predefined schedules, mainly thanks to their resilience to dynamically changing connectivity [83].

However, existing studies on asynchronous gossip optimization usually adopt several strong assumptions that make the proposed solutions less applicable in realistic scenarios, particularly when involving wireless communication protocols. Early works on distributed optimization have relied on doubly stochastic weights, which are suitable only for undirected or balanced networks [84]. Despite the prevalence of the assumptions among many related studies, algorithms designed with doubly stochastic weights cannot be constructed over arbitrarily directed graphs [85]. Distributed optimization over directed graphs has been extensively studied in control theory [86]–[89]. More recently, federated learning (FL) research has started addressing challenges related to asymmetric connectivity, using row-stochastic matrices [90] and time-varying directed graphs [91]. Some works have even incorporated personalization [92]. However, a significant research gap remains in exploring how collaborative learning networks can maintain robustness in the presence of unreliable transmissions.

These stringent assumptions have been introduced intentionally or inevitably since decen-

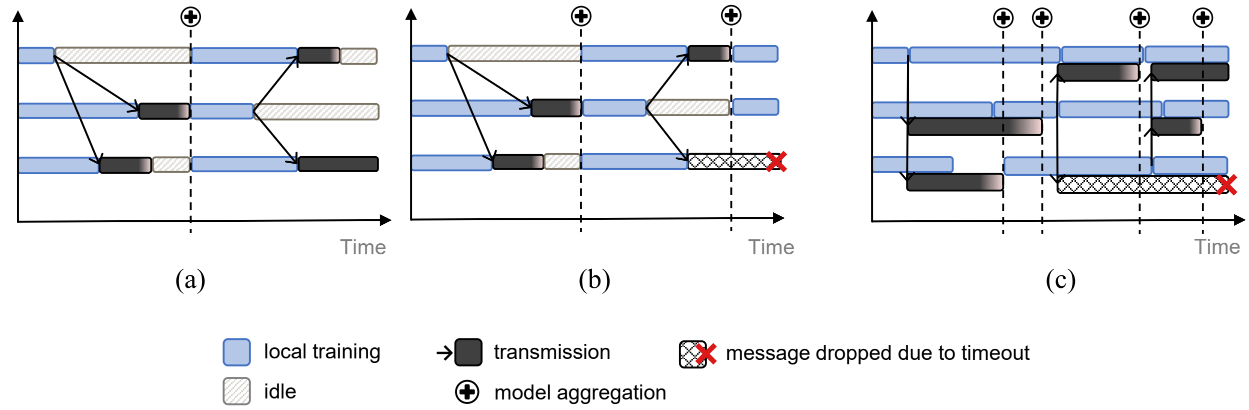


Figure 3.1. A schematic view of DRACO's timelines with comparisons. (a) Synchronous FL; (b) asynchronous FL with transmission delay deadline; (c) (in DRACO) fully asynchronous FL with delay deadline, but the iteration count is continuous.

tralized learning over gossip communication presents several technical challenges. One of the major challenges is uncertainty in convergence; for instance, irregular or non-uniform communication probabilities in the gossip network can lead to variable convergence rates or convergence to suboptimal solutions. Furthermore, the performance of decentralized learning algorithms, for instance in terms of convergence rate and communication load per iteration, is more sensitive to the specific network topology or graph [93]. Small changes in communication probabilities or network configuration could significantly impact learning dynamics. Therefore, addressing these difficulties requires specialized algorithmic solutions and in-depth analyses to guarantee the effectiveness, stability, and convergence of asynchronous learning over decentralized networks. This involves developing algorithms that adapt to irregular communication probabilities while maintaining robustness and efficiency across diverse network structures.

In this work, we introduce DRACO, a novel framework for decentralized asynchronous FL. In brief, we propose a scheme characterized by two foundational elements: (i) it facilitates continuous, asynchronous operations without a global iteration count, and (ii) it employs decoupling communication and computation strategies, integrated with gradient pushing.

Firstly, our asynchronous learning approach operates continuously, permitting messages to be sent and received at non-uniform, non-integer time instants. This flexibility translates into that message arrivals are not confined to specific multiples of a global round duration, allowing each node to operate independently, based on its own schedule. The variability in each client's timeline is illustrated in Fig. 3.1(c). For instance, while User 3 is engaged in its second local updating round, User 1 is already progressing through its third round. Although this approach makes it difficult to trace the progress of each client's model at any given moment due to the variability in their timelines, it significantly lowers their idle time, enhancing the efficiency of the learning process. To alleviate the impact of outdated gradients that may impede local models from being optimized, messages that exceed a certain delay threshold are disregarded.

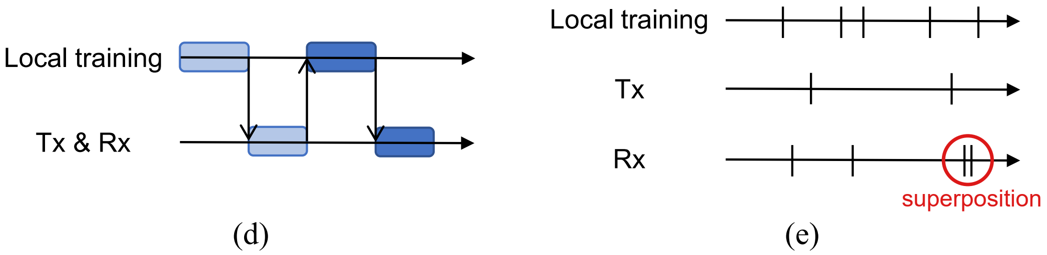


Figure 3.2. A schematic view of DRACO's timelines with comparisons. (d) sequential computation and communication over a doubly-stochastic network; (e) timelines of DRACO with decoupled computation and communication over a row-stochastic network. If two messages arrive at the same agent with a negligibly small time gap (in red circle), they are considered simultaneous and are used for the same model aggregation step. The concept of superposition window is elaborated in Section 3.2.2.

Secondly, DRACO leverages decoupled communication and computation schedules, as illustrated in Fig. 3.2(e). In a fully asynchronous network, the integrated learning process is less likely to stagnate when local training and transmission occur independently. In an environment where all users are busy, as in Fig. 3.1(c), if they always forward their new reference models after aggregating local updates from its neighbors (one-hop senders) like in Fig. 3.2(d), that way of exchange jeopardizes the optimization by communication overloads twice as heavy as push-based collaboration. Furthermore, the content delivered to each other can often be duplicated or overwritten. Thus, separating the two types of schedules departs from conventional methodologies that mandate a sequential or predetermined order for gradient updates and gossip communications.

This chapter introduces an asynchronous learning framework within a fully decentralized network, accommodating asymmetric communication weight graphs. Our approach distinguishes itself from existing works in several key aspects. First, it introduces an asynchronous and decentralized learning model in a continuously defined timeline, removing the need for quantized transmission schedules. Consequently, our proposed technique exhibits adaptability to dynamic network conditions.

Second, we introduce a novel and more realistic approach to addressing asynchrony in intelligent wireless networks. Rather than limiting the analysis to comparisons between synchronous and asynchronous communication or centralized and decentralized learning, our study embraces asynchrony and the absence of a central server as inherent challenges. In this context, we aim to investigate whether our proposed framework can substantially improve user performance. Additionally, we address the uncertainty and variability inherent in wireless networks, enhancing the scheme's resilience to fluctuations in connectivity. By incorporating these features, our work contributes a unique perspective to decentralized learning, offering a practical and efficient solution for real-world scenarios.

3.1.1 Related Works

Asynchronous decentralized learning In synchronous learning systems, all participants ought to wait for the slowest learner, known as a straggler, before proceeding to the next global round. As depicted in Fig. 3.1b, asynchronous learning with a transmission delay deadline effectively reduces the overall training time of synchronous systems by excluding users whose updates arrive after a predetermined deadline [52], [57]. This approach is applied not only to asynchronous settings but to synchronous learning through partial participation [94]. Both asynchronous learning and partially participating synchronous learning face the challenge of variance reduction since only a subset of local updates is considered in each training round [95]–[98]. Despite fewer average participation in model aggregation per user compared to synchronous methods, asynchronous learning performs as well as its counterpart, especially in solving large-scale multi-user optimization problems [99]. Nevertheless, this approach requires users to start their computations simultaneously to synchronize the global phase, leading to idle times when a message arrives before the start of the next iteration. Additionally, sufficient local storage is necessary to manage multiple messages queued in the receive buffer until the next round.

Randomized communication over serverless and directed networks Recent studies in decentralized learning have explored algorithms implementable for networks modeled by directed graphs, where the connectivity matrix is not necessarily doubly stochastic. This adaptation is often necessary when neither full-duplex nor half-duplex systems can ensure stable gradient transmissions. Techniques, such as push-sum [100]–[106], push-pull [107], [108], and random walk [109]–[111], have been proposed to improve decentralized optimization on directed graphs. Meanwhile, row-stochastic communication [112] significantly reduces both the number of communication rounds and storage requirements on edge devices; hence, this benefits in tackling complex problems, specifically those involving small-scale neural networks [89]. Among random communication protocols, gossip protocol is well-known for its rapid information spread but also criticized for its high network resource consumption [113]. Consequently, asynchronous gossip learning in such contexts needs innovative approaches to manage information flow among edge devices [114].

Decoupling communication and computation Unlike traditional methods that align gradient computation and communication either sequentially or in parallel, decoupling these processes significantly accelerates peer-to-peer averaging by releasing clients from waiting for others [115]–[117]. In AD-OGP [75], the authors replaced global communication slots with an event-based aggregation system, encompassing activities such as prediction and local updating. This unified timeline of events is particularly well-suited for environments where users train locally at different computational speeds. However, the event types of AD-OGP are restricted to “prediction” and “local updating”, overlooking the impact of transmission delay. The authors assume that message delay, defined as the time gap between the latest prediction and the local updating event within a user, provides no insight into how long it takes for a message to reach a neighboring node. Despite the growing interest in approaches for effective timeline integration, only a few studies have explored decoupled model averaging over unreliable wireless networks, where issues such as packet loss or delays are prevalent.

3.2 System Model

We consider the following optimization task over N clients whose goal is to minimize

$$f(\theta) := \frac{1}{N}\sum_{i=1}^{N} f_i(\theta) \tag{3.1}$$

where $\theta \in \mathbb{R}^{n_\theta}$ is an n_θ-dimensional model parameter and $\mathcal{U} = \{1, \cdots, N\}$ is the set of network users. In a serverless network, there is no global model θ_t; instead, each agent i holds $\theta_t^{(i)}$, which serves as a reference for the globally acquired model. Therefore, the objective function can be rewritten as

$$\theta^* = \inf_{\theta \in \mathbb{R}^{n_\theta}} \sum_{i=1}^{N} f_i(\theta^{(i)}) \, . \tag{3.2}$$

To tackle the minimization problem described in (3.1) or (3.2), we adopt a decentralized stochastic gradient descent (DSGD) approach. In this approach, individual devices iteratively enhance their local models $\theta^{(i)}$ and subsequently share these estimates with their neighboring nodes, which in turn could vary over time.

3.2.1 Absence of a global belief

The underlying assumption regarding the global consensus is that each user cannot reach a global “true parameter”, denoted by θ^*, by local updates only. The global model θ should be a vector combined with the beliefs (pseudo-global model) at each agent, said $\theta = \{\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(N)}\}$. However, in practice, none of the agents works as a central server or aggregator, which can obtain a centralized global model. We therefore adopt a virtual global model $\bar{\theta}$ that could have been acquired by the superposition of all beliefs if the network had an entirely authorized server, i.e.,

$$\bar{\theta} = \mathbb{E}_{i \in \mathcal{U}}[\theta^{(i)}] = \frac{1}{N}\sum_{i=1}^{N}\theta^{(i)}.$$

Therefore, the expectation of model update during P is

$$\bar{\theta}_{t_0+P} - \bar{\theta}_{t_0} = \frac{1}{N}\sum_{i=1}^{N}\left(\theta^{(i)}_{t_0+P} - \theta^{(i)}_{t_0}\right).$$

3.2.2 Communication systems

In our work, the processes of computation and communication are decoupled, hence when to locally train and when to transmit the updates are determined independently at each user. Since there can be infinite instants between any two close events on the continuous timeline, each message is likely to arrive at a different moment. Thus, practically speaking, there is no aggregation during the entire process even though two updates arrive at the same destination node by a narrow margin of time. In this regard, we introduce a *superposition window*, which is analogous to congestion windows in Transmission Control Protocol (TCP) [118]. Similar to a TCP window, the superposition window in DRACO controls the flow of received updates by grouping the messages for one aggregation. This leads to lower computation costs due to the fact that renewal of the local reference model every time a message arrives is avoided.

This chapter investigates the influence of unreliable wireless communications and controlled transmissions on performance of DRACO. In our scenarios, we assumed time-invariant connectivity graphs, representing stable network topologies during the learning process. This simplified assumption enables a focused examination of how the frequency of successful message receptions and the inherent structure of the communication network affect learning convergence. Unlike traditional fixed-topology models, we explicitly account for the inherent unreliability of wireless channels. In our model, successful message delivery between connected nodes is not guaranteed, influenced by physical distance, interference, and channel capacity limitations. User nodes are randomly distributed in the environment, and their geographical positions directly impact communication probabilities. To provide a more comprehensive understanding beyond standard fixed-topology analyses, we evaluate the impact of the frequency of successful message receptions within a defined unification period (detailed in Section 3.3.1). This approach allows for a nuanced learning process assessment under various wireless channel conditions.

A weighted graph at a certain instance is mathematically defined as a $N \times N$-sized matrix where each element indicates whether i transmits its message to one of its neighbors j or not.

It follows a conditional probability distribution if there is a communication event on client i. Transmission incidents are defined as

$$q_k^i = \begin{cases} 1, & \text{if } i \text{ broadcasts } \Delta^{(i)} \text{ at } k \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

$$q_k^{i\to j} = \begin{cases} 1, & \text{if } j \text{ receives } i\text{'s message sent at } k \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

where k is the index of an event. We define the neighborhood of user i, denoted by $\mathcal{N}_t(i) = \{j|q_t^{i\to j} = 1\}$, as the set of all users j that have an edge going from i to j at time t. It is also possible to denote the neighbor set with respect to event k, such as $\mathcal{N}_k(i)$. Following the notation in [94], these participation indicators are normalized across all moments, i.e., $\sum_{j\in\mathcal{U}\setminus\{i\}} q_t^{i\to j} = 1$ for $q_t^{i\to j} \geq 0$ and for all $i,\ t$. In addition to the definition, we define $\rho < 1$ that satisfies $\sum_{j\in\mathcal{U}\setminus\{i\}}(q_t^{i\to j})^2 \leq \rho^2$ for all $i,\ t$.

A broadcasting event is a necessary condition for a reception event. Let e indicate the event of a client node i broadcasting its local update at time t. When this event e occurs, the other nodes receive the message based on a conditional probability distribution, which also implies that a set of neighbors (recipients) is decided after the occurrence of e. If by $\delta_i = 1$ we indicate the incidence that node i has broadcast a message at a given instant, the probability that another node j receives this message is conditioned on this event e. In other words, $P\left[\delta_{ij} = 1|\delta_i = 0\right] = 0$ for all i, j.

3.2.3 Local gradient computations

Each user performs stochastic gradient computations by iterating B batches of the local training datasets. Δ represents the local update of the model, defined as the difference between its state prior to the mini-batch training and its state after completing training on B batches of training samples.

Assumption 3.1. *(Exponential local gradient computation time.)* The computation time τ_i of the stochastic gradient $g_i(\theta) \in \mathbb{R}^d$ at user i is exponentially distributed, i.e., $\tau_i \sim Exp(\lambda_i)$.

In the context of point processes, one can consider a Poisson Point Process (PPP) along the real line by examining the count of points within a specific interval $(t_0, t_0 + P]$ [119]. For a homogeneous PPP with rate parameter $\lambda > 0$, the likelihood that the count of points, denoted by $num(t_0, t_0 + P]$, equals a certain integer m can be described by the following expression:

$$\Pr\{num(t_0, t_0 + P] = m\} = \frac{(\lambda P)^m}{m!} e^{-\lambda P}.$$

This formula calculates the probability of exactly m occurrences within the interval based on the Poisson distribution, where λP represents the expected number of points in the interval and $e^{-\lambda P}$ adjusts for the total rate of occurrences over the span.

3.3 DRACO: Proposed Decentralized Asynchronous Learning

The main rationale behind the proposed algorithm is to provide an answer to the following question: *How can we issue instructions to each user in the absence of a global time loop in the network?* To resolve this issue, we design the system such that each node focuses solely on its actions without considering the training progress or the channel conditions of the other nodes. Defining the algorithm within a unified time loop in asynchronous and fully decentralized networks presents several practical limitations in real-world systems. A significant challenge lies in the absence of a consistent global time reference, such as global iteration rounds, denoted as t, or timestamps marking the completion of each user's local computations, marked as k. This inconsistency arises because the total number of local training iterations varies across users, even when their updates are observed simultaneously. As a result, if the algorithm mandates exchanges every t_P seconds or every k_P global slots, some local models may fall behind in development due to completing fewer local training steps compared to others. To address this issue, we avoid defining the procedure as either sequential or simultaneous. Our algorithm adopts instead a unified global loop where all users work in parallel. This global loop effectively encapsulates the learning process conducted by each user.

At each instance, every user selects one of the following three statuses based on a probability distribution: (1) remaining idle, (2) transmitting a message to neighboring nodes, or (3) conducting local model training. Local computation involves batch training iterated B times to compute the update, termed Δ. During transmission, the user broadcasts its local update. If a node recognizes delivery from the other nodes, it switches to a fourth (4) status (receiving mode), renewing its reference model by aggregating the model updates from neighboring nodes. Unlike these four statuses, a node turns to the fifth (5) status when a periodic timeout occurs. As depicted in the yellow box in Figure 3.3, a temporary hub broadcasts its reference model instead of a local update when the time is a multiple of the period P. The corresponding explanation as a form of algorithm is provided in Algorithm 3 on page 31.

Note that the 'idle' state is included since we assume that the agents alter their states instantaneously, i.e., without delay. The node's status is considered idle when it does none of the aforementioned steps. However, in practice, any activity takes time to complete, implying that each timestamp represents the moment that each action just finished. By this interpretation,

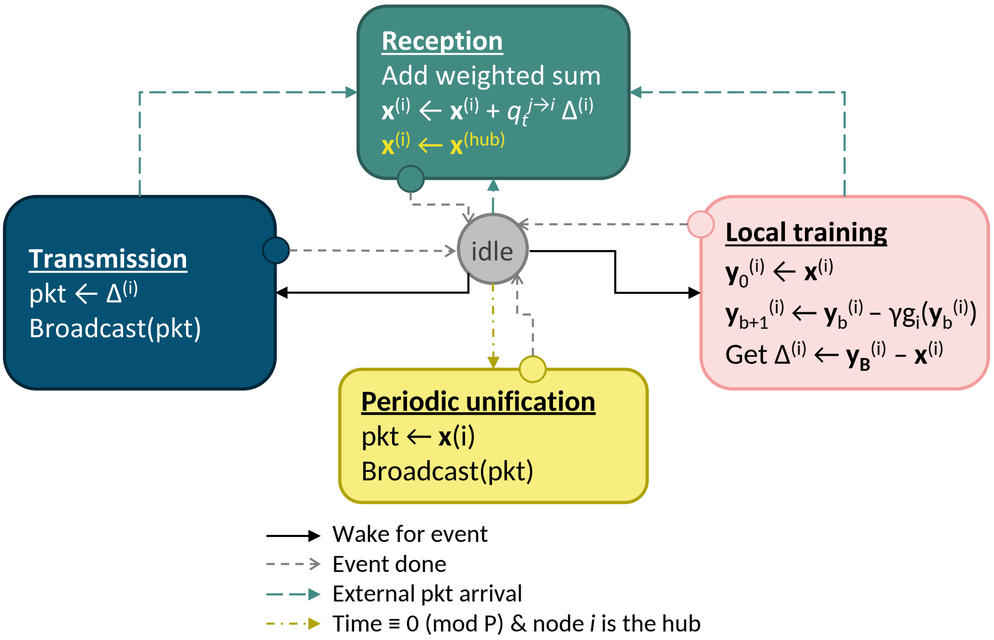

Reception
Add weighted sum
$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} + q_t^{j\to i}\,\Delta^{(i)}$
$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(hub)}$
idle
Transmission
pkt ← $\Delta^{(i)}$
Broadcast(pkt)
Local training
$\mathbf{y}_0^{(i)} \leftarrow \mathbf{x}^{(i)}$
$\mathbf{y}_{b+1}^{(i)} \leftarrow \mathbf{y}_b^{(i)} - \gamma g_i(\mathbf{y}_b^{(i)})$
Get $\Delta^{(i)} \leftarrow \mathbf{y}_B^{(i)} - \mathbf{x}^{(i)}$
Periodic unification
pkt ← $\mathbf{x}$(i)
Broadcast(pkt)
Wake for event
Event done
External pkt arrival
Time ≡ 0 (mod P) & node *i* is the hub

Figure 3.3. Our proposed algorithm in a chain graph illustrating the states of possible actions within each agent i.

the participants do not have an actual break time in practical scenarios, which is also applied to the experiments in Section 3.5.

Notations. $\mathcal{U}$ represents the set of participants within the network with $\mathcal{Q} := \{q_t^{i\to j}\}$ for all $i, j \in \mathcal{U}$ and t. Also, $\sum_{j\neq i}$ represents the summations of variables attributed to any user other than user i, i.e., $j \in \mathcal{U} \setminus \{i\}$. Throughout this manuscript, the term 'update' is used only as a noun that signifies the result derived from the difference between a local reference model and a newly obtained model through batch training. A user i's local update at time t is symbolized as $\Delta_t^{(i)}$. When a user i sends $\Delta^{(i)}$ or $\theta^{(i)}$, the recipient j receives $\tilde{\Delta}^{(i)}$ or $\tilde{\theta}^{(i)}$, which are identical to the sender's original contents if the transmission is free from distortion. To avoid potential confusion, any instances in this paper that involve the action of updating are called alternatively, such as 'renew' or 'iterate on'.

3.3.1 Periodic Unification

Local models are likely to diverge when the network does not use a central server, because no one synchronizes its different learning stages. Like conventional FedAvg, periodic unification can effectively resolve the variance-reduction problem among local reference models. A countable upper bound for the number of messages per unit time is required for analysis because otherwise, the losses diverge to infinite. It is also reasonable to assume that it is finite because, in real-life applications, messages are countable even though the number of definable instances is infinite. Based on this, Assumption 3.2 and Definition 3.1 are introduced as follows.

Algorithm 3: User-oriented algorithm of DRACO. A pseudo-algorithm for source code reproduction is provided in Appendix B.3.

```
Input: $\eta, \theta_0, B, T, P$
Output: $\{\theta_t : \forall t\}$
for $i = 1, \cdots, N$ do in parallel
    while $t < T$ do
        $t \leftarrow$ clock()
        if there is an event at time $t$ then
            if grad computation step then
                $\mathbf{y}_0^{(i)} \leftarrow \theta^{(i)}$
                for $b = 0, \cdots, B-1$ do
                    $\mathbf{y}_{b+1}^{(i)} \leftarrow \mathbf{y}_b^{(i)} - \eta g_i(\mathbf{y}_b^{(i)})$
                end for
                $\Delta^{(i)} \leftarrow \mathbf{y}_B^{(i)} - \theta^{(i)}$                // local batch training
            else if transmission step then
                $i$ sends $\Delta^{(i)}$ to its neighbors
                for $j \in \mathcal{N}(i)$ do
                    $j$ receives $\tilde{\Delta}^{(i)}$
                    $\theta^{(j)} \leftarrow \theta^{(j)} + \sum_{j \neq i} q_t^{ij} \tilde{\Delta}^{(i)}$        // aggregation
                end for
            end if
        end if
        if $t \equiv 0$ (mod $P$) and $t > 0$ and $i$ is the hub at $t$ then
            $i$ broadcasts $\theta^{(i)}$
            for $j \in \mathcal{U} \setminus \{i\}$ do
                $j$ receives $\tilde{\theta}^{(i)}$
                $\theta^{(j)} \leftarrow \tilde{\theta}^{(i)}$                      // unification
            end for
        end if
    end while
end for
return $\left(\theta_T^{(i)}\right)_{1 \leq i \leq N}$
```

Assumption 3.2. *(Finite number of messages during a unit time period) During every period P, the number of messages that each user receives is finite.*

Definition 3.1. *(Maximum number of receiving messages per user) Let $\psi_i(t_{start}, t_{end})$ indicate the function that counts the number of messages arrived at user i since time t_{start} until time t_{end}. For any $i \in \mathcal{U}$ and $m \in [0, 1, \cdots, \lfloor \frac{T}{P} \rfloor - 1]$,*

$$\psi_i(mP, (m+1)P) \leq \Psi \ ,$$

where Ψ is the maximum number of messages that a user permits to receive during time duration $[mP, (m+1)P)$.

The Ψ term not only justifies the number of messages to be countable but also functions as a communication budget per period. Interestingly, when a decentralized network has a fixed communication budget per unit time, performing many consensus steps can effectively reduce the error even though each gossiping step renders low precision. [120]

3.4 Convergence Analysis

In this section, we analyze the convergence performance of DRACO. For that, following the common practice in the literature, we make the subsequent assumptions along with the objective function.

Assumption 3.3. *(Lipschitz gradient.) For any $\theta, \mathbf{y} \in \mathbb{R}^d$ and for any $i \in \mathcal{U}$, there is a nonnegative L that satisfies*

$$\|\nabla f_i(\theta) - \nabla f_i(\mathbf{y})\| \leq L\|\theta - \mathbf{y}\|, \ \forall \theta, \mathbf{y}, \ i. \tag{3.5}$$

Assumption 3.4. *(Unbiased stochastic gradient with bounded variance.) For all $\theta, \ i$,*

$$\mathbb{E}[g_i(\theta)|\theta] = \nabla f_i(\theta) \text{ and } \mathbb{E}\left[\|g_i(\theta) - \nabla f_i(\theta)\|^2 |\theta\right] \leq \sigma^2 \tag{3.6}$$

Assumption 3.5. *(Bounded gradient divergence.) For all $t \in [0, T)$ and $i \in \mathcal{U}$, the gradient divergence is bounded by ζ, i.e.,*

$$\|\nabla f_i(\theta_t^{(i)}) - \nabla f(\theta_t)\|^2 \leq \zeta^2. \tag{3.7}$$

From Assumption 3.5, an alternative deviation of local gradients is derived as in Lemma 3.1.

Lemma 3.1. *(Deviation of local gradients) When $N > 4$, for all $\theta, \ t$,*

$$\left\|\sum_{j\in\mathcal{U}} q_t^{j\to i}[\nabla f_i(\theta_t^{(i)}) - \nabla f_j(\theta_t^{(j)})]\right\|^2 \leq \frac{2N\zeta^2}{N-4}.$$

Proof. The left side of the inequality above can be rephrased as

$$\left\|\sum_{j\in\mathcal{U}} q_t^{j\to i}[\nabla f_i(\theta_t^{(i)}) - \nabla f_j(\theta_t^{(j)})]\right\|^2 = \left\|\nabla f_i(\theta_t^{(i)}) - \sum_{j\in\mathcal{U}} q_t^{j\to i}\nabla f_j(\theta_t^{(j)})\right\|^2.$$

By adding and subtracting $\nabla f(\theta_t)$, we have

$$\left\|\nabla f_i(\theta_t^{(i)}) - \sum_{j\in\mathcal{U}} q_t^{j\to i}\nabla f_j(\theta_t^{(j)})\right\|^2$$

$$
\begin{aligned}
&= \left\| \nabla f_i(\theta_t^{(i)}) - \nabla f(\theta_t) + \nabla f(\theta_t) - \sum_{j=1}^{N} q_t^{j\to i} \nabla f_j(\theta_t^{(j)}) \right\|^2 \\
&= \left\| \nabla f_i(\theta_t^{(i)}) - \nabla f(\theta_t) + \frac{1}{N}\sum_{i'=1}^{N} \nabla f_{i'}(\theta_t^{(i')}) - \frac{1}{N}\sum_{i'=1}^{N}\sum_{j=1}^{N} q_t^{j\to i} \nabla f_j(\theta_t^{(j)}) \right\|^2 \\
&\overset{(a)}{\leq} 2\left\| \nabla f_i(\theta_t^{(i)}) - \nabla f(\theta_t) \right\|^2 + 2\left\| \frac{1}{N}\sum_{i'=1}^{N} \Big[\nabla f_{i'}(\theta_t^{(i')}) - \sum_{j=1}^{N} q_t^{j\to i} \nabla f_j(\theta_t^{(j)}) \Big] \right\|^2 \\
&\overset{(b)}{\leq} 2\zeta^2 + \frac{2}{N}\sum_{i'=1}^{N} \left\| \nabla f_{i'}(\theta_t^{(i')}) - \sum_{j=1}^{N} q_t^{j\to i} \nabla f_j(\theta_t^{(j)}) \right\|^2,
\end{aligned}
$$

where (a) uses $(\|\mathbf{z}_1 + \mathbf{z}_2\|^2)/2 \leq \|\mathbf{z}_1\|^2 + \|\mathbf{z}_2\|^2$; (b) is from the definition of ζ^2 in Assumption 3.5 on the first term and Jensen's inequality on the second term. By rearranging the second term of the right side of the inequality, we get

$$
\begin{aligned}
&\Big(1 - \frac{2}{N}\Big) \left\| \nabla f_i(\theta_t^{(i)}) - \sum_{j\in\mathcal{U}} q_t^{j\to i} \nabla f_j(\theta_t^{(j)}) \right\|^2 \\
&\leq 2\zeta^2 + \frac{2}{N}\sum_{i'\in\mathcal{U}\setminus\{i\}} \left\| \nabla f_{i'}(\theta_t^{(i')}) - \sum_{j=1}^{N} q_t^{j\to i} \nabla f_j(\theta_t^{(j)}) \right\|^2 \\
&\leq 2\zeta^2 + \frac{2}{N}\sum_{i'=1}^{N} \left\| \nabla f_{i'}(\theta_t^{(i')}) - \sum_{j=1}^{N} q_t^{j\to i'} \nabla f_j(\theta_t^{(j)}) \right\|^2.
\end{aligned}
$$

With another rearrangement to the left side, the inequality becomes

$$
\Big(1 - \frac{4}{N}\Big) \left\| \nabla f_i(\theta_t^{(i)}) - \sum_{j\in\mathcal{U}} q_t^{j\to i} \nabla f_j(\theta_t^{(j)}) \right\|^2 \leq 2\zeta^2.
$$

□

Considering all the above assumptions, we obtain an upper bound on the expectation of the original objective's gradient when $\mathcal{Q}$ is given in advance.

Theorem 3.1. *Let $\mathcal{F} := f(\theta_0) - \min_\theta f(\theta)$. Under all the aforementioned assumptions, we have*

$$
\min_t \mathbb{E}\big[\|\nabla f(\theta_t)\|^2 \big| \mathcal{Q}\big] \leq \mathcal{O}\Big(\frac{\mathcal{F}}{B\eta\Psi} + \frac{\zeta^2}{N-4} + \sigma^2 + N\zeta^2 + BL^2\eta^2\sigma^2 + \frac{L\eta\rho^2\sigma^2}{N\Psi}\Big) \tag{3.8}
$$

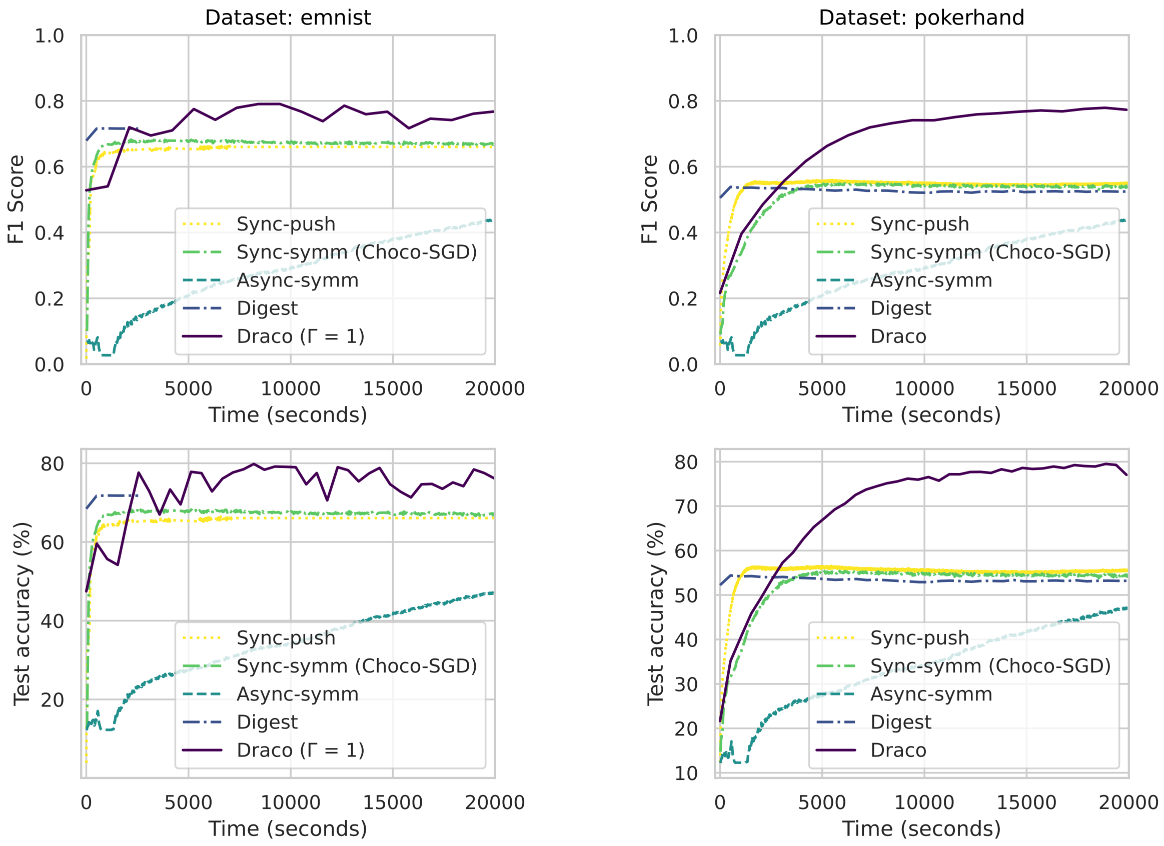
for $\eta \leq \frac{1}{8BLN\Psi}$, $N > 4$, and $\Psi \geq 3$.

Remark. We begin, following a similar approach to [94], by deriving an inequality rooted in the smoothness of f_i. This inequality establishes a connection between two local losses from the same user at different timestamps, namely $f_i(\theta_{t_0+P}^{(i)})$ and $f_i(\theta_{t_0}^{(i)})$. Within this inequality, an inner product term unfolds into several components. Notably, it comprises three distinct subterms: one involving $||\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)}||^2$ (refer to Lemma B.2), another featuring $||\theta_t^{(j)} - \theta_{t_0}^{(j)}||^2$ (see Lemma B.3) which is mainly derived from Algorithm 3, and a third term with $||\nabla f_i(\theta_t^{(i)}) - \nabla f_j(\theta_t^{(j)})||^2$, of which the expectation has an upper bound (refer to Lemma 3.1). Our proof is novel in the sense that it effectively converts and simplifies the terms on the continuous timeline into discrete values. Detailed proof is available in Appendix B.1.4. □

3.5 Experimental Results

We conducted experiments with federated learning on two datasets: (1) balanced EMNIST [121] dataset with 47 class labels for image classification tasks, and (2) the Poker hand dataset [122] for multi-class classification tasks, which is widely applied in automatic rule generation. Each user possesses 1000 local training samples arranged into training batches with 64 samples per batch. The default number of participants in each simulation is $N = 25$, otherwise it is specified accordingly. The sampling interval is 500 events, i.e., the evaluation of each local model is done under a test set whenever the 500th event is finished. The rate parameter of exponential distribution in local gradient computation is $\lambda_i = 0.1$ for all users by default. In this study, the impact on model compression is not evaluated, implying that the packet size is as large as the raw model. The convolutional neural network (CNN) architecture used in the simulations takes up 596776 B (0.57 MB) for feeding samples from EMNIST, and 51640 B (0.05 MB) from Poker hand, respectively. This value is used to quantify the message size.

We performed simulations using two topologies: cycle and complete, with a time-invariant $\mathcal{Q}$. The connectivity graph is fixed throughout the whole collaboration process. Each user, indexed i without losing generality, spends some time computing local gradient following $exp(\lambda_i)$ as mentioned in Assumption 3.1. Whenever a local update is done at t, user i sends $\Delta_t^{(i)}$ to its neighbors $j \in \mathcal{N}(i)$, where $\mathcal{N}(i)$ indicates a set of user i's neighbors. Although a pre-defined topology outlines the intended communication paths between nodes, the inherent unreliability of wireless channels can significantly affect data transmission. Factors such as fading, interference, and physical obstructions can disrupt connectivity, resulting in packet losses and delays, thereby undermining the efficiency and reliability of communication within the network. We used parameters reported in [123] and [124] for the wireless communication settings. The radius of the field where the nodes can be scattered is $R = 500$ m. We fix the transmit power of each user as $P_i = 30$ dBm (1000 mW). We also set the path loss exponent $\alpha = 4$, the bandwidth $W = 10$ MHz, and the noise power density $N_0 = -174$ dBm/Hz. We assumed that two nodes interfere with each other during transmission if their distance is closer than $0.1R$. Due to those wireless communication characteristics, the mechanism for realizing DRACO is slightly differ-



(a) Impact on topology

(b) Impact on Ψ

Figure 3.4. Performance comparison with the literature under (a) EMNIST dataset, and (b) Poker hand dataset.

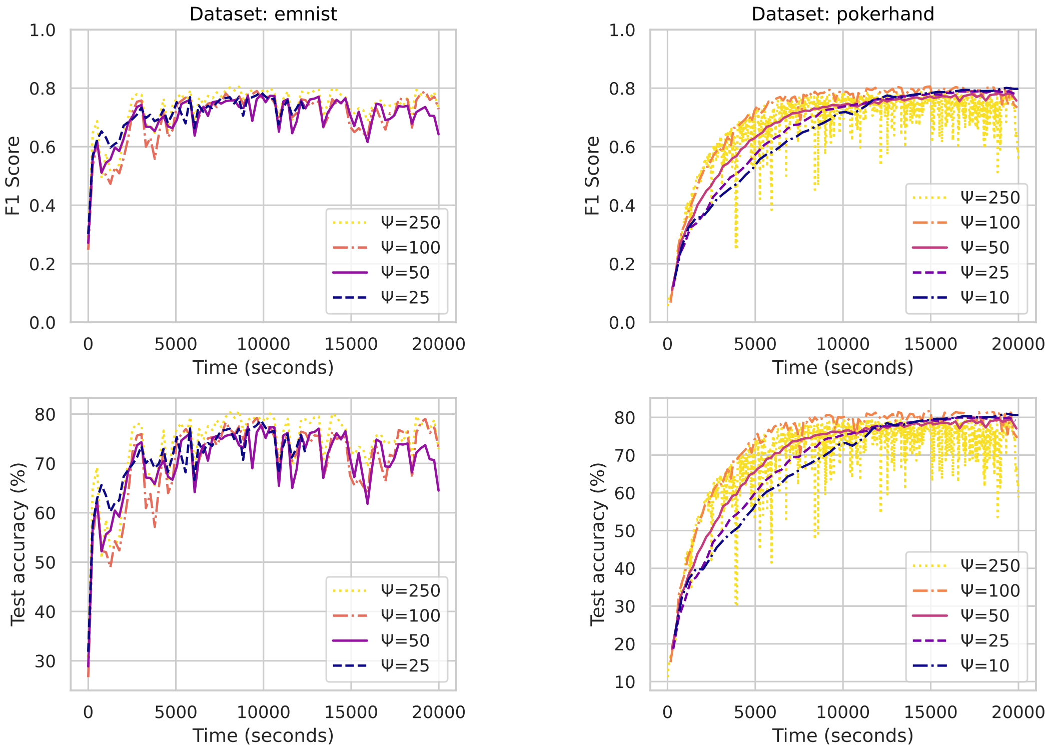
ent. Specifically, when user i has performed local training at time t, it broadcasts its update $\Delta_t^{(i)}$ to all $j \in \mathcal{U} \setminus \{i\}$. It takes

$$\Gamma_{i \to j} = \frac{\text{message size}}{W \cdot \log_2(1 + \text{SINR}_{i,j})} + \frac{\text{distance}(i,j)}{\text{lightspeed}}$$

seconds for the message to arrive at node j. Here, the signal-to-interference-plus-noise ratio (SINR) between the two nodes is defined as

$$\text{SINR}_{i,j} = \frac{P_i h_{ji} \text{distance}(j,i)^{-\alpha}}{\sum_{n \in \Phi_j} P_i h_{jn} \text{distance}(j,n)^{-\alpha} + z^2} ,$$

where $h_{ji} \sim exp(1)$ denotes the small-scale fading gain, Φ_j is a set of nodes interfering node j, and z^2 characterizes the variance of AWGN (Additive White Gaussian Noise). As long as the transmission duration $\Gamma_{i \to j}$ is shorter than the predetermined threshold $\Gamma_{\max}$, user j succeeds to receive $\Delta^{(i)}$ at time $t + \Gamma_{i \to j}$. (i.e., $q_{t+\Gamma_{i \to j}}^{i \to j} = 1$.)



(a)

(b)

Figure 3.5. Results for different upper bounds on the number of received messages per user. ($\Gamma_{\text{max}} = 10$)

The performance of DRACO is evaluated across different network topologies and datasets. For EMNIST, a cycle topology is employed, where each user is connected to two neighbors. In contrast, the Poker hand dataset utilizes a fully connected topology, with each user directly connected to all others. DRACO's performance is compared against four benchmark methods:

- sync-symm: Synchronous learning with symmetric connectivity (Choco-SGD [8])
- sync-push: Synchronous learning with directed connectivity.
- async-symm: Asynchronous learning with symmetric connectivity (Decentralized Asynchronous SGD [52]).
- async-push: Asynchronous learning with directed connectivity (Digest [114]).

The term "Push" denotes the use of the push-sum algorithm for directed graphs.

The Poker hand dataset presents a unique challenge due to its imbalanced class distribution. To comprehensively assess model performance, both test accuracy and F1-score were evaluated, the latter accounting for both precision and recall.

While the choice of dataset had a minor impact on overall trends, the network topology significantly influenced performance. In the cycle topology, where each user exchanges information with only two neighbors, unreliable channels (e.g., due to fading) can lead to frequent client isolation. As shown in Fig. 3.4a, synchronous methods exhibited comparable performance, but async-symm underperformed async-push, despite using a doubly stochastic matrix. This highlights the sensitivity of async-symm to strict transmission deadlines, emphasizing the importance of well-designed scheduling in asynchronous learning.

In the fully connected topology in Fig. 3.4b, where every user is connected to all others, the virtual global model can be trained more robustly, even when some edges are intermittently disrupted. While convergence speeds vary, all algorithms ultimately achieve similar performance.

DRACO consistently outperformed competitors in both test accuracy and F1-score. This advantage stems from its parallel aggregation and unification mechanisms, which effectively mitigate the divergence of local models common in asynchronous decentralized learning. DRACO periodically unifies local reference models and regulates the number of received messages, enhancing robustness in continuous operation and fading environments.

During implementation, performance oscillations were observed when users received excessive redundant updates due to high transmission frequencies (large Ψ values in Fig. 3.5a and 3.5b). Conversely, excessively small Ψ values slowed learning by limiting crucial updates' reception. These findings align with prior work [120] and the theoretical analysis presented in Theorem 3.1.

3.6 Concluding Remarks

We have studied decentralized asynchronous learning optimization through row-stochastic gossip communication networks and proposed a novel method termed DRACO. By facilitating the learning process obviating the need for global iteration counts, our technique presents local user performance defined on a continuous timeline. We provided practical instructions for each participant by decoupling training and transmission schedules, resulting in complete autonomy and simplified implementations in real-world applications. We analyzed the algorithm convergence and provided experimental results that support the efficacy and feasibility of the proposed framework.

In the remainder, we highlight some promising yet challenging directions that require further investigation.

- **Bandwidth allocation.** In this chapter, bandwidth is equally distributed to all users. If the users exchange their SINR information, as well as their weight updates, a bandwidth allocation algorithm can be added within the "for *i*" loop, as proposed in [124].

- **More realistic experiments with aggregation time threshold.** We can consider that each user has a predetermined threshold to aggregate its neighbors' local updates. The user can perform superposition to its local reference model only after the timeout occurs.

For instance, each user j might have an upper bound on the number of $\Delta^{(i)}$'s that it can accept during its receiving period.

- **Improve robustness against collisions.** Random access is known to have a higher probability of collision occurrence. However, it is cumbersome or impractical to predetermine the communication schedule because while carrying out DRACO, the participants decide whether to transmit and/or train their local models without communication or agreement with the other users. Collision in a random access protocol, such as in the context of federated learning where clients transmit messages, can be alleviated by adapting classical approaches in wireless networks. These approaches include configuring a random backoff time after a collision for retransmission attempts, adopting collision detection mechanisms, or allowing clients to dynamically adjust the size of their messages or the transmission power. On the other hand, considering collisions from the resource allocation perspective, the system can assign different priority levels to clients based on factors, such as their data urgency or historical collision rates.
- **Reception control** We manually selected the rate parameters for transmissions ($\lambda_{j\to i}$) because we assumed that the participants are not able to predict the frequency of message-receiving events, even in fixed Q cases. Nevertheless, there exist techniques that enable edge devices to roughly estimate in advance the ratio of successful message reception. With this in mind, it will be possible to study how to manage the reception events in realizing DRACO.

4 Personalized Decentralized Federated Learning with Knowledge Distillation

Personalization in FL functions as a coordinator for clients with high variance in data or behavior. Ensuring the convergence of these clients' models relies on how closely users collaborate with those with similar patterns or preferences. However, it is generally challenging to quantify similarity under limited knowledge about other users' models given to users in a decentralized network. To cope with this issue, we propose a personalized and fully decentralized FL algorithm, leveraging knowledge distillation techniques to empower each device to discern statistical distances between local models. Each client device can enhance its performance without sharing local data by estimating the similarity between two intermediate outputs from feeding local samples as in knowledge distillation. Our empirical studies demonstrate that the proposed algorithm improves the test accuracy of clients in fewer iterations under highly non-i.i.d. data distributions and is beneficial to agents with small datasets, even without the need for a central server.

4.1 Introduction

Since the appearance of FL [1] as a promising and efficient solution for distributed learning with collaborative clients, numerous research studies have investigated this paradigm in distributed networks of users under various hindrance factors, such as limited local storage [125], information leakage [126], biases across user experiences [127], and transmission impairments [128]. The main objective of FL and of many of its decentralized variants is generally to acquire a global model across all devices. Nevertheless, a single common model deduced from all participants may not satisfy the clients whose tasks or data distributions significantly deviate from the rest. On this account, personalized FL [61]–[63], [129]–[132] has been considered as a means to provide a customized solution to users with statistical heterogeneity. A widely used procedure for personalized FL first constructs a global model using a central aggregator as a draft and then customizes it under each agent's control. [61]–[63] On the other hand,

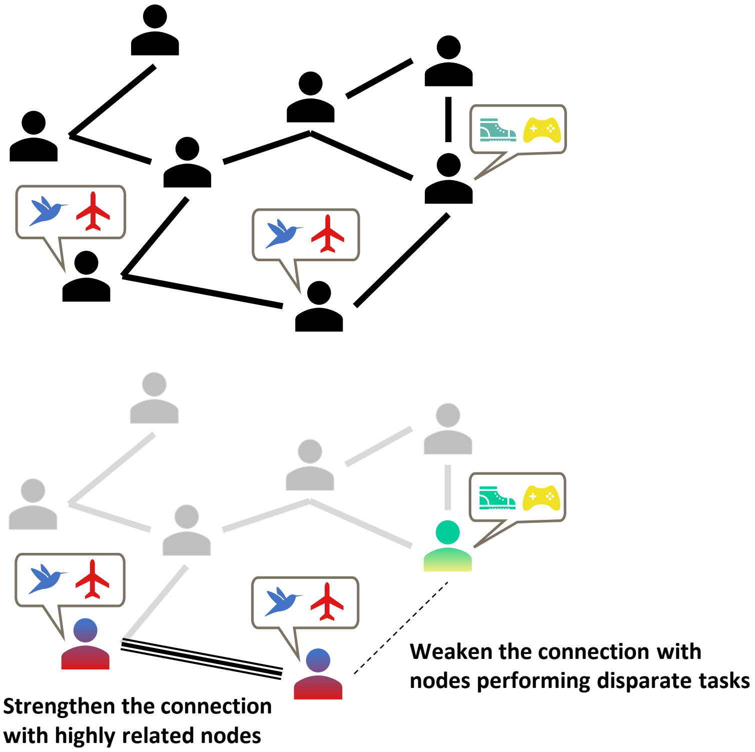

Weaken the connection with
nodes performing disparate tasks
Strengthen the connection
with highly related nodes

Figure 4.1. Challenges in personalized FL with peer-to-peer communication include privacy concerns from sharing data distribution indicators and the difficulty in selecting appropriate distance metrics that balance information exchange complexity and accuracy.

personalized FL is particularly commensurate with serverless networks as each agent executes the training process autonomously, thereby enabling asynchronous learning [6], [133]–[138].

Despite its practical relevance, there are several issues and technical challenges associated to personalized federated learning, which we briefly discuss below. First, among users with diverse characteristics, each agent has to find sufficiently similar peers since putting more weights on received model parameters with higher similarity during superposition improves its performance [138]. Unfortunately, choosing the right distance measurement, which can capture the actual similarity, is challenging. Moreover, the system has to encounter a tradeoff between model complexity and metric complexity. If the exchanged information has a more complex structure, participants have no choice but to use simpler metrics [139].

Second, FL fundamentally requires ensuring privacy within each agent, which commonly implies but is not limited to maintaining data samples private. For instance, the information on possessed class labels for classification tasks should also be kept private [140]. However, because of their omniscient viewpoint, most existing personalized and decentralized learning schemes do not thoroughly preserve inter-device privacy. Occasionally, the agents take a constant for local training, which is derived from public knowledge, e.g., the number of others' training samples [133]. Even though the variables used for training are perfectly separated, a decentralized collaborative learning scheme introduced in [134] assumes the presence of a proxy dataset accessible to anyone. In [6], [129], the selected agents request to exchange hypotheses represented as a weighted sum of base data distributions. These identify the direct information of which class labels one possesses from its neighbors.

To address the aforementioned challenges, we propose KD-PDFL [54], a personalized decentralized FL scheme that provides a completely enclosed service. With KD-PDFL, each client can individually update all parameters from their first-person perspective, including the connectivity graph, the local model, and the local dataset. In this approach, each user receives only model copies from its neighbors and determines their optimal combination. This property differentiates our algorithm from prior works, which either ask users to seek additional external information or rely on isolated personalization methods like local fine-tuning.

As a powerful tool that enhances the inference capability of clients with simple models, we introduce knowledge distillation into our proposed scheme, where agents evaluate similarity with co-distillation based on local validation datasets. Thanks to embracing the characteristic of distillation, agents are also free from the need for model homogeneity since distillation enables cooperation across models with different layer structures as long as the models have a common layer with the same dimension. Our experimental results show that KD-PDFL achieves higher test accuracy within smaller global iterations compared to other personalized decentralized FL schemes, given that the amount of exchanged information is the same. We also provide a guideline for tuning the hyperparameters used in implementing our experiments.

4.2 Preliminaries

In this section, before introducing our work, we provide a brief overview of the two major ingredients of our proposed solution: (i) personalized decentralized learning, which describes the overall protocol of how users exchange information; (ii) knowledge distillation, which elaborates on how they draw relevance from others.

4.2.1 Personalized decentralized learning

In contrast to distributed networks with a central server, a fully decentralized network implies that no node has the authority or accessibility to construct a global consensus model at any instant of the learning process. Thus, personalized and decentralized FL naturally exclude building a common model before local customization. We consider a decentralized learning system that assigns different central entities for each global iteration. In this network, a randomly selected node wakes up to serve as a temporary center node, which is also called a “star node”. First, this star node collects gradient parameters from its neighbors. Subsequently, it calculates inferences from all received gradients, which are used to measure the similarities

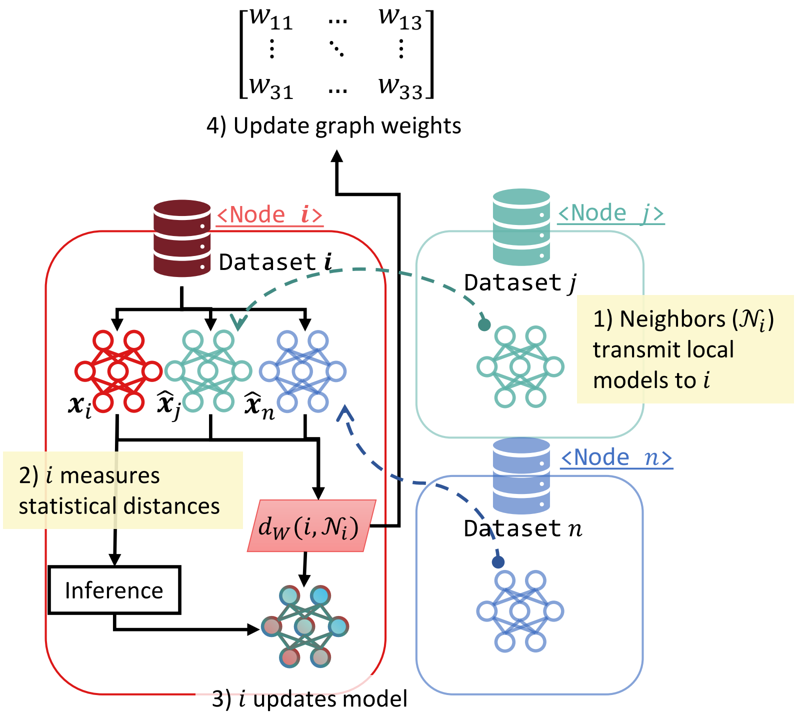

$\begin{bmatrix} w_{11} & \cdots & w_{13} \\ \vdots & \ddots & \vdots \\ w_{31} & \cdots & w_{33} \end{bmatrix}$
4) Update graph weights
<Node i>
Dataset i
<Node j>
Dataset j
1) Neighbors ($\mathcal{N}_i$) transmit local models to i
x_i
$\hat{x}_j$
$\hat{x}_n$
2) i measures statistical distances
$d_W(i, \mathcal{N}_i)$
<Node n>
Dataset n
Inference
3) i updates model

Figure 4.2. A general schematic of personalized decentralized FL.

among the learning objectives thereafter. Users manage the traits of these similarities by selecting and focusing on communicating with neighbors that have the highest similarities. In literature, these semantic traits are modeled using a *collaboration graph*, which can be a property of communication links between nodes or a target variable to optimize mixing weights. A collaboration graph is a weighted graph whose edge weights represent quantified relevance between learning tasks. (See also Fig. 4.2)

Once the star node finishes the calculation described above, it updates the weights of the collaboration graph according to the similarity variables. Meanwhile, it also updates its model using a weighted sum of all received parameters, where the weights are those obtained from the collaboration graph.

4.2.2 Knowledge Distillation and Co-Distillation

Knowledge Distillation (KD) [141] is a knowledge transfer method in which multiple clients can compare the outputs from a common dataset to absorb the inferred knowledge of the others. They quantify the similarity between the logits. In Co-Distillation (CD) [142], all clients are in an equal position to learn from the other as everyone works as a student. KD and CD have the advantage of allowing the users to transfer knowledge between models with heterogeneous structures. Yet, the system must have a public/common dataset in order to let the participants compare each logit one by one with identical batches. This conflicts with the vanilla FL approach that guarantees the data maintenance attribute without transmission of data samples.

4.3 Problem Setting

We consider a joint minimization problem across N users. Each user indicated as $i \in \mathcal{U} = \{1, 2, \cdots, N\}$ has access to a training set $\mathcal{D}_i$ and its goal is to minimize its loss function $f_i : \mathbb{R}^{n_\mathcal{X}} \rightarrow \mathbb{R}^+$ where $n_\mathcal{X}$ is the dimension of the input features. The objective of the whole system is to minimize the total local loss and the jointly calculated dissimilarity.

We adapt the idea of a collaboration graph W to represent the connectivity between any two nodes in the network. A collaboration graph $W = [w_1, w_2, \ldots, w_N]$ is a matrix of stacked connectivity vectors of each client. Each column vector of W, denoted as $w_i = [w_{i1}, \ldots, w_{iN}]^T$, is a connectivity vector of client i whose elements are the edge weights. In this chapter, the term "connectivity" implies relevance across two nodes, thus $w_{ij} \geq 0$ is proportional to the degree of recognition of node i for how relevant node j's task is to its own task.

Objective function: The main target of our system is to learn the personalized models $\Theta = \{\theta_1, ..., \theta_N\}$ and the collaboration graph $W \in \mathbb{R}^{N \times N}$ that minimize the following joint optimization problem

$$\min_{\Theta, W} J(\Theta, W) = \sum_{i=1}^{N} f_i(\theta_i; \mathcal{D}_i) + \frac{\mu_1}{2} \sum_{i,j \in \mathcal{U}} w_{ij} d_w(i,j) + \mu_2 g(w) \tag{4.1}$$

where $f_i(\cdot;\cdot)$ is a local loss function and $d_w(i,j)$ is the measured distance (dissimilarity) between the model estimations of two clients i and j. The second term allows assessing task relevance by penalizing the links between any two nodes with large statistical distances. The third term $g(w)$ is a regularization term that strongly encourages the users to participate in collaboration by giving a high penalty when a user tries only local training. μ_1 and μ_2 are hyperparameters for adjusting the influence of each term above, respectively.

4.4 Personalized Decentralized Learning with KD

We consider a fully decentralized network where agents do not implement strict synchronization and cooperate through peer-to-peer communication. Particularly, our definition of decentralization imparts autonomy in determining the collaboration graph. In other words, each user evaluates the collaborative weights independently and privately instead of accessing a row of a connectivity matrix shared in public. The clients follow a cross-silo setting where each client performs all steps of the learning process, i.e., has datasets locally distributed for training, validation, and test purposes. In this section, we focus on the method to extract relevant information across the nodes in a serverless fashion. In every exchange interval T_{ex}, the agents follow the step-by-step instructions below (see also Fig. 4.3):

1. A node assigned as a star node, say user i, wakes up to ask for transmission from a group of its neighbors at time t, $\mathcal{N}_i^{(t)}$. The peers send their local model updates to the star node i. At the end of the transmission, i has $|\mathcal{N}_i^{(t)}|$-copies of model parameters of its neighbors if no packet loss has occurred.

2. From i's local training dataset $\mathcal{D}_i$, i packs training samples in a batch B_i. The batch B_i is fed to model parameters of all $j \in \mathcal{N}_i^{(t)}$ in order to get intermediate outputs (e.g., logits), denoted as $\mathbf{z}_{ij}$.

3. Node i measures the statistical distance $d_W(i,j)$ for all $j \in \mathcal{N}_i^{(t)}$, then updates w_i of which the gradient function $\nabla J_i(w)$ is the partial derivative of Eq. (4.1) with respect to the computation entity i. Based on this gradient, node i updates a collaboration weight of j from the viewpoint of i, which is always bounded to a nonnegative value. The following equation illustrates the policy of updating the connectivity vector of i:

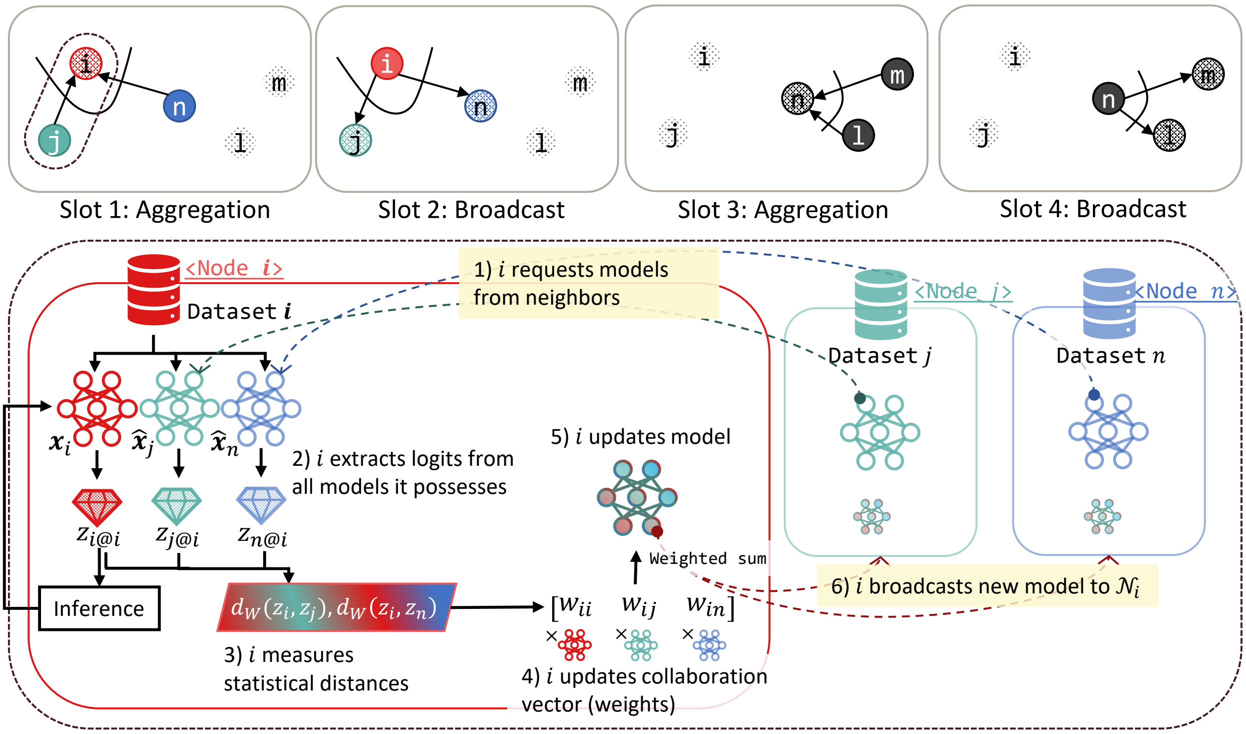
$$\begin{aligned}
&\nabla J_i(w_i^{(t)}) = \mu_1 d_{W,i} + \mu_2 \nabla g_i(w_i^{(t)}) \\
&\beta^{(t)} = 1./|\nabla J_i(w_i^{(t)})| \\
&w_{ij}^{(t+1)} = \max(0, w_{ij}^{(t)} - \beta^{(t)} \nabla J_i(w_i^{(t)}))
\end{aligned} \tag{4.2}$$

   where $d_{W,i} = [d_W(i,1), \cdots, d_W(i,N)]$ indicates the statistical distance vector of node i.

4. i updates its connectivity vector using $d_W(i,j)$. To quantify the distance between two logits that are formed as probability distributions, the given user calculates the batched mean of the Wasserstein distance between the two variables. In particular, under classification tasks where the probability distribution is discrete and the number of possible classes is known, a user can compute the arithmetic mean of all Wasserstein distances from each single data sample of the batch. With n_i local batch samples and n_L possible classification labels, the Wasserstein distance of two logits with $(n_i \times n_L)$ size is computed as follows:

$$d_W(i,j) = \frac{1}{n_i} \sum_{x=1}^{n_i} \sum_{l=1}^{n_L} \|p_{i,l}^{(x)} - p_{j,l}^{(x)}\|_2^2 \tag{4.3}$$

   where $p_{i,l}^{(x)}$ indicates the logit for class label l that goes through user i's model from an input data point with index x. Note that the importance of i from the viewpoint of j may differ from that of j from the viewpoint of i due to differences in model complexity or



Slot 1: Aggregation
Slot 2: Broadcast
Slot 3: Aggregation
Slot 4: Broadcast
i
j
n
m
l
<Node i>
Dataset i
1) i requests models from neighbors
<Node j>
<Node n>
Dataset j
Dataset n
x_i
$\hat{x}_j$
$\hat{x}_n$
2) i extracts logits from all models it possesses
5) i updates model
$z_{i@i}$
$z_{j@i}$
$z_{n@i}$
Weighted sum
Inference
$d_W(z_i, z_j), d_W(z_i, z_n)$
$[w_{ii}$ w_{ij} $w_{in}]$
6) i broadcasts new model to $\mathcal{N}_i$
3) i measures statistical distances
4) i updates collaboration vector (weights)

Figure 4.3. Above: communication protocol during exchange intervals. Below: model and collaboration vector update elaborated. The figure illustrates the process executed only in the star node.

the number of trainable samples. Thus, W is not symmetric, and agents do not need to share updated connectivity weights.

5. A new model for i is a weighted sum of all footprints it has at that time, where the weights are the elements of the connectivity vector.

6. In the broadcast phase at the next time slot, node i broadcasts $\theta_i^{(t)}$ to the peer group of the previous time slot, $\mathcal{N}_i^{(t-1)}$.

7. Other than the exchanging iterations, each agent performs only local learning.

One cannot find its own weight w_{ii} of the connectivity vector since having statistical distance with itself does not make sense. For that, we adopt a widely used concept named confidence to indicate the weight of local model updates. A confidence of node i at time t, denoted as c_i, is defined as a time-varying term dependent on the size of its local training set:

$$c_i^{(t)} = \min\left(\frac{|\mathcal{D}_i|}{c_{base}}, \frac{1}{|\mathcal{N}_i^{(t)}|+1}\right) \tag{4.4}$$

where c_{base} is a constant for a base confidence that is neither induced from shared information across the devices nor to be shared with others.

The function `Midgetter` in Alg. (4) extracts outputs from the intermediate layers of two neural networks. The intermediate layer can be either the output layer or one of the hidden layers depending on whether the agents transmit the entire models or only the base layers. Feeding the identical batch of client i's training samples, θ_i and θ_j returns the logits $\mathbf{z}_{i@i}$ and $\mathbf{z}_{j@i}$, respectively. The expression $a@b$ refers to an output of client a's model under control of client b, i.e., the distillation is conducted at user b's device while using b's computation resources without sharing.

Algorithm 4: KD-PDFL: Distillation-based personalized decentralized FL

```
Input: $\theta_i^{(0)} = \mathbf{0} \in \mathbb{R}^{n_i}$, $w_{ii} = 0 \ \forall i \in \mathcal{U} = \{1, 2, ..., N\}$, $w_{ij} = 1/N \ \forall j \in \mathcal{U} \setminus i$
Output: $\Theta^{(R)}$, $W^{(R)}$
for $r$ in $(0, R]$ do
    if $r \equiv 0 \pmod{T_{ex}}$ then
        Random user $i$ wakes up & draw a subset $\mathcal{N}_i^{(r)}$
        for each neighbor $j \in \mathcal{N}_i^{(r)}$ do in parallel
            Receive $\tilde{\theta}_j^{(r-1)}$ from each $j$
            $(\mathbf{z}_{i@i}, \mathbf{z}_{j@i}) =$ MidGetter($\theta_i, \tilde{\theta}_j, \mathcal{D}_i$)    // find intermediate outputs
            $d_W(i, j) =$ Wasserstein2D($\mathbf{z}_{i@i}, \mathbf{z}_{j@i}$)    // find statistic distances
        end for
        ConnVectorUpdate($i, j$) as in Equation (4.2)
        Update $\theta_i^{(r+1)} = \sum_{j \in \mathcal{U} \setminus \{i\}} w_{ij}^{(r+1)} \tilde{\theta}_j^{(r-1)} + c_i w_{ii}^{(r+1)} \theta_i^{(r)}$
    else if $r \equiv 1 \pmod{T_{ex}}$ then
        for each neighbor $j \in \mathcal{N}_i^{(r-1)}$ do
            Receive $\tilde{\theta}_i^{(r-1)}$ from $i$
            Update $\theta_j^{(r+1)} = \tilde{\theta}_i^{(r)}$
        end for
    else
        $\theta_i^{(r+1)} = \theta_i^{(r)} - \eta \nabla f_i(\theta_i^{(r)})$    // Local Update
    end if
end for
```

4.5 Experiments

In this section, we evaluate the performance of KD-PDFL in terms of per-client test accuracy. We compare the proposed scheme with three related baseline scenarios for decentralized

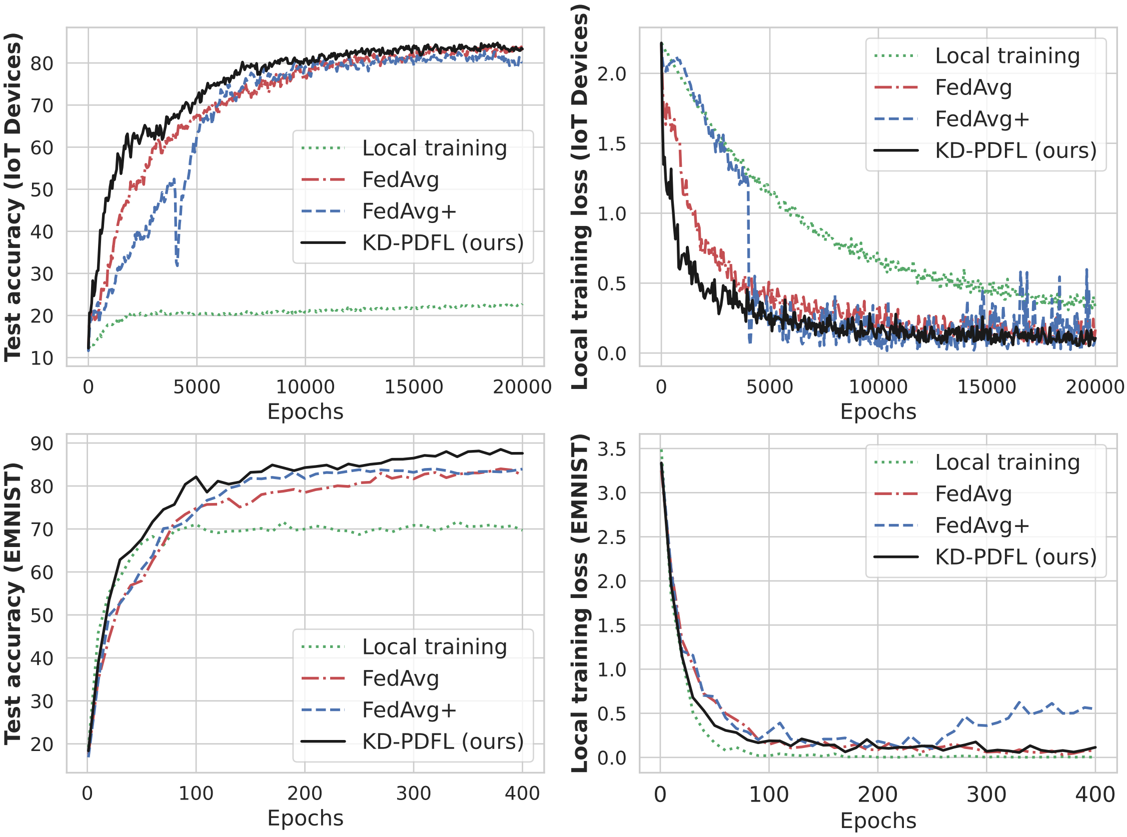


Figure 4.4. Learning progress of standalone and collaborative users in different collaboration methods over iterations ($N = 40$).

networks: (i) standalone case where all users perform local training only; (ii) conventional federated averaging (FedAvg [2]), which returns the arithmetic mean of received model parameters; and (iii) FedAvg followed by fine-tuning, which switches the algorithm into Reptile [143]. (FedAvg+ [62]) Note that in our evaluations, FedAvg corresponds to a decentralized learning scheme but with non-personalized service, i.e., it aims to build a single global model for all clients.

We carry out the experiments with two types of datasets. The first one we consider is a set of smart users practicing classification tasks using web traffic information from IoT devices. The data samples have 296 input features and the output dimension of the dataset is 9.[1] Each user contains 15 to 100 training samples with non-i.i.d. distribution and 100 local test samples, which follows the same distribution with the local training set. We split the IoT devices training dataset into data points of which the class labels follow a symmetric Dirichlet distribution with parameter 0.1. This data split is the same as in a benchmark in [144] but is more biased due to the smaller parameter, allocating one to four class labels per each user with uneven number of data samples per label. A neural network model in every user includes one batch

[1] https://www.kaggle.com/datasets/fanbyprinciple/iot-device-identification

Dataset	Methods	N=10	N=20	N=40
IoT devices	Local learning		0.210±0.089	
	FedAvg	0.759±0.650	0.643±0.107	0.610±0.493
	FedAvg+	0.802±0.028	0.726±0.630	0.697±0.117
	KD-PDFL (ours)	**0.816±0.032**	0.739±0.040	0.716±0.101
EMNIST	Local learning		0.697±0.209	
	FedAvg	0.764±0.119	0.784±0.108	0.824±0.142
	FedAvg+	0.771±0.104	0.806±0.123	0.841±0.136
	KD-PDFL (ours)	0.787±0.108	0.835±0.116	**0.870±0.082**

Table 4.1. Summary of per-client test accuracy under IoT devices ($T_{ex} = 20$) and EMNIST datasets ($T_{ex} = 5$).

normalization layer, one Rectified Linear Unit (ReLU) layer, and two linear layers. Each model has $39,769$ trainable parameters in total. The other set is for classification tasks using the EMNIST [121] dataset, which is composed of 28×28 pixel images of handwritten Roman alphabets and letters. In our experiments, a total of 47 class labels are included under balanced split settings. Each user has a CNN made of two convolutional layers, two max pooling layers, and two fully connected linear layers, which in total has $970,847$ trainable parameters. For both datasets, 10 to 40 users participate per experiment. The channel gain between each pair of participants follows Rayleigh fading, resulting in 5 neighbors reachable on average for each exchanging interval.

Fig. 4.4 shows that equal averaging over all participants, as provided in FedAvg, achieves the poorest performance as the environment is significantly non-i.i.d. [145]. Meanwhile, FedAvg+ compensates for the overfitting loss from FedAvg. The fine-tuning process began to adjust the hyperparameters from the moment the iteration reached $r = 2000$ in case of experiments with IoT devices dataset and $r = 150$ in case of EMNIST, which led to temporal deterioration, higher per-client test accuracies and lower training losses at the end.

Table 4.1 shows that KD-PDFL enables the clients to extend the upper bound of their estimated test accuracy without collaboration. Notably, users with small local training sets benefit from increased test accuracy from 21.0% to 81.6% on average. These weak users with small training sets also undergo a more challenging similarity decision. Since the lack of training set incurs blurry distance divergence across intermediate outputs that they calculate, estimating connectivity weights becomes more difficult.

Figuring out the relative importance becomes more manageable when a client can search over a lot of peers at once. Nonetheless, as N increases, settling down on the most effective collaborative graph is more challenging. Fig. 4.5 shows the trend in test accuracy with respect to the number of communicable peers per collaboration. Putting many neighbors into simultaneous consideration ends up walking on the same track with FedAvg, resulting in a prolonged

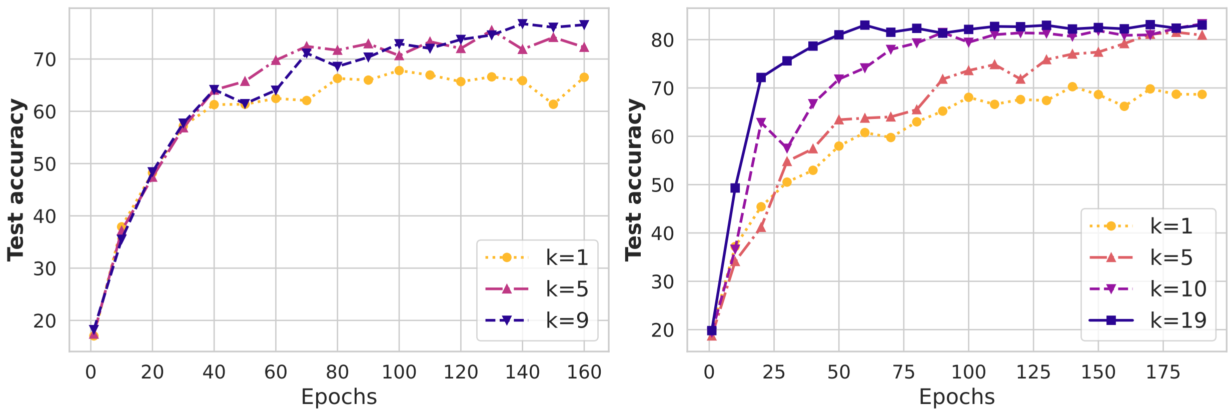


Figure 4.5. Trend on test accuracy improvement with respect to the number of connecting peers for each communication slot. (Left: $N = 10$, Right: $N = 20$, $T_{ex} = 5$ for both cases, dataset: EMNIST)

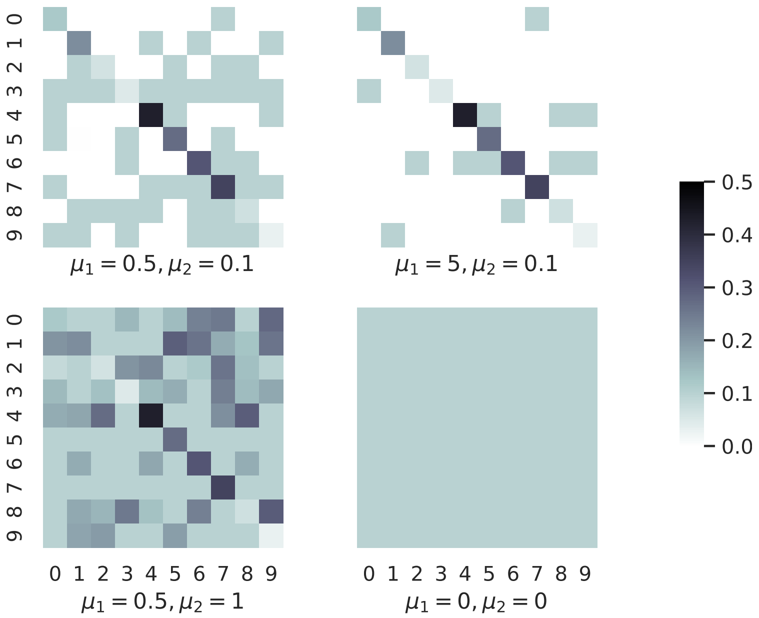


Figure 4.6. Heatmap of collaboration weight matrix under different values for μ_1 and μ_2.

"getting to know" session and slower model convergence. On this account, it is recommended to set an upper limit on the number of neighbors accessing per exchange interval, even though the channel conditions are good enough to cover many neighbors at once.

Fig. 4.6 visualizes the impact of coefficients of penalty and regularization terms on collaboration graphs. μ_1 is a multiplier on the personalization term of the joint loss function that controls the sensitivity to the statistic distances of the system. $\mu_1 \simeq 0$ turns off the effect of statistic distances on the training loss. Meanwhile, μ_2 adjusts the regularization to the weight values; thus, $\mu_2 \simeq 0$ may let some participants abandon cooperation and run only local training. A situation in which $\mu_1 = \mu_2 = 0$ is equivalent to federated learning that aims to construct a global model.

4.6 Concluding remarks

We introduced a distillation-based algorithm for personalized federated learning over fully decentralized networks, leveraging the privacy preservation and the convenience of measuring the statistical distance across clients using logits generated from local storage. Our experimental results showed that the proposed KD-PDFL is a promising decentralized approach compared to other personalized FL methods, with each device having full autonomy in computing, including updating a collaboration graph.

In future work, distillation-based personalization can be extended to unsupervised learning tasks since the distance measurement part does not require class labels as long as the local loss function does not include target labels in its metric. An interesting problem that arises by connecting topologies and connectivity graphs regarding physical distances among edge devices [146] is how to choose a subset of neighbors of a personalized and decentralized FL system in a communication-efficient way. A crossover with multi-task learning [147] also remains a potential direction of this work.

5 Final Remarks and Outlooks

5.1 Summary of Contributions

This thesis has made significant strides in the field of decentralized collaborative learning over wireless networks, focusing on enhancing the robustness, efficiency, and personalization of decentralized learning algorithms. One of the primary contributions of this work is the development of an asynchronous Decentralized Stochastic Gradient Descent (DSGD) algorithm designed to handle the inherent communication and computation failures in wireless networks. This algorithm maintains performance despite these challenges, ensuring reliable model updates. It has been rigorously analyzed, providing non-asymptotic convergence guarantees, and extensive experimental evaluations have demonstrated its effectiveness, showcasing the benefits of asynchronicity and the reuse of outdated gradient information.

Additionally, the thesis proposed a framework for asynchronous decentralized learning that decouples communication and computation timelines. This approach allows for greater autonomy among network users and improves the efficiency of the learning process. Our scheme, which is the first to consider collaborative model updates on a continuous timeline, was validated through convergence analysis and numerical experiments, highlighting its robustness and performance enhancements.

Addressing the need for personalization, we developed a personalized DSGD algorithm leveraging knowledge distillation to measure and quantify statistical dissimilarity between models. This approach ensures that users with similar data distributions are strongly connected, while those with different distributions are weakly connected. This personalization strategy significantly improves per-client performance by tailoring neural network models to each user's unique learning goals.

5.2 Future Directions

While this thesis has addressed several critical challenges in decentralized collaborative learning, numerous avenues for future research can further enhance the field. Future work could improve decentralized learning algorithms' robustness to more dynamic network topologies and varying communication conditions. Exploring adaptive algorithms that can quickly respond to changes in network states could be beneficial.

It is essential to investigate methods to further reduce the communication overhead and computational complexity in large-scale decentralized networks. Researchers focusing on communication efficiency in distributed learning can explore techniques such as more advanced model compression, adaptive communication schedules, and hierarchical aggregation. Additionally, integrating decentralized learning with emerging technologies such as 5G/6G networks, edge computing, and blockchain could open up new possibilities for secure, efficient, and scalable learning. Research could focus on leveraging these technologies to enhance the performance and applicability of decentralized learning systems.

While privacy preservation has been a focus, further enhancing the security of decentralized learning algorithms against potential attacks, such as adversarial attacks or data poisoning, remains a crucial area. Developing robust defense mechanisms to safeguard the learning process is vital. Tailoring decentralized learning algorithms to specific applications, such as autonomous vehicles, smart cities, and healthcare, could yield significant benefits. Understanding the unique requirements and constraints of these applications can lead to more effective and efficient learning solutions, underscoring the relevance of our research.

Future work could also explore how the system handles older or outdated updates, improving reliability and efficiency. Considering different learning rates or adjustments across various devices could enhance the algorithm's performance over a range of device capabilities. Additionally, adapting continuity in asynchronous DSGD for mobility scenarios, where distances and communication paths change over time in a three-dimensional space, could make our approach more practical for real-world applications. Simplifying the instructions would make it easier to use and more accessible in different settings. Addressing these challenges could further enhance the performance and usefulness of our methods, opening up new research avenues in asynchronous decentralized federated learning.

Expanding the personalization strategies to consider more complex user preferences and behaviors and incorporating multi-task learning approaches could improve the adaptability and relevance of the models trained in decentralized networks.

Additionally, the attitude and behavior of users in decentralized systems present another significant area for exploration. In this thesis, all participating devices have been assumed to be honest, actively engaged in collaboration, and altruistic. These devices selflessly help stragglers or slower learners, implying that their local objectives include collaborative goals beyond mere performance improvement. Future research could explore applying our proposed methods in systems with selfish or malicious users. Understanding and mitigating the impact of such behavior would be crucial for the robustness of decentralized learning systems.

Handling dynamic changes in the user population, where users may join or leave the network, making the number of participants (N) time-variant, is also a significant area for future research. Developing algorithms that can adapt to these changes will be essential for maintaining performance and reliability in decentralized learning environments.

In conclusion, this thesis has laid a strong foundation for decentralized collaborative learning over wireless networks. The proposed solutions have advanced the state-of-the-art, and the outlined future directions provide a roadmap for further research to push the boundaries of this exciting field. By addressing these challenges and exploring new avenues, future research can continue to enhance decentralized learning systems' efficiency, robustness, and applicability.

Appendix of Chapter 2

A.1 Proof of Theorem 2.1

We denote stale gradients by $g_i(\tilde{\theta}_i^{(r)}) = g_i(\theta_i^{(r-\tau_i)})$. According to the update rule, at each iteration $r+1$, we have

$$\mathbb{E}[f(\bar{\theta}^{r+1})] = \mathbb{E}\left[f\left(\bar{\theta}^r - \frac{1}{N}\sum_{i=1}^{N}\left(\tilde{\eta}_i^{(r)} g_i(\tilde{\theta}_i^{(r)}) + \zeta \tilde{n}_i^{(r)}\right)\right)\right]$$

where the expectation is w.r.t. the stochastic gradients, the communication noise $\Xi^{(r)}$, and the computation and communication failures at iteration $r+1$. For an L-smooth objective function, we have

$$\begin{aligned}\mathbb{E}[f(\bar{\theta}^{(r+1)})] \leq\ & f(\bar{\theta}^{(r)}) \underbrace{- \frac{1}{N}\sum_{i=1}^{N}\left\langle \nabla f(\bar{\theta}^{(r)}), \mathbb{E}[\tilde{\eta}_i^{(r)} g_i(\tilde{\theta}_i^{(r)}))]\right\rangle}_{:=T_1} \\ & \underbrace{+ \frac{L}{2N^2}\mathbb{E}\left\|\sum_{i=1}^{N}\tilde{\eta}_i^{(r)} g_i(\tilde{\theta}_i^{(r)}))\right\|^2}_{:=T_2} + \frac{L}{2N^2}\zeta^2\sum_{i=1}^{N}\mathbb{E}\left\|\tilde{n}_i^{(r)}\right\|^2\end{aligned}$$

where we used the fact that the communication noise has zero mean and is independent across users.

Adding and subtracting $\nabla f_i(\bar{\theta}^{(r)})$ to each summand of T_1, and as $\mathbb{E}[\tilde{\eta}_i^{(r)} g_i(\tilde{\theta}_i^{(r)})] = \eta \nabla f_i(\tilde{\theta}_i^{(r)})$, with $\eta = \min_j (1-\nu_j)/(\sqrt{4LR})$, we obtain

$$\begin{aligned} T_1 &= -\eta\left\langle \nabla f(\bar{\theta}^{(r)}), \frac{1}{N}\sum_{i=1}^{N}\nabla f_i(\tilde{\theta}_i^{(r)})\right\rangle \\ &= \frac{\eta}{2}\left\|\nabla f(\bar{\theta}^{(r)}) - \frac{1}{N}\sum_{i=1}^{N}\nabla f_i(\tilde{\theta}_i^{(r)})\right\|^2 - \frac{\eta}{2}\left\|\nabla f(\bar{\theta}^{(r)})\right\|^2 - \frac{\eta}{2N^2}\left\|\sum_{i=1}^{N}\nabla f_i(\tilde{\theta}_i^{(r)})\right\|^2 \\ &\leq \frac{\eta c}{2}\left\|\nabla f(\bar{\theta}^{(r)})\right\|^2 + \frac{\eta L^2}{2N}\sum_{i=1}^{N}\left\|\theta_i^{(r)} - \bar{\theta}^{(r)}\right\|^2 - \frac{\eta}{2}\left\|\nabla f(\bar{\theta}^{(r)})\right\|^2 - \frac{\eta}{2N^2}\left\|\sum_{i=1}^{N}\nabla f_i(\tilde{\theta}_i^{(r)})\right\|^2 \end{aligned}$$

where we have used the staleness assumption. The last term can be bounded using the property of the stochastic gradient and the fact that $\tilde{\eta}_i^{(r)} \leq 1/(\sqrt{4LR}) \leq 1/(\sqrt{4L})$ as

$$\begin{aligned}
T_2 &\leq \frac{L}{2N^2}\mathbb{E}\left\|\sum_{i=1}^{N}\tilde{\eta}_i^{(r)}[g_i(\tilde{\theta}_i^{(r)})-\nabla f_i(\tilde{\theta}_i^{(r)})]\right\|^2 + \frac{L}{2N^2}\mathbb{E}\left\|\sum_{i=1}^{N}\tilde{\eta}_i^{(r)}\nabla f_i(\tilde{\theta}_i^{(r)})\right\|^2 \\
&\leq \frac{\sigma^2}{8mT} + \frac{\eta}{8N^2}\mathbb{E}\left\|\sum_{i=1}^{N}\nabla f_i(\tilde{\theta}_i^{(r)})\right\|^2.
\end{aligned}$$

Summing T_1 and T_2 we obtain

$$T_1 + T_2 \leq -\frac{\eta}{2}(1-c)\left\|\nabla f(\bar{\theta}^{(r)})\right\|^2 + \frac{\sigma^2}{8NR} + \frac{\eta L^2}{2N}\sum_{i=1}^{N}\left\|\theta_i^{(r)} - \bar{\theta}^{(r)}\right\|^2 - \frac{\eta}{4N^2}\left\|\sum_{i=1}^{N}\nabla f_i(\tilde{\theta}_i^{(r)})\right\|^2.$$

Defining $c' = (1-c)$, telescoping and taking expectations we obtain

$$\begin{aligned}
\frac{1}{R}\sum_{r=1}^{R}\left\|\nabla f(\bar{\theta}^{(r)})\right\|^2 \leq 2\frac{f(\bar{\theta}^0)-f(\bar{\theta}^R)}{\eta R c'} + \frac{\sigma^2}{4\eta c' NR} + \frac{1}{R}\sum_{t=1}^{R}\frac{L^2}{Nc'}\sum_{i=1}^{N}\mathbb{E}\left\|\theta_i^{(r)} - \bar{\theta}^{(r)}\right\|^2 \\
+ \frac{1}{R}\sum_{r=1}^{R}\frac{L\zeta^2}{\eta N^2 c'}\sum_{i=1}^{N}\mathbb{E}\left\|\tilde{n}_i^{(r)}\right\|^2.
\end{aligned}$$

Defining $\sigma_{w,i}^2 = \max_{r=0}^{R}\mathbb{E}\left\|\tilde{n}_i^{(r)}\right\|^2$ and bounding the consensus term by Lemma 2.2, we obtain

$$\begin{aligned}
\frac{1}{R}\sum_{r=1}^{R}\left\|\nabla f(\bar{\theta}^{(r)})\right\|^2 \leq 2\frac{f(\bar{\theta}^0)-f(\bar{\theta}^R)}{\eta R c'} + \frac{L^2}{Nc'}\left(\eta^2\frac{12NG^2}{(p\zeta)^2} + \zeta\frac{2}{p}\sum_{i=1}^{N}\sigma_{w,i}^2\right) \\
+ \frac{\sigma^2}{4\eta c' NR} + \frac{L\zeta^2}{\eta N^2 c'}\sum_{i=1}^{N}\sigma_{w,i}^2.
\end{aligned}$$

The final result is obtained setting $\eta = 1/\sqrt{4LR}$ and $\zeta = 1/R^{3/8}$.

Appendix of Chapter 3

B.1 Proofs

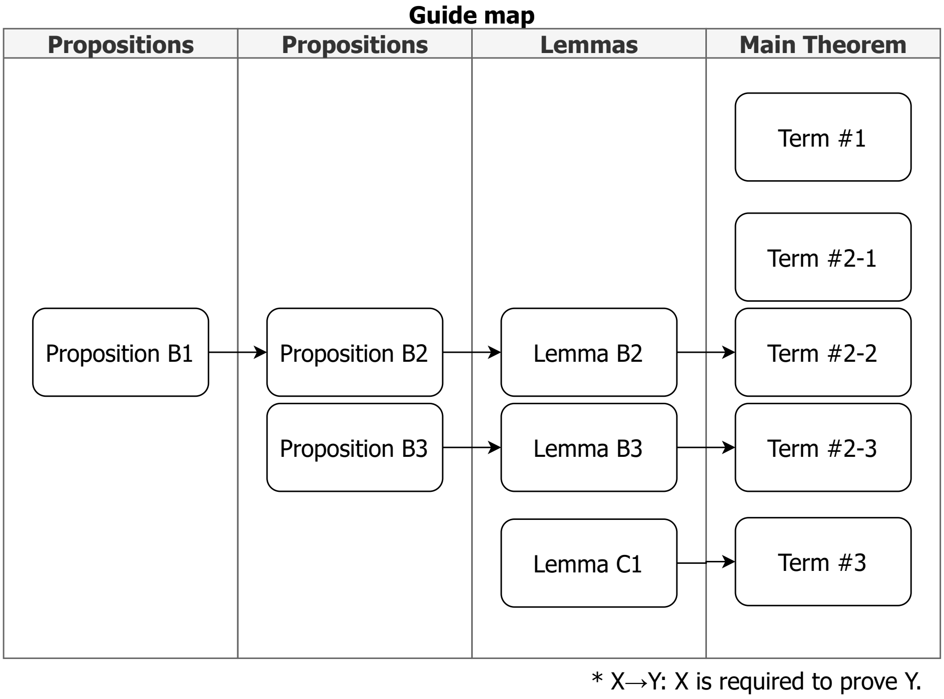


Figure B.1. A metaphoric map that guides the correlation of each proposition and lemma in order to prove the main theorem.

B.1.1 Preliminaries

Before the proof of Theorem 3.1, it is essential to verify (i) how many communication events and (ii) how many local gradient updates occur during P. In PPP, communication events occur $\lambda_i P$ times on average during P, which indicates the expectation of broadcasting frequency of node i. In order to find the bound for $\theta_{t_0+P}-\theta_{t_0}$, we need to specify how many reception events happen in a random node i during the elapsed time of P. For simplicity, we write $\int_P$ to indicate

$\int_{t_0}^{t_0+P}$. The reference model of node i update during P is

$$
\begin{aligned}
\theta_{t_0+P}^{(i)} - \theta_{t_0}^{(i)} &= \int_P \sum_j \Pr[i \in \mathcal{N}_t(j)] \Delta_t^{(j)} \, \mathrm{d}t \\
&= \int_P \sum_j q_t^{j \to i} \Delta_t^{(j)} \, \mathrm{d}t \\
&= \eta \int_P \sum_j q_t^{j \to i} \sum_{b=0}^{B-1} \mathbf{g}_j(\mathbf{y}_{\lfloor t \rfloor, b}^{(j)}) \, \mathrm{d}t \, ,
\end{aligned}
$$

which is heterogeneous across nodes. A floored notation $\lfloor t \rfloor$ indicates the latest moment no later than time t that user j computes $\Delta^{(j)}$.

A superscripted or subscripted $\star$ on some variables is analogous to a "don't-care" (DC) term in digital logic [148]. For instance, $q_\star$ is the same as any q_i, where i can be any user index in $\mathcal{U}$ without loss of generality.

B.1.2 Propositions

Proposition B.1. *If Assumption 3.5 is satisfied, a decentralized learning network with $N \geq 4$ clients satisfies*

$$
\sum_{j \neq i} q_t^{j \to i} \left\| \nabla f_j(\theta_t^{(j)}) - \nabla f_i(\theta_t^{(i)}) \right\|^2 \leq \frac{9N\zeta^2}{4}
$$

for all $i, j \in \mathcal{U}$ and $t \in [0, T)$.

Proof.

$$
\begin{aligned}
&\sum_{j \neq i} q_t^{j \to i} \left\| \nabla f_j(\theta_t^{(j)}) - \nabla f_i(\theta_t^{(i)}) \right\|^2 \\
&= \sum_{j \neq i} q_t^{j \to i} \left\| \nabla f_j(\theta_t^{(j)}) - \nabla f(\theta_t) - \nabla f_i(\theta_t^{(i)}) + \nabla f(\theta_t) \right\|^2 \\
&\overset{(a)}{\leq} \Big(1 + \frac{1}{\sqrt{N}}\Big) \sum_{j \neq i} q_t^{j \to i} \left\| \nabla f_j(\theta_t^{(j)}) - \nabla f(\theta_t) \right\|^2 + (\sqrt{N}+1) \sum_{j \neq i} q_t^{j \to i} \left\| \nabla f_i(\theta_t^{(i)}) - \nabla f(\theta_t) \right\|^2 \\
&\overset{(b)}{\leq} \Big(1 + \frac{1}{\sqrt{N}}\Big) \sum_{j \neq i} q_t^{j \to i} \left\| \nabla f_j(\theta_t^{(j)}) - \nabla f(\theta_t) \right\|^2 + (\sqrt{N}+1)\zeta^2 \\
&\overset{(c)}{\leq} (N + 2\sqrt{N} + 1)\zeta^2 \quad \overset{(d)}{\leq} \frac{9N\zeta^2}{4},
\end{aligned}
$$

where (a) is due to Young's inequality; (b) comes from Assumption 3.5; (c) takes the fact that $q_\star^\star \leq 1$ for any user nodes; (d) is always true for $N \geq 4$ since $\frac{5N}{4} - 2\sqrt{N} - 1 \geq 0$ for any $\sqrt{N} \geq 2$, which satisfies the given condition about N. □

Remark. This proposition appears in the proof of Proposition B.2.

Proposition B.2. *(Upper bound for superpositioned model deviations.) Let $h_j(b) = \mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)}$ denote the difference between a local model calculated by feeding each batch with an index b and the local reference model. For all $i, j \in \mathcal{U}$, when $\eta \leq \frac{1}{8BL}$, we have*

$$\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\big[\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)}\|^2\big] \leq \frac{2}{5}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\theta_t^{(i)} - \theta_{t_0}^{(i)}\Big\|^2\Big] + \frac{9N\zeta^2}{10L^2} + \frac{16B\eta^2\sigma^2}{5} + \frac{128B^2\eta^2}{5}\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2]\,.$$

Proof. We rephrase the $b+1^{\text{th}}$ term, $h_j(b+1) = \mathbf{y}_{t,b+1}^{(j)} - \theta_t^{(j)}$, as follows.

$$\begin{aligned}
&\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\mathbf{y}_{t,b+1}^{(j)} - \theta_t^{(j)}\Big\|^2\Big] \\
&= \sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)} - \eta g_j(\mathbf{y}_{t,b}^{(j)})\Big\|^2\Big] \\
&\overset{(a)}{=} \sum_{j\neq i} q_t^{j\to i}\Big\|\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)} - \eta g_j(\mathbf{y}_{t,b}^{(j)})\Big]\Big\|^2 \\
&\quad + \sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Bigg[\Big\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)} - \eta g_j(\mathbf{y}_{t,b}^{(j)}) - \mathbb{E}_{\cdot|\mathcal{Q}}\Big[\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)} - \eta g_j(\mathbf{y}_{t,b}^{(j)})\Big]\Big\|^2\Bigg] \\
&= \sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)} - \eta\nabla f_j(\mathbf{y}_{t,b}^{(j)})\Big\|^2\Big] + \sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\eta\big(g_j(\mathbf{y}_{t,b}^{(j)}) - \nabla f_j(\mathbf{y}_{t,b}^{(j)})\big)\Big\|^2\Big] \\
&\overset{(b)}{\leq} \sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)} - \eta\nabla f_j(\mathbf{y}_{t,b}^{(j)})\Big\|^2\Big] + \eta^2\sigma^2 \\
&= \sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)} + \eta\nabla f_j(\theta_t^{(j)}) - \eta\nabla f_j(\mathbf{y}_{t,b}^{(j)}) - \eta\nabla f_j(\theta_t^{(j)}) - \eta\nabla f_i(\theta_t^{(i)}) \\
&\qquad\qquad + \eta\nabla f_i(\theta_t^{(i)}) - \eta\nabla f_i(\theta_{t_0}^{(i)}) + \eta\nabla f_i(\theta_{t_0}^{(i)})\Big\|^2\Big] + \eta^2\sigma^2 \\
&\overset{(c)}{\leq} \Big(1 + \frac{1}{2B-1}\Big)\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\big[\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)}\|^2\big] \\
&\quad + 2B\eta^2\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\nabla f_j(\mathbf{y}_{t,b}^{(j)}) - \nabla f_j(\theta_t^{(j)}) + \nabla f_j(\theta_t^{(j)}) - \nabla f_i(\theta_t^{(i)}) \\
&\qquad\qquad + \nabla f_i(\theta_t^{(i)}) - \nabla f_i(\theta_{t_0}^{(i)}) + \nabla f_i(\theta_{t_0}^{(i)})\Big\|^2\Big] + \eta^2\sigma^2 \\
&\leq \Big(1 + \frac{1}{2B-1}\Big)\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\big[\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)}\|^2\big] \\
&\quad + 8B\eta^2\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\nabla f_j(\mathbf{y}_{t,b}^{(j)}) - \nabla f_j(\theta_t^{(j)})\Big\|^2\Big] \\
&\quad + 8B\eta^2\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\big[\|\nabla f_j(\theta_t^{(j)}) - \nabla f_i(\theta_t^{(i)})\|^2\big]
\end{aligned}$$

$$
\begin{aligned}
&+ 8B\eta^2 \sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_t^{(i)}) - \nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big] \\
&+ 8B\eta^2 \sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big] + \eta^2\sigma^2 \\
\overset{(d)}{\leq}& \Big(1+\frac{1}{2B-1}\Big)\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)}\big\|^2\Big] + 8BL^2\eta^2\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)}\big\|^2\Big] \\
&+ 18BN\eta^2\zeta^2 + 8BL^2\eta^2\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)} - \theta_{t_0}^{(i)}\big\|^2\Big] + 8B\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2] + \eta^2\sigma^2 \\
=& \Big(1 + 8BL^2\eta^2 + \frac{1}{2B-1}\Big)\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\big[\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)}\|^2\big] \\
&+ 18BN\eta^2\zeta^2 + \eta^2\sigma^2 + 8BL^2\eta^2\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)} - \theta_{t_0}^{(i)}\big\|^2\Big] + 8B\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2] \\
\overset{(e)}{\leq}& \left(1+\frac{5}{8\big(B-\frac{1}{2}\big)}\right)\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\big[\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)}\|^2\big] + \frac{9N\zeta^2}{32BL^2} + \eta^2\sigma^2 \\
&+ \frac{1}{8B}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)} - \theta_{t_0}^{(i)}\big\|^2\Big] + 8B\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2] \qquad \text{(B.1)}
\end{aligned}
$$

where (a) is from the definition of variance; (b) is derived from the definition of σ in Assumption 3.4; (c) Young's inequality; (d) uses L-smoothness on the second and the fourth term, and applies Proposition B.1 on the third term. Afterwards, the first two terms are integrated; (e) is derived from the fact that $\eta^2 \leq \frac{1}{64B^2L^2}$ and that

$$
8BL^2\eta^2 + \frac{1}{2B-1} \leq \frac{1}{8B} + \frac{1}{2B-1} \leq \frac{1}{8B-4} + \frac{1}{2B-1} = \frac{5}{8\big(B-\frac{1}{2}\big)}.
$$

Let $H(b)$ indicate $\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)}\big\|^2\Big]$. From the last line of inequality B.1, we have

$$
\begin{aligned}
H(b+1) \leq& \left(1+\frac{5}{8\big(B-\frac{1}{2}\big)}\right)H(b) + \frac{1}{8B}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)} - \theta_{t_0}^{(i)}\big\|^2\Big] \\
&+ 18BN\eta^2\zeta^2 + \eta^2\sigma^2 + 8B\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2]\,. \qquad \text{(B.2)}
\end{aligned}
$$

Since $\mathbf{y}_{t,0}^{(j)} = \theta_t^{(j)}$ for all $t,\ j$ based on Algorithm 3,

$$
H(0) = \sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\big[\|\mathbf{y}_{t,0}^{(j)} - \theta_t^{(j)}\|^2\big] = 0.
$$

Recurring inequality B.2 from $H(0)$, we can get

$$
H(b) \leq \left(1+\frac{5}{8\big(B-\frac{1}{2}\big)}\right)^b H(0) + \sum_{b'=0}^{b-1}\left(1+\frac{5}{8\big(B-\frac{1}{2}\big)}\right)^{b'}
$$

$$
\begin{aligned}
&\cdot\Big(\frac{1}{8B}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)}-\theta_{t_0}^{(i)}\big\|^2\Big]+\frac{9N\zeta^2}{32BL^2}+\eta^2\sigma^2+8B\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2]\Big)\\
\le&\Big(\frac{1}{8B}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)}-\theta_{t_0}^{(i)}\big\|^2\Big]+\frac{9N\zeta^2}{32BL^2}+\eta^2\sigma^2+8B\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2]\Big)\\
&\cdot\sum_{b=0}^{B-1}\left(1+\frac{5}{8\big(B-\frac{1}{2}\big)}\right)^{b}\\
=&\left[\left(1+\frac{5}{8\big(B-\frac{1}{2}\big)}\right)^{B}-1\right]\cdot\frac{8\big(B-\frac{1}{2}\big)}{5}\\
&\cdot\Big(\frac{1}{8B}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)}-\theta_{t_0}^{(i)}\big\|^2\Big]+\frac{9N\zeta^2}{32BL^2}+\eta^2\sigma^2+8B\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2]\Big)\\
=&\left[\left(1+\frac{5}{8\big(B-\frac{1}{2}\big)}\right)^{B-\frac{1}{2}}\left(1+\frac{5}{8\big(B-\frac{1}{2}\big)}\right)^{\frac{1}{2}}-1\right]\cdot\frac{8\big(B-\frac{1}{2}\big)}{5}\\
&\cdot\Big(\frac{1}{8B}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)}-\theta_{t_0}^{(i)}\big\|^2\Big]+\frac{9N\zeta^2}{32BL^2}+\eta^2\sigma^2+8B\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2]\Big)\\
\overset{(a)}{\le}&\left[e^{\frac{5}{8}}\cdot\frac{3}{2}-1\right]\cdot\frac{8\big(B-\frac{1}{2}\big)}{5}\\
&\cdot\Big(\frac{1}{8B}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)}-\theta_{t_0}^{(i)}\big\|^2\Big]+\frac{9N\zeta^2}{32BL^2}+\eta^2\sigma^2+8B\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2]\Big)\\
\overset{(b)}{\le}&\frac{16}{5}B\Big(\frac{1}{8B}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)}-\theta_{t_0}^{(i)}\big\|^2\Big]+\frac{9N\zeta^2}{32BL^2}+\eta^2\sigma^2+8B\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2]\Big)\\
=&\frac{2}{5}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)}-\theta_{t_0}^{(i)}\big\|^2\Big]+\frac{9N\zeta^2}{10L^2}+\frac{16B\eta^2\sigma^2}{5}+\frac{128B^2\eta^2}{5}\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2]
\end{aligned}
$$

where (a) comes from $(1+x)^{1/x}\le e$ and $B\ge 1$, which results in $\Big(1+\frac{5}{8(B-(1/2))}\Big)^{1/2}\le\big(1+\frac{5}{4}\big)^{1/2}=\frac{3}{2}$; (b) is due to $\frac{3}{2}e^{\frac{5}{8}}-1\le 2$ and $B-\frac{1}{2}\le B$. □

Remark. This proposition appears in the proof of Lemma B.2, which is used at the first term of inequality B.10.

Proposition B.3. *(Upper bound for the local reference model change) When Assumption 3.2 holds and* $\eta\le\min(\frac{1}{8BL},\frac{1}{8BLN\Psi})$*, we have*

$$
\begin{aligned}
\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_t^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big]\le 2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\|\mathbf{y}_{t,b}^{(j)}-\theta_t^{(j)}\|^2\Big]+\frac{8\zeta^2}{L^2(N-4)}\\
+\frac{1}{16L^2N^2}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big]+\frac{3\sigma^2}{16L^2},
\end{aligned}
$$

for all $i,j\in\mathcal{U}$ *and for* $t\in[t_0,t_0+P)$.

Proof. Here, we use $\int_\tau$ to replace $\int_{\tau=t_0}^{t}$ for simplicity of writing. The left side of the inequality can be rephrased as below by bringing Appendix B.1.1.

$$
\begin{aligned}
&\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_t^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big] \\
&= \mathbb{E}_{\cdot|\mathcal{Q}}\bigg[\sum_{j\neq i} q_t^{j\to i}\bigg\|\eta\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1} g_n(\mathbf{y}_{\tau,b}^{(n)})\,\mathrm{d}\tau\ \bigg\|^2\bigg] \\
&= \mathbb{E}_{\cdot|\mathcal{Q}}\bigg[\sum_{j\neq i} q_t^{j\to i}\bigg\|\eta\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1} \nabla f_n(\mathbf{y}_{\tau,b}^{(n)})\,\mathrm{d}\tau \\
&\qquad\qquad + \eta\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1}\big[g_n(\mathbf{y}_{\tau,b}^{(n)})-\nabla f_n(\mathbf{y}_{\tau,b}^{(n)})\big]\,\mathrm{d}\tau\ \bigg\|^2\bigg] \\
&\overset{(i)}{\leq} \mu_1\mathbb{E}_{\cdot|\mathcal{Q}}\bigg[\sum_{j\neq i} q_t^{j\to i}\bigg\|\eta\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1} \nabla f_n(\mathbf{y}_{\tau,b}^{(n)})\,\mathrm{d}\tau\ \bigg\|^2\bigg] \\
&\quad + \Big(1+\frac{1}{\mu_1-1}\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\bigg[\sum_{j\neq i} q_t^{j\to i}\bigg\|\eta\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1}\big[g_n(\mathbf{y}_{\tau,b}^{(n)})-\nabla f_n(\mathbf{y}_{\tau,b}^{(n)})\big]\,\mathrm{d}\tau\ \bigg\|^2\bigg] \\
&\overset{(a)}{\leq} \mu_1\mathbb{E}_{\cdot|\mathcal{Q}}\bigg[\sum_{j\neq i} q_t^{j\to i}\bigg\|\eta\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1} \nabla f_n(\mathbf{y}_{\tau,b}^{(n)})\,\mathrm{d}\tau\ \bigg\|^2\bigg] \\
&\quad + \Big(1+\frac{1}{\mu_1-1}\Big)\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}\bigg[\sum_{j\neq i} q_t^{j\to i}\psi_j(t_0,t)\int_{/\tau}\bigg\|\sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1}\big[g_n(\mathbf{y}_{\tau,b}^{(n)})-\nabla f_n(\mathbf{y}_{\tau,b}^{(n)})\big]\bigg\|^2\,\mathrm{d}\tau\ \bigg] \\
&\leq \mu_1\mathbb{E}_{\cdot|\mathcal{Q}}\bigg[\sum_{j\neq i} q_t^{j\to i}\bigg\|\eta\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1} \nabla f_n(\mathbf{y}_{\tau,b}^{(n)})\,\mathrm{d}\tau\ \bigg\|^2\bigg] \\
&\quad + \Big(1+\frac{1}{\mu_1-1}\Big)N\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}\bigg[\sum_{j\neq i} q_t^{j\to i}\psi_j(t_0,t)\int_{\tau}\sum_{n\neq j} q_\tau^{n\to j}\bigg\|\sum_{b=0}^{B-1}\big[g_n(\mathbf{y}_{\tau,b}^{(n)})-\nabla f_n(\mathbf{y}_{\tau,b}^{(n)})\big]\bigg\|^2\,\mathrm{d}\tau\ \bigg] \\
&\leq \mu_1\mathbb{E}_{\cdot|\mathcal{Q}}\bigg[\sum_{j\neq i} q_t^{j\to i}\bigg\|\eta\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1} \nabla f_n(\mathbf{y}_{\tau,b}^{(n)})\,\mathrm{d}\tau\ \bigg\|^2\bigg] \\
&\quad + \Big(1+\frac{1}{\mu_1-1}\Big)BN\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}\bigg[\sum_{j\neq i} q_t^{j\to i}\psi_j(t_0,t)\int_{\tau}\sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1}\big\|g_n(\mathbf{y}_{\tau,\star}^{(n)})-\nabla f_n(\mathbf{y}_{\tau,\star}^{(n)})\big\|^2\,\mathrm{d}\tau\ \bigg] \\
&\overset{(b)}{\leq} \mu_1\mathbb{E}_{\cdot|\mathcal{Q}}\bigg[\sum_{j\neq i} q_t^{j\to i}\bigg\|\eta\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1} \nabla f_n(\mathbf{y}_{\tau,b}^{(n)})\,\mathrm{d}\tau\ \bigg\|^2\bigg] + \Big(1+\frac{1}{\mu_1-1}\Big)B^2N^2\eta^2\psi_j^2(t_0,t)\sigma^2 \\
&\leq \mu_1\mathbb{E}_{\cdot|\mathcal{Q}}\bigg[\sum_{j\neq i} q_t^{j\to i}\bigg\|\eta\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1}\Big[\nabla f_n(\mathbf{y}_{\tau,b}^{(n)})-\nabla f_n(\theta_\tau^{(n)})+\nabla f_n(\theta_\tau^{(n)})-\nabla f_j(\theta_\tau^{(j)}) \\
&\qquad + \nabla f_j(\theta_\tau^{(j)})-\nabla f_j(\theta_{t_0}^{(j)})+\nabla f_j(\theta_{t_0}^{(j)})-\nabla f_i(\theta_{t_0}^{(i)})+\nabla f_i(\theta_{t_0}^{(i)})\Big]\,\mathrm{d}\tau\ \bigg\|^2\bigg]
\end{aligned}
$$

$$
\begin{aligned}
&+\Big(1+\frac{1}{\mu_1-1}\Big)B^2N^2\eta^2\psi_j^2(t_0,t)\sigma^2\\
&=\mu_1\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i}q_t^{j\to i}\Big\|\eta\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\sum_{b=0}^{B-1}\Big[\nabla f_n(\mathbf{y}_{\tau,b}^{(n)})-\nabla f_n(\theta_\tau^{(n)})\Big]\mathrm{d}\tau\\
&\qquad\qquad+\eta\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\sum_{b=0}^{B-1}\Big[\nabla f_n(\theta_\tau^{(n)})-\nabla f_j(\theta_\tau^{(j)})\Big]\mathrm{d}\tau\\
&\qquad\qquad+\eta\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\sum_{b=0}^{B-1}\Big[\nabla f_j(\theta_\tau^{(j)})-\nabla f_j(\theta_{t_0}^{(j)})\Big]\mathrm{d}\tau\\
&\qquad\qquad+\eta\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\sum_{b=0}^{B-1}\Big[\nabla f_j(\theta_{t_0}^{(j)})-\nabla f_i(\theta_{t_0}^{(i)})\Big]\mathrm{d}\tau\\
&\qquad\qquad+\eta\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\sum_{b=0}^{B-1}\nabla f_i(\theta_{t_0}^{(i)})\,\mathrm{d}\tau\Big\|^2\Big]+\Big(1+\frac{1}{\mu_1-1}\Big)B^2N^2\eta^2\psi_j^2(t_0,t)\sigma^2\\
&\overset{(ii,c)}{\leq}4\mu_1\mu_2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i}q_t^{j\to i}\Big\|\eta\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\sum_{b=0}^{B-1}\Big[\nabla f_n(\mathbf{y}_{\tau,b}^{(n)})-\nabla f_n(\theta_\tau^{(n)})\Big]\mathrm{d}\tau\Big\|^2\Big]\\
&+4\mu_1\mu_2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i}q_t^{j\to i}\Big\|\eta\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\sum_{b=0}^{B-1}\Big[\nabla f_n(\theta_\tau^{(n)})-\nabla f_j(\theta_\tau^{(j)})\Big]\mathrm{d}\tau\Big\|^2\Big]\\
&+4\mu_1\mu_2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i}q_t^{j\to i}\Big\|\eta\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\sum_{b=0}^{B-1}\Big[\nabla f_j(\theta_\tau^{(j)})-\nabla f_j(\theta_{t_0}^{(j)})\Big]\mathrm{d}\tau\Big\|^2\Big]\\
&+4\mu_1\mu_2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i}q_t^{j\to i}\Big\|\eta\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\sum_{b=0}^{B-1}\Big[\nabla f_j(\theta_{t_0}^{(j)})-\nabla f_i(\theta_{t_0}^{(i)})\Big]\mathrm{d}\tau\Big\|^2\Big]\\
&+\mu_1\Big(1+\frac{1}{\mu_2-1}\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i}q_t^{j\to i}\Big\|\eta\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\sum_{b=0}^{B-1}\nabla f_i(\theta_{t_0}^{(i)})\,\mathrm{d}\tau\Big\|^2\Big]\\
&+\Big(1+\frac{1}{\mu_1-1}\Big)B^2N^2\eta^2\psi_j^2(t_0,t)\sigma^2\\
&\leq 4\mu_1\mu_2BN\eta^2\psi_j(t_0,t)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i}q_t^{j\to i}\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\sum_{b=0}^{B-1}\Big\|\nabla f_n(\mathbf{y}_{\tau,b}^{(n)})-\nabla f_n(\theta_\tau^{(n)})\Big\|^2\mathrm{d}\tau\Big]\\
&+4\mu_1\mu_2B^2\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i}q_t^{j\to i}\Big\|\psi_j(t_0,t)\sum_{n\neq j}q_\star^{n\to j}\Big[\nabla f_n(\theta_\star^{(n)})-\nabla f_j(\theta_\star^{(j)})\Big]\Big\|^2\Big]\\
&+4\mu_1\mu_2B^2\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i}q_t^{j\to i}\Big\|\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\Big[\nabla f_j(\theta_\tau^{(j)})-\nabla f_j(\theta_{t_0}^{(j)})\Big]\mathrm{d}\tau\Big\|^2\Big]\\
&+4\mu_1\mu_2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i}q_t^{j\to i}\Big\|\nabla f_j(\theta_{t_0}^{(j)})-\nabla f_i(\theta_{t_0}^{(i)})\Big\|^2\cdot\Big\|\eta\int_\tau\sum_{n\neq j}q_\tau^{n\to j}\sum_{b=0}^{B-1}1\,\mathrm{d}\tau\Big\|^2\Big]\\
&+\mu_1\Big(1+\frac{1}{\mu_2-1}\Big)B^2\eta^2\psi_j^2(t_0,t)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\nabla f_i(\theta_{t_0}^{(i)})\Big\|^2\Big]+\Big(1+\frac{1}{\mu_1-1}\Big)B^2N^2\eta^2\psi_j^2(t_0,t)\sigma^2
\end{aligned}
$$

$$
\begin{aligned}
&\overset{(d)}{\le} 4\mu_1\mu_2 BL^2 N\eta^2\psi_j(t_0,t)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1}\big\|\mathbf{y}_{\tau,b}^{(n)}-\theta_\tau^{(n)}\big\|^2\,\mathrm{d}\tau\Big]\\
&\quad+4\mu_1\mu_2 B^2\eta^2\psi_j^2(t_0,t)\cdot\frac{2N\zeta^2}{N-4}\\
&\quad+4\mu_1\mu_2 B^2 L^2\eta^2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\Big\|\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\big[\theta_\tau^{(j)}-\theta_{t_0}^{(j)}\big]\mathrm{d}\tau\Big\|^2\Big]\\
&\quad+\frac{8\mu_1\mu_2 B^2 N\eta^2\psi_j^2(t_0,t)\zeta^2}{N-4}+\mu_1\Big(1+\frac{1}{\mu_2-1}\Big)B^2\eta^2\psi_j^2(t_0,t)\mathbb{E}_{\cdot|\mathcal{Q}}\big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\big]\\
&\quad+\Big(1+\frac{1}{\mu_1-1}\Big)B^2N^2\eta^2\psi_j^2(t_0,t)\sigma^2\\
&\overset{(e)}{\le} 4\mu_1\mu_2 BL^2 N\eta^2\psi_j(t_0,t)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1}\big\|\mathbf{y}_{\tau,b}^{(n)}-\theta_\tau^{(n)}\big\|^2\,\mathrm{d}\tau\Big]\\
&\quad+4\mu_1\mu_2 B^2 L^2\eta^2\psi_j^2(t_0,t)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_{\tau_{\max}}^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big]\\
&\quad+\frac{16\mu_1\mu_2 B^2 N\eta^2\psi_j^2(t_0,t)\zeta^2}{N-4}+\mu_1\Big(1+\frac{1}{\mu_2-1}\Big)B^2\eta^2\psi_j^2(t_0,t)\mathbb{E}_{\cdot|\mathcal{Q}}\big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\big]\\
&\quad+\Big(1+\frac{1}{\mu_1-1}\Big)B^2N^2\eta^2\psi_j^2(t_0,t)\sigma^2\ , \qquad\qquad (B.3)
\end{aligned}
$$

where two coefficients larger than one, denoted by μ_1 and μ_2, are introduced in (i) and (ii), respectively. In inequality B.3, (a) uses

$$
\Big\|\int_{\tau=t_0}^{t}\sum_{n\neq j} q_\tau^{nj}\mathbf{z}_\tau\,\mathrm{d}\tau\Big\|^2 \le \psi_j(t_0,t)\int_{\tau=t_0}^{t}\|\sum_{n\neq j} q_\tau^{nj}\mathbf{z}_\tau\|^2\,\mathrm{d}\tau
$$

and

$$
\Big\|\sum_{m=1}^{M}\mathbf{z}_m\Big\|^2 \le M\sum_{m=1}^{M}\|\mathbf{z}_m\|^2
$$

for any vector $\mathbf{z}_\star \in \mathbb{R}^d$.[1] (b) comes from the definition of σ^2 in Assumption 3.4. In (c), Jensen's inequality is applied once again. (d) takes L-smoothness on the first and the second term, while Lemma 3.1 is applied on the third term. In (e) the third term of inequality B.3 already contains the current lemma. Here, we introduce an index of the instant $\tau_{\max} \in [t_0, t)$ that satisfies $\tau_{\max} = \operatorname{argmax}_\tau \|\theta_\tau^{(j)} - \theta_{t_0}^{(j)}\|^2$.

The first term of inequality B.3, which includes Lemma B.2, can be rephrased as follows:

$$
\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\sum_{b=0}^{B-1}\big\|\mathbf{y}_{\tau,b}^{(n)}-\theta_\tau^{(n)}\big\|^2\,\mathrm{d}\tau\Big]
$$

[1] This results in $\big\|\sum_{j\in\mathcal{U}} q_\star^{j\to i}\mathbf{z}_j\big\|^2 \le N\sum_{j\in\mathcal{U}} q_\star^{j\to i}\|\mathbf{z}_j\|^2$ and $\|\sum_{b=0}^{B-1}\mathbf{z}_b\|^2 \le B\sum_{b=0}^{B-1}\|\mathbf{z}_b\|^2$.

$$
\begin{aligned}
&\leq B\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\int_\tau \sum_{n\neq j} q_\tau^{n\to j}\big\|\mathbf{y}_{\tau,\star}^{(n)}-\theta_\tau^{(n)}\big\|^2\,\mathrm{d}\tau\Big]\\
&\leq B\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\psi_j(t_0,t)\sum_{n\neq j} q_\star^{n\to j}\big\|\mathbf{y}_{\star,\star}^{(n)}-\theta_\star^{(n)}\big\|^2\Big]\\
&\leq B\psi_j(t_0,t)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{n\neq j} q_\tau^{n\to j}\|\mathbf{y}_{\tau,b}^{(n)}-\theta_\tau^{(n)}\|^2\Big] \qquad \text{(B.4)}
\end{aligned}
$$

We continue rephrasing the primary inequality B.3:

$$
\begin{aligned}
&\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_t^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big]\\
&\leq 4\mu_1\mu_2B^2L^2N\eta^2\psi_j^2(t_0,t)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{n\neq j} q_\tau^{n\to j}\|\mathbf{y}_{\tau,b}^{(n)}-\theta_\tau^{(n)}\|^2\Big]\\
&\quad+4\mu_1\mu_2B^2L^2\eta^2\psi_j^2(t_0,t)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_{\tau_{\max}}^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big]\\
&\quad+\frac{16\mu_1\mu_2B^2N\eta^2\psi_j^2(t_0,t)\zeta^2}{N-4}+\mu_1\Big(1+\frac{1}{\mu_2-1}\Big)B^2\eta^2\psi_j^2(t_0,t)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big]\\
&\quad+\Big(1+\frac{1}{\mu_1-1}\Big)B^2N^2\eta^2\psi_j^2(t_0,t)\sigma^2
\end{aligned}
$$

After rearranging the inequality in order to integrate those terms including $\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_\star^{j\to i}\big\|\theta_\star^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big]$, we have

$$
\begin{aligned}
&\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_t^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big]-4\mu_1\mu_2B^2L^2\eta^2\psi_j^2(t_0,t)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_{\tau_{\max}}^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big]\\
&\leq(1-4\mu_1\mu_2B^2L^2\eta^2\Psi^2)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_t^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big]\\
&\leq(1-4\mu_1\mu_2B^2L^2N\eta^2\Psi^2)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_t^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big].
\end{aligned}
$$

Hence, we can rephrase the inequality as

$$
\begin{aligned}
&\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_t^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big]\\
&\leq\frac{1}{1-4\mu_1\mu_2B^2L^2N\eta^2\Psi^2}\cdot\Bigg[4\mu_1\mu_2B^2L^2N\eta^2\Psi^2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{n\neq j} q_\tau^{n\to j}\|\mathbf{y}_{\tau,b}^{(n)}-\theta_\tau^{(n)}\|^2\Big]\\
&\quad+\frac{16\mu_1\mu_2B^2N\eta^2\zeta^2\Psi^2}{N-4}+\mu_1\Big(1+\frac{1}{\mu_2-1}\Big)B^2\eta^2\Psi^2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big]\\
&\quad+\Big(1+\frac{1}{\mu_1-1}\Big)B^2N^2\eta^2\sigma^2\Psi^2\Bigg]
\end{aligned}
$$

$$\begin{aligned}
&\overset{(i)}{\leq} 3 \cdot \left[\frac{2}{3} \mathbb{E}_{\cdot|\mathcal{Q}} \Big[\sum_{n \neq j} q_\tau^{n \to j} \| \mathbf{y}_{\tau,b}^{(n)} - \theta_\tau^{(n)} \|^2 \Big] + \frac{8\zeta^2}{3L^2(N-4)} + \frac{B\eta\Psi}{6LN(1-BL\eta\Psi)} \mathbb{E}_{\cdot|\mathcal{Q}} \Big[\Big\| \nabla f_i(\theta_{t_0}^{(i)}) \Big\|^2 \Big] \right. \\
&\qquad \left. + \frac{B^2N^2\eta^2\sigma^2\Psi^2}{1-6BLN\eta\Psi} \right] \\
&= 2\mathbb{E}_{\cdot|\mathcal{Q}} \Big[\sum_{n \neq j} q_\tau^{n \to j} \| \mathbf{y}_{\tau,b}^{(n)} - \theta_\tau^{(n)} \|^2 \Big] + \frac{8\zeta^2}{L^2(N-4)} + \frac{B\eta\Psi}{2LN(1-BL\eta\Psi)} \mathbb{E}_{\cdot|\mathcal{Q}} \Big[\Big\| \nabla f_i(\theta_{t_0}^{(i)}) \Big\|^2 \Big] \\
&\quad + \frac{3B^2N^2\eta^2\sigma^2\Psi^2}{1-6BLN\eta\Psi}, \qquad \text{(B.5)}
\end{aligned}$$

where (i) $\mu_1 = \frac{1}{6BLN\gamma\Psi}$ and $c = \frac{1}{BL\gamma\Psi}$ are applied. Additionally, if $\Psi > 0$, $\gamma \leq \frac{1}{8BLN\Psi}$ is the tighter upper bound than $\gamma \leq \frac{1}{8BL}$. With this remark, the upper bound in inequality B.5 can be simplified even more as

$$\begin{aligned}
&\mathbb{E}_{\cdot|\mathcal{Q}} \Big[\sum_{j \neq i} q_t^{j \to i} \Big\| \theta_t^{(j)} - \theta_{t_0}^{(j)} \Big\|^2 \Big] \\
&\leq 2\mathbb{E}_{\cdot|\mathcal{Q}} \Big[\sum_{n \neq j} q_\tau^{n \to j} \| \mathbf{y}_{\tau,b}^{(n)} - \theta_\tau^{(n)} \|^2 \Big] + \frac{8\zeta^2}{L^2(N-4)} \\
&\quad + \frac{\frac{1}{8LN}}{2LN(1-\frac{1}{8N})} \mathbb{E}_{\cdot|\mathcal{Q}} \Big[\Big\| \nabla f_i(\theta_{t_0}^{(i)}) \Big\|^2 \Big] + \frac{3B^2N^2\sigma^2\Psi^2 \cdot \frac{1}{64B^2L^2N^2\Psi^2}}{1 - \frac{6BLN\Psi}{8BLN\Psi}} \\
&\leq 2\mathbb{E}_{\cdot|\mathcal{Q}} \Big[\sum_{n \neq j} q_\tau^{n \to j} \| \mathbf{y}_{\tau,b}^{(n)} - \theta_\tau^{(n)} \|^2 \Big] + \frac{8\zeta^2}{L^2(N-4)} \\
&\quad + \frac{1}{2L^2N(8N-1)} \mathbb{E}_{\cdot|\mathcal{Q}} \Big[\Big\| \nabla f_i(\theta_{t_0}^{(i)}) \Big\|^2 \Big] + \frac{3\sigma^2}{16L^2} \\
&\leq 2\mathbb{E}_{\cdot|\mathcal{Q}} \Big[\sum_{n \neq j} q_\tau^{n \to j} \| \mathbf{y}_{\tau,b}^{(n)} - \theta_\tau^{(n)} \|^2 \Big] + \frac{8\zeta^2}{L^2(N-4)} \\
&\quad + \frac{1}{16L^2N^2} \mathbb{E}_{\cdot|\mathcal{Q}} \Big[\Big\| \nabla f_i(\theta_{t_0}^{(i)}) \Big\|^2 \Big] + \frac{3\sigma^2}{16L^2} \\
&\overset{(a)}{\leq} 2\mathbb{E}_{\cdot|\mathcal{Q}} \Big[\sum_{j \neq i} q_t^{j \to i} \| \mathbf{y}_{\tau,b}^{(j)} - \theta_\tau^{(j)} \|^2 \Big] + \frac{8\zeta^2}{L^2(N-4)} \\
&\quad + \frac{1}{16L^2N^2} \mathbb{E}_{\cdot|\mathcal{Q}} \Big[\Big\| \nabla f_i(\theta_{t_0}^{(i)}) \Big\|^2 \Big] + \frac{3\sigma^2}{16L^2}, \qquad \text{(B.6)}
\end{aligned}$$

where (a) is satisfied without loss of generality. □

Remark. This proposition appears in Lemma B.3, which is then used at the second term of inequality B.10.

B.1.3 Lemmas

In collaborative learning, local computations often occur more frequently than communication. This is to avoid duplicating transmissions, which can occur in the reverse scenario.

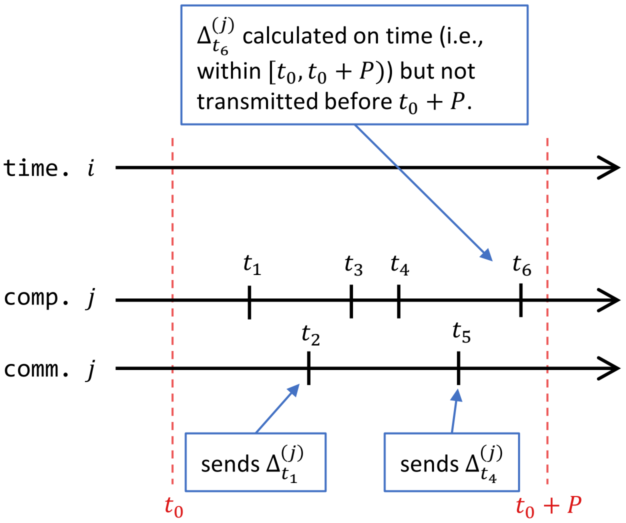

$\Delta_{t_6}^{(j)}$ calculated on time (i.e., within $[t_0, t_0+P)$) but not transmitted before t_0+P.
time. i
t_1
t_3
t_4
t_6
comp. j
t_2
t_5
comm. j
sends $\Delta_{t_1}^{(j)}$
sends $\Delta_{t_4}^{(j)}$
t_0
t_0+P

Figure B.2. The latest local update of j is unable to be transmitted within the given range $[t_0, t_0+P)$ because of the independence between computation timestamps and communication (transmission) timestamps.

Lemma B.1. *For all $n \in \mathcal{U}$, there are no fewer $\mathbf{y}_{t,b}^{(n)}$ than $\mathbf{y}_{\lfloor t\rfloor,b}^{(n)}$ within any given range of time $\{t | t \in [t_0, t_0+P)\}$. In other words, it also satisfies that*

$$\left\| \int_P \sum_{b=0}^{B-1} \mathbf{g}_n(\mathbf{y}_{\lfloor t\rfloor,b}^{(n)})\,\mathrm{d}t \right\|^2 \leq \left\| \int_P \sum_{b=0}^{B-1} \mathbf{g}_n(\mathbf{y}_{t,b}^{(n)})\,\mathrm{d}t \right\|^2. \tag{B.7}$$

Proof. In Fig. (B.2), the value of $\Delta_{t_4}^{(j)}$ can differ from $\Delta_{t_3}^{(j)}$ if another node transmits a message to node j, thereby affecting the value of $\theta^{(j)}$. To facilitate our analysis, we assume that each user creates a backup of the non transmitted local updates for the upcoming transmission event. Returning to the scenario depicted in Fig. (B.2), based on this assumption, user j sends both $\Delta_{t_3}^{(j)}$ and $\Delta_{t_4}^{(j)}$ to user i at the earliest transmission event time, which is t_5. □

Lemma B.2. *(Upper bound for superpositioned model deviations.) For all $i, j \in \mathcal{U}$, when $\eta \leq \min(\frac{1}{8BL}, \frac{1}{8BLN\Psi})$, we have*

$$\begin{aligned}&\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\big[\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)}\|^2\big] \\ &\leq \frac{16\zeta^2}{L^2(N-4)} + \frac{3\sigma^2}{8L^2} + \frac{9N\zeta^2}{2L^2} + 16B\eta^2\sigma^2 + \Big(\frac{1}{8L^2N} + 128B^2\eta^2\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\nabla f_i(\theta_{t_0}^{(i)})\Big\|^2\Big]\,.\end{aligned}$$

Proof. Proposition B.2 and Proposition B.3 can be interpreted as a system of linear inequalities. Applying (a) Proposition B.2 and (b) Proposition B.3 respectively, we get

$$\begin{aligned}&\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\big[\|\mathbf{y}_{t,b}^{(j)} - \theta_t^{(j)}\|^2\big] \\ &\overset{(a)}{\leq} \frac{2}{5}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\theta_t^{(i)} - \theta_{t_0}^{(i)}\Big\|^2\Big] + \frac{9N\zeta^2}{10L^2} + \frac{16B\eta^2\sigma^2}{5} + \frac{128B^2\eta^2}{5}\mathbb{E}_{\cdot|\mathcal{Q}}\big[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2\big]\end{aligned}$$

$$
\begin{aligned}
&\overset{(b)}{\leq} \frac{2}{5}\Big(2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\|\mathbf{y}_{t,b}^{(j)}-\theta_t^{(j)}\|^2\Big]+\frac{8\zeta^2}{L^2(N-4)}+\frac{1}{16L^2N^2}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big]+\frac{3\sigma^2}{16L^2}\Big)\\
&\quad+\frac{9N\zeta^2}{10L^2}+\frac{16B\eta^2\sigma^2}{5}+\frac{128B^2\eta^2}{5}\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2]\\
&=\frac{4}{5}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\mathbf{y}_{t,b}^{(j)}-\theta_t^{(j)}\big\|^2\Big]+\frac{16\zeta^2}{5L^2(N-4)}+\frac{1}{40L^2N}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big]+\frac{3\sigma^2}{40L^2}\\
&\quad+\frac{9N\zeta^2}{10L^2}+\frac{16B\eta^2\sigma^2}{5}+\frac{128B^2\eta^2}{5}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big],
\end{aligned}
$$

and therefore,

$$
\begin{aligned}
&\frac{1}{5}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}[\|\mathbf{y}_{t,b}^{(j)}-\theta_t^{(j)}\|^2]\\
&\leq \frac{16\zeta^2}{5L^2(N-4)}+\frac{3\sigma^2}{40L^2}+\frac{9N\zeta^2}{10L^2}+\frac{16B\eta^2\sigma^2}{5}+\Big(\frac{1}{40L^2N}+\frac{128B^2\eta^2}{5}\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big].
\end{aligned}
$$

□

Lemma B.3. *(Upper bound for the local reference model change) When Assumption 3.2 holds true during $[t_0, t)$ for all users (i.e., when the number of events during the given period $[t_0, t)$ is finite) and $\eta \leq \min(\frac{1}{8BL}, \frac{1}{8BLN\Psi})$, we have*

$$
\begin{aligned}
&\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_t^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big]\\
&\leq \frac{9N\zeta^2}{L^2}+32B\eta^2\sigma^2+\frac{40\zeta^2}{L^2(N-4)}+\frac{15\sigma^2}{16L^2}+\Big(256B^2\eta^2+\frac{5}{16L^2N^2}\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big].
\end{aligned}
$$

Proof. Approaching in the same fashion of proving as in Lemma B.2, we have

$$
\begin{aligned}
&\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_t^{(j)}-\theta_{t_0}^{(j)}\big\|^2\Big]\\
&\overset{(a)}{\leq} 2\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\|\mathbf{y}_{t,b}^{(j)}-\theta_t^{(j)}\|^2\Big]+\frac{8\zeta^2}{L^2(N-4)}+\frac{1}{16L^2N^2}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big]+\frac{3\sigma^2}{16L^2}\\
&\overset{(b)}{\leq} 2\Big(\frac{2}{5}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)}-\theta_{t_0}^{(i)}\big\|^2\Big]+\frac{9N\zeta^2}{10L^2}+\frac{16B\eta^2\sigma^2}{5}+\frac{128B^2\eta^2}{5}\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2]\Big)\\
&\quad+\frac{8\zeta^2}{L^2(N-4)}+\frac{1}{16L^2N^2}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big]+\frac{3\sigma^2}{16L^2}\\
&=\frac{4}{5}\sum_{j\neq i} q_t^{j\to i}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\theta_t^{(i)}-\theta_{t_0}^{(i)}\big\|^2\Big]+\frac{9N\zeta^2}{5L^2}+\frac{32B\eta^2\sigma^2}{5}+\frac{8\zeta^2}{L^2(N-4)}+\frac{3\sigma^2}{16L^2}\\
&\quad+\Big(\frac{256B^2\eta^2}{5}+\frac{1}{16L^2N^2}\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big],
\end{aligned}
$$

where (a) uses Proposition B.3, and (b) comes from Lemma B.2. □

B.1.4 Proof of Theorem 3.1

The proof of Theorem 3.1 is based on the proof provided in [94].

Beginning with rephrasing the L-smoothness between $f_i(\theta^{(i)}_{t_0+P})$ and $f_i(\theta^{(i)}_{t_0})$, we have

$$
\begin{aligned}
&\mathbb{E}_{\cdot|\mathcal{Q},t_0}[f_i(\theta^{(i)}_{t_0+P})] \\
&\leq f_i(\theta^{(i)}_{t_0}) + \mathbb{E}_{\cdot|\mathcal{Q},t_0}[\langle \nabla f_i(\theta^{(i)}_{t_0}),\ \theta^{(i)}_{t_0+P} - \theta^{(i)}_{t_0}\rangle] + \frac{L}{2}\mathbb{E}_{\cdot|\mathcal{Q},t_0}\Big[\big\|\theta^{(i)}_{t_0+P} - \theta^{(i)}_{t_0}\big\|^2\Big] \\
&\leq f_i(\theta^{(i)}_{t_0}) - \eta\Big\langle \nabla f_i(\theta^{(i)}_{t_0}), \mathbb{E}_{\cdot|\mathcal{Q},t_0}\Big[\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\mathbf{g}_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\Big]\Big\rangle \\
&\quad + \frac{\eta^2 L}{2}\mathbb{E}_{\cdot|\mathcal{Q},t_0}\Big[\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\mathbf{g}_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\Big\|^2\Big] \\
&= f_i(\theta^{(i)}_{t_0}) - \eta\Big\langle \nabla f_i(\theta^{(i)}_{t_0}), \mathbb{E}_{\cdot|\mathcal{Q},t_0}\Big[\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\mathbb{E}\big[\mathbf{g}_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})|\mathcal{Q},\mathbf{y}^{(j)}_{\lfloor t\rfloor,b},\theta^{(i)}_{t_0}\big]\,\mathrm{d}t\Big]\Big\rangle \\
&\quad + \frac{\eta^2 L}{2}\mathbb{E}_{\cdot|\mathcal{Q},t_0}\Big[\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\mathbf{g}_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\Big\|^2\Big] \\
&= f_i(\theta^{(i)}_{t_0}) - \eta\Big\langle \nabla f_i(\theta^{(i)}_{t_0}), \mathbb{E}_{\cdot|\mathcal{Q},t_0}\Big[\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\Big]\Big\rangle \\
&\quad + \frac{\eta^2 L}{2}\mathbb{E}_{\cdot|\mathcal{Q},t_0}\Big[\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\mathbf{g}_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\Big\|^2\Big]
\end{aligned}
$$

Taking expectation on both sides over $\theta^{(i)}_{t_0}$, we obtain

$$
\begin{aligned}
\mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta^{(i)}_{t_0+P})] \leq \mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta^{(i)}_{t_0})] - \eta\mathbb{E}_{\cdot|\mathcal{Q},t_0}\Big[\Big\langle \nabla f_i(\theta^{(i)}_{t_0}), \int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\Big\rangle\Big] \\
+ \frac{\eta^2 L}{2}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\mathbf{g}_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\Big\|^2\Big]
\end{aligned}
\tag{B.8}
$$

Here, we reintroduce a finite variable from Definition 3.1, $\Psi \in \mathbb{R}^+$, to indicate the maximum total number of all message exchanging events during the time period $[t_0, t_0 + P)$. We set an assumption that $\Psi \geq 3$ for any time elapse $[t_0, t_0 + P)$ in which t_0 is multiple to P.

Considering the second term in the inequality B.8,

$$
\begin{aligned}
&-\Big\langle \nabla f_i(\theta^{(i)}_{t_0}), \int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\Big\rangle \\
&= -\frac{1}{B\Psi}\Big\langle B\Psi\nabla f_i(\theta^{(i)}_{t_0}), \int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\Big\rangle
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{2B\Psi}\Big\| B\Psi\nabla f_i(\theta^{(i)}_{t_0}) - \int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\,\Big\|^2 \\
&\quad - \frac{B\Psi}{2}\|\nabla f_i(\theta^{(i)}_{t_0})\|^2 - \frac{1}{2B\Psi}\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\,\Big\|^2 \\
&= \frac{1}{2B\Psi}\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\big[\nabla f_i(\theta^{(i)}_{t_0}) - \nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\big]\,\mathrm{d}t\,\Big\|^2 \\
&\quad - \frac{B\Psi}{2}\|\nabla f_i(\theta^{(i)}_{t_0})\|^2 - \frac{1}{2B\Psi}\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\,\Big\|^2 \\
&= \frac{1}{2B\Psi}\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}[\nabla f_i(\theta^{(i)}_{t_0}) - \nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})]\,\mathrm{d}t\,\Big\|^2 \\
&\quad - \frac{B\Psi}{2}\|\nabla f_i(\theta^{(i)}_{t_0})\|^2 - \frac{1}{2B\Psi}\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\,\Big\|^2
\end{aligned}
$$

In order to deal with two variables controlled by different agents, two terms are added and subtracted for further proof: local model gradient calculated by j and its local reference model, respectively.

$$
\begin{aligned}
&= \frac{1}{2B\Psi}\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\big[\nabla f_i(\theta^{(i)}_{t_0}) - \nabla f_j(\theta^{(j)}_{t_0}) + \nabla f_j(\theta^{(j)}_{t_0}) - \nabla f_j\big(\theta^{(j)}_{t}\big) + \nabla f_j\big(\theta^{(j)}_{t}\big) - \nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\big]\,\mathrm{d}t\,\Big\|^2 \\
&\quad - \frac{B\Psi}{2}\|\nabla f_i(\theta^{(i)}_{t_0})\|^2 - \frac{1}{2B\Psi}\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\,\Big\|^2 \\
&\le \frac{3}{2B\Psi}\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\big[\nabla f_j\big(\theta^{(j)}_{t}\big) - \nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\big]\,\mathrm{d}t\,\Big\|^2 \\
&\quad + \frac{3}{2B\Psi}\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\big[\nabla f_j(\theta^{(j)}_{t_0}) - \nabla f_j\big(\theta^{(j)}_{t}\big)\big]\,\mathrm{d}t\,\Big\|^2 \\
&\quad + \frac{3}{2B\Psi}\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}[\nabla f_i(\theta^{(i)}_{t_0}) - \nabla f_j(\theta^{(j)}_{t_0})]\,\mathrm{d}t\,\Big\|^2 \\
&\quad - \frac{B\Psi}{2}\|\nabla f_i(\theta^{(i)}_{t_0})\|^2 - \frac{1}{2B\Psi}\Big\|\int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\,\Big\|^2 \\
&\overset{(a)}{\le} \frac{3}{2B\Psi}\Big\| B\Psi\sum_{j\neq i} q_t^{j\to i}\big[\nabla f_j\big(\theta^{(j)}_{t}\big) - \nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\big]\Big\|^2 \\
&\quad + \frac{3}{2B\Psi}\Big\| B\Psi\sum_{j\neq i} q_t^{j\to i}\big[\nabla f_j(\theta^{(j)}_{t_0}) - \nabla f_j\big(\theta^{(j)}_{t}\big)\big]\Big\|^2 \\
&\quad + \frac{3}{2B\Psi}\Big\| B\Psi\sum_{j\neq i} q_t^{j\to i}[\nabla f_i(\theta^{(i)}_{t_0}) - \nabla f_j(\theta^{(j)}_{t_0})]\Big\|^2
\end{aligned}
$$

$$
\begin{aligned}
&- \frac{B\Psi}{2}\|\nabla f_i(\theta_{t_0}^{(i)})\|^2 - \frac{1}{2B\Psi}\Big\| \int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\Big\|^2\\
&\overset{(b)}{\leq} \frac{3BN\Psi}{2}\sum_{j\neq i} q_t^{j\to i}\Big\|\nabla f_j\big(\theta_t^{(j)}\big) - \nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\Big\|^2\\
&\quad + \frac{3BN\Psi}{2}\sum_{j\neq i} q_t^{j\to i}\Big\|\nabla f_j(\theta_{t_0}^{(j)}) - \nabla f_j\big(\theta_t^{(j)}\big)\Big\|^2\\
&\quad + \frac{3B\Psi}{2}\Big\|\sum_{j\neq i} q_t^{j\to i}\big[\nabla f_i(\theta_{t_0}^{(i)}) - \nabla f_j(\theta_{t_0}^{(j)})\big]\Big\|^2\\
&\quad - \frac{B\Psi}{2}\|\nabla f_i(\theta_{t_0}^{(i)})\|^2 - \frac{1}{2B\Psi}\Big\| \int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\Big\|^2\\
&\overset{(c)}{\leq} \frac{3BL^2N\Psi}{2}\sum_{j\neq i} q_t^{j\to i}\big\|\theta_t^{(j)} - \mathbf{y}_{\lfloor t\rfloor,b}^{(j)}\big\|^2 + \frac{3BL^2N\Psi}{2}\sum_{j\neq i} q_t^{j\to i}\big\|\theta_{t_0}^{(j)} - \theta_t^{(j)}\big\|^2 + \frac{3BN\Psi\zeta^2}{N-4}\\
&\quad - \frac{B\Psi}{2}\|\nabla f_i(\theta_{t_0}^{(i)})\|^2 - \frac{1}{2B\Psi}\Big\| \int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\Big\|^2 \qquad \text{(B.9)}
\end{aligned}
$$

where (a) reflects Jensen's inequality on the first three terms, pulling the terms out of the L2 norms; (b) is valid because $\|\sum_{j=1}^{N} q(j)\mathbf{z}(j)\|^2 \leq N\sum_{j=1}^{N} q(j)\|\mathbf{z}(j)\|^2$ for all $q_\star \in [0,1]$; (c) L-smoothness on the first two terms and Lemma 3.1 on the third term.

Hence, the expectation can be bounded as follows:

$$
\begin{aligned}
&\mathbb{E}_{\cdot|\mathcal{Q}}\Big[-\Big\langle \nabla f_i(\theta_{t_0}^{(i)}), \int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\Big\rangle\Big]\\
&\leq \frac{3BL^2N\Psi}{2}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_t^{(j)} - \mathbf{y}_{\lfloor t\rfloor,b}^{(j)}\big\|^2\Big]\\
&\quad + \frac{3BL^2N\Psi}{2}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{j\neq i} q_t^{j\to i}\big\|\theta_{t_0}^{(j)} - \theta_t^{(j)}\big\|^2\Big] + \frac{3BN\Psi\zeta^2}{N-4}\\
&\quad - \frac{B\Psi}{2}\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2] - \frac{1}{2B\Psi}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\| \int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\Big\|^2\Big]\\
&\overset{(a)}{\leq} \frac{3BL^2N\Psi}{2}\Big[\frac{16\zeta^2}{L^2(N-4)} + \frac{3\sigma^2}{8L^2} + \frac{9N\zeta^2}{2L^2} + 16B\eta^2\sigma^2 + \Big(\frac{1}{8L^2N} + 128B^2\eta^2\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big]\Big]\\
&\quad + \frac{3BL^2N\Psi}{2}\Big[\frac{9N\zeta^2}{L^2} + 32B\eta^2\sigma^2 + \frac{40\zeta^2}{L^2(N-4)} + \frac{15\sigma^2}{16L^2} + \Big(256B^2\eta^2 + \frac{5}{16L^2N^2}\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big]\Big]\\
&\quad + \frac{3BN\Psi\zeta^2}{N-4} - \frac{B\Psi}{2}\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2] - \frac{1}{2B\Psi}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\| \int_P \sum_{j\neq i} q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\Big\|^2\Big]\\
&= \frac{3BL^2N\Psi}{2}\Big[\frac{56\zeta^2}{L^2(N-4)} + \frac{21\sigma^2}{16L^2} + \frac{27N\zeta^2}{2L^2} + 48B\eta^2\sigma^2 + \Big(\frac{7}{16L^2N} + 384B^2\eta^2\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big]\Big]
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{3BN\Psi\zeta^2}{N-4}-\frac{B\Psi}{2}\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta_{t_0}^{(i)})\|^2]-\frac{1}{2B\Psi}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\int_P\sum_{j\neq i}q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\,\Big\|^2\Big]\\
=&\frac{87BN\zeta^2\Psi}{N-4}+\frac{63BN\sigma^2\Psi}{32}+\frac{81BN^2\zeta^2\Psi}{4}+72B^2L^2N\eta^2\sigma^2\Psi\\
&+B\Psi\Big(\frac{21}{32N}+576B^2L^2N\eta^2-\frac{1}{2}\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\nabla f_i(\theta_{t_0}^{(i)})\Big\|^2\Big]\\
&-\frac{1}{2B\Psi}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\int_P\sum_{j\neq i}q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\,\Big\|^2\Big]
\end{aligned}\tag{B.10}
$$

where (a) uses Lemma B.2, Lemma B.3, and Lemma 3.1 on the first three terms, respectively.

Considering the third term in the inequality B.8,

$$
\begin{aligned}
&\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\int_P\sum_{j\neq i}q_t^{j\to i}\sum_{b=0}^{B-1}\mathbf{g}_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\,\Big\|^2\Big]\\
&=\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\int_P\sum_{j\neq i}q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\,\Big\|^2\Big]\\
&\quad+\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\int_P\sum_{j\neq i}q_t^{j\to i}\sum_{b=0}^{B-1}[\mathbf{g}_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})-\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})]\,\mathrm{d}t\,\Big\|^2\Big]\\
&=\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\int_P\sum_{j\neq i}q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\,\Big\|^2\Big]\\
&\quad+\int_P\sum_{j\neq i}(q_t^{j\to i})^2\sum_{b=0}^{B-1}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\|\mathbf{g}_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})-\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\|^2\Big]\,\mathrm{d}t\\
&\overset{(a)}{\leq}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\int_P\sum_{j\neq i}q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\,\Big\|^2\Big]+B\rho^2\sigma^2
\end{aligned}\tag{B.11}
$$

where (a) is derived from the definition of σ in Assumption 3.4 and ρ.

Plugging B.10 and B.11, the inequality B.8 is rephrased as:

$$
\begin{aligned}
&\mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta_{t_0+P}^{(i)})]\\
&\leq\mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta_{t_0}^{(i)})]+\eta\Bigg[\frac{87BN\zeta^2\Psi}{N-4}+\frac{63BN\sigma^2\Psi}{32}+\frac{81BN^2\zeta^2\Psi}{4}+72B^2L^2N\eta^2\sigma^2\Psi\\
&\quad+B\Psi\Big(\frac{21}{32N}+576B^2L^2N\eta^2-\frac{1}{2}\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\nabla f_i(\theta_{t_0}^{(i)})\Big\|^2\Big]\\
&\quad-\frac{1}{2B\Psi}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\int_P\sum_{j\neq i}q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\,\Big\|^2\Big]\Bigg]\\
&\quad+\frac{\eta^2L}{2}\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\int_P\sum_{j\neq i}q_t^{j\to i}\sum_{b=0}^{B-1}\nabla f_j(\mathbf{y}_{\lfloor t\rfloor,b}^{(j)})\,\mathrm{d}t\,\Big\|^2\Big]+\frac{BL\eta^2\rho^2\sigma^2}{2}
\end{aligned}
$$

$$
\begin{aligned}
&\leq \mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta^{(i)}_{t_0})] + \frac{87BN\eta\zeta^2\Psi}{N-4} + \frac{63BN\eta\sigma^2\Psi}{32} + \frac{81BN^2\eta\zeta^2\Psi}{4} + 72B^2L^2N\eta^3\sigma^2\Psi \\
&\quad + B\eta\Psi\Big(\frac{21}{32N} + 576B^2L^2N\eta^2 - \frac{1}{2}\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta^{(i)}_{t_0})\big\|^2\Big] \\
&\quad + \Big(\frac{\eta^2L}{2} - \frac{\eta}{2B\Psi}\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\int_P \sum_{j\neq i} q_t^{j\to i} \sum_{b=0}^{B-1} \nabla f_j(\mathbf{y}^{(j)}_{\lfloor t\rfloor,b})\,\mathrm{d}t\,\Big\|^2\Big] + \frac{BL\eta^2\rho^2\sigma^2}{2} \\
&\overset{(a)}{\leq} \mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta^{(i)}_{t_0})] + \frac{87BN\eta\zeta^2\Psi}{N-4} + \frac{63BN\eta\sigma^2\Psi}{32} + \frac{81BN^2\eta\zeta^2\Psi}{4} + 72B^2L^2N\eta^3\sigma^2\Psi \\
&\quad + B\eta\Psi\Big(\frac{21}{32N} + 576B^2L^2N\eta^2 - \frac{1}{2}\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta^{(i)}_{t_0})\big\|^2\Big] + \frac{BL\eta^2\rho^2\sigma^2}{2} \\
&\overset{(b)}{\leq} \mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta^{(i)}_{t_0})] + \frac{87BN\eta\zeta^2\Psi}{N-4} + \frac{63BN\eta\sigma^2\Psi}{32} + \frac{81BN^2\eta\zeta^2\Psi}{4} + 72B^2L^2N\eta^3\sigma^2\Psi \\
&\quad + B\eta\Psi\Big(\frac{21}{32N} + \frac{9}{N\Psi^2} - \frac{1}{2}\Big)\mathbb{E}_{\cdot|\mathcal{Q}}\Big[\big\|\nabla f_i(\theta^{(i)}_{t_0})\big\|^2\Big] + \frac{BL\eta^2\rho^2\sigma^2}{2},
\end{aligned}
\tag{B.12}
$$

where (a) negates the term including $\nabla f_j(\mathbf{y}^{(j)}_{t,b})$ because $\frac{\eta^2 L}{2} - \frac{\eta}{2B\Psi} < 0$ based on the upper bound of η; (b) bounds the coefficient of the term including $\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta^{(i)}_{t_0})\|^2]$ to simplify the further analysis.

After rearrangement, we have

$$
\begin{aligned}
&\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f_i(\theta^{(i)}_{t_0})\|^2] \\
&\leq \frac{\mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta^{(i)}_{t_0})] - \mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta^{(i)}_{t_0+P})]}{B\eta\Psi\Big(\frac{1}{2} - \frac{21}{32N} - \frac{9}{N\Psi^2}\Big)} + \frac{1}{\frac{1}{2} - \frac{21}{32N} - \frac{9}{N\Psi^2}}\Big(\frac{87N\zeta^2}{N-4} + \frac{63N\sigma^2}{32} + \frac{81N^2\zeta^2}{4} \\
&\quad + 72BL^2N\eta^2\sigma^2 + \frac{L\eta\rho^2\sigma^2}{2\Psi}\Big) \\
&\overset{(a)}{\leq} \frac{128\big(\mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta^{(i)}_{t_0})] - \mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta^{(i)}_{t_0+P})]\big)}{11B\eta\Psi} + \frac{128}{11}\Big(\frac{87N\zeta^2}{N-4} + \frac{63N\sigma^2}{32} + \frac{81N^2\zeta^2}{4} \\
&\quad + 72BL^2N\eta^2\sigma^2 + \frac{L\eta\rho^2\sigma^2}{2\Psi}\Big) \\
&= \frac{128\big(\mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta^{(i)}_{t_0})] - \mathbb{E}_{\cdot|\mathcal{Q}}[f_i(\theta^{(i)}_{t_0+P})]\big)}{11B\eta\Psi} + \frac{11136N\zeta^2}{11(N-4)} + \frac{252N\sigma^2}{11} + \frac{2592N^2\zeta^2}{11} \\
&\quad + 9216BL^2N\eta^2\sigma^2 + \frac{64L\eta\rho^2\sigma^2}{11\Psi}\,,
\end{aligned}
\tag{B.13}
$$

where (a) makes the denominator smaller than the derived upper bound of inequality B.12 by using $N > 4$ and $\Psi \geq 3$, resulting in

$$
\frac{1}{\frac{1}{2} - \frac{21}{32N} - \frac{9}{N\Psi^2}} \leq \frac{1}{\frac{1}{2} - \frac{21}{32\cdot 4} - \frac{9}{4\cdot 3^2}} = \frac{128}{11}.
$$

Finally, the minimum value of $\mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f(\theta_t)\|^2]$ over time t can be found as:

$$
\begin{aligned}
&\min_t \mathbb{E}_{\cdot|\mathcal{Q}}[\|\nabla f(\theta_t)\|^2] \\
&= \min_t \mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(\theta_t^{(i)})\Big\|^2\Big] \\
&\leq \min_{t_0\in\{0,P,\cdots,(\lfloor \frac{T}{P}\rfloor-1)P\}} \mathbb{E}_{\cdot|\mathcal{Q}}\Big[\Big\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(\theta_{t_0}^{(i)})\Big\|^2\Big] \\
&\overset{(a)}{\leq} \min_{t_0\in\{0,P,\cdots,(\lfloor \frac{T}{P}\rfloor-1)P\}} \frac{1}{N}\cdot \mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{i=1}^{N}\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big] \\
&\leq \frac{1}{N\lfloor \frac{T}{P}\rfloor}\cdot \sum_{t_0=0,P,2P,\cdots,(\lfloor \frac{T}{P}\rfloor-1)P} \mathbb{E}_{\cdot|\mathcal{Q}}\Big[\sum_{i=1}^{N}\big\|\nabla f_i(\theta_{t_0}^{(i)})\big\|^2\Big] \\
&\overset{(b)}{\leq} \frac{1}{N\lfloor \frac{T}{P}\rfloor}\sum_{i=1}^{N}\Bigg[\frac{128\big(f_i(\theta_0^{(i)})-f_i^*\big)}{11B\eta\Psi}\Bigg] + \frac{11136\zeta^2}{11(N-4)} + \frac{252\sigma^2}{11} + \frac{2592N\zeta^2}{11} \\
&\quad + 9216BL^2\eta^2\sigma^2 + \frac{64L\eta\rho^2\sigma^2}{11N\Psi} \\
&\overset{(c)}{\leq} \frac{128}{11B\eta\Psi}(f(\theta_0)-f^*) + \frac{11136\zeta^2}{11(N-4)} + \frac{252\sigma^2}{11} + \frac{2592N\zeta^2}{11} \\
&\quad + 9216BL^2\eta^2\sigma^2 + \frac{64L\eta\rho^2\sigma^2}{11N\Psi} \\
&= \mathcal{O}\Big(\frac{\mathcal{F}}{B\eta\Psi} + \frac{\zeta^2}{N-4} + \sigma^2 + N\zeta^2 + BL^2\eta^2\sigma^2 + \frac{L\eta\rho^2\sigma^2}{N\Psi}\Big)
\end{aligned}
$$

where (a) is due to Jensen's inequality; the first term of (b) is an implantation of inequality B.13 whereas the other terms are independent on t_0; (c) takes that $P \leq T$ and the definition of $f(\theta_\star)$

B.2 Additional experiment results

In this section, we investigate the effect of (i) topology and (ii) wireless channels on DRACO, which is not considered separately in the main contents of this thesis. The first case, which involves a time-invariant connectivity graph, covers scenarios in which the topology is fixed throughout the learning process. For any two user nodes connected to an edge, a message sent from one node is always successfully received at the other. The physical distance (i.e., geographical coordinates) is not considered. Under this setting, we study the impact of the frequency of successfully received messages and the characteristics of the connectivity graphs. On the other hand, in the second case, the connectivity graph changes over time. Here, user nodes are positioned randomly with coordinates following uniform distribution. Their information exchange is influenced by factors such as interference and channel capacity limitations. Differentiating from the fixed topology scenarios, we mainly verify the influence of the transmission duration deadline, superposition window, and unification period accounting in general for all wireless channel conditions.

Experiments were conducted on image classification tasks over serverless networks using the MNIST dataset [149]. Each user possesses 50 local training samples arranged into training batches with 15 samples per batch. The default number of participants in each simulation is $N = 25$, unless otherwise specified. The sampling interval was set at 500 events, meaning the evaluation of each local model occurs after the completion of the 500th event. The rate parameter of the exponential distribution in local gradient computation is $\lambda_i = 1$ for all users by default. This study does not evaluate the impact of model compression, assuming that the packet size is equivalent to the raw model size. The CNN architecture employed in the simulations occupies 596776 B (0.57 MB). This value serves as the basis for quantifying the message size in time-variant $\mathcal{Q}$ cases.

We conducted simulations under two different scenarios according to the time-variability of $\mathcal{Q}$. In time-invariant cases (see Section B.2.1), the connectivity graph is fixed throughout the whole collaboration process. Each user, indexed i without losing generality, spends some time computing local gradient following $Exp(\lambda_i)$ as mentioned in Assumption 3.1. Whenever a local update is done at t, user i sends $\Delta_t^{(i)}$ to its neighbors $j \in \mathcal{N}(i)$, where $\mathcal{N}(i)$ indicates a set of user i's neighbors. Slightly after that moment (i.e., at time $t + t_\epsilon$), $q_{t+t_\epsilon}^{i\to j} = 1$ for all j.

Meanwhile, in time-variant cases (see Section B.2.2), we also consider communication over wireless channels in which nodes are randomly connected over time without following any predetermined scheduling policies. The readers may refer to Section 3.5 of the thesis for detailed hyperparameter settings.

B.2.1 Comparison within fixed topology

Since delivery is always successful in fixed $\mathcal{Q}$ scenarios, the participants do not experience performance degradation owing to wireless communication impairments. We use by default $\lambda_i = 0.1$ for all $i \in \mathcal{U}$ as the rate parameter that determines local computation time.

To evaluate the effect of topology on the performance of DRACO, we compare five graph types: complete, cycle, bipartite, star, and k-nearest neighbor graph (k-NNG), as depicted in Fig. B.3. We manually fix $k = 3$ for k-NNG.[2] Among these graph types, a fully connected network reaches the highest test accuracy and the fastest convergence, thanks to doubly stochastic connection graphs with accessibility to all devices. This difference in performance is attributed to each topology having a different number of neighbors, which affects the reception frequency ψ_i per unification period.

We regulate the number of reception events per user by randomly removing exchange events in each unit time $[mP, (m+1)P)$ for $m = 0, 1, \cdots, \lfloor T/P \rfloor - 1$, ensuring that each user receives no more than Ψ messages during that period. As shown in Fig. B.4, a larger Ψ reaches smaller training losses, which is consistent with the results in [120] and our Theorem 3.1. We find that the average test accuracy saturates on a higher point also in cases with larger Ψ. Hence, it is meaningful to determine Ψ based on which one between final test accuracy and saturating speed weighs more in the experiments.

[2] The parameter k in k-NNG is the number of edges of each node connected in the nearest order. It shall not be confused with k for indexing an event as used in Section 3.2.2 and Alg. 5 of this chapter.

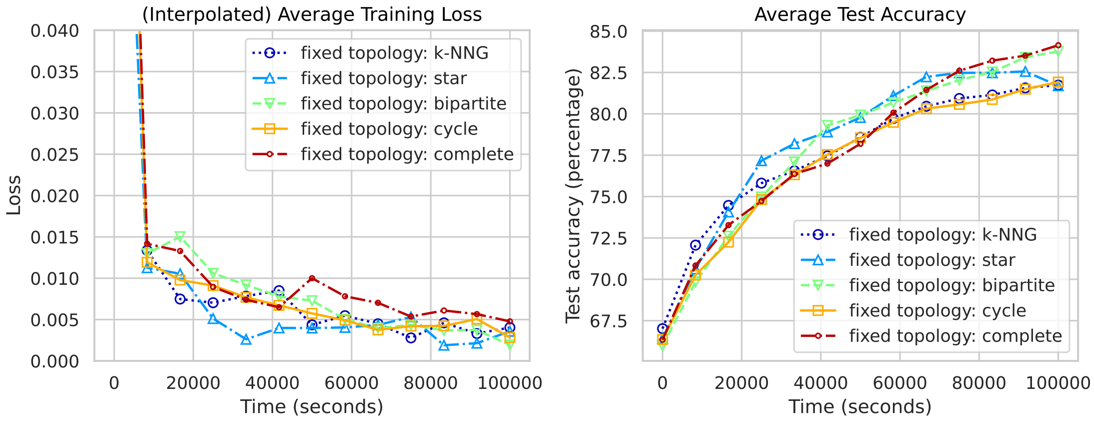


Figure B.3. Results with respect to different network topology ($\Gamma_{\max} = 0.5$)

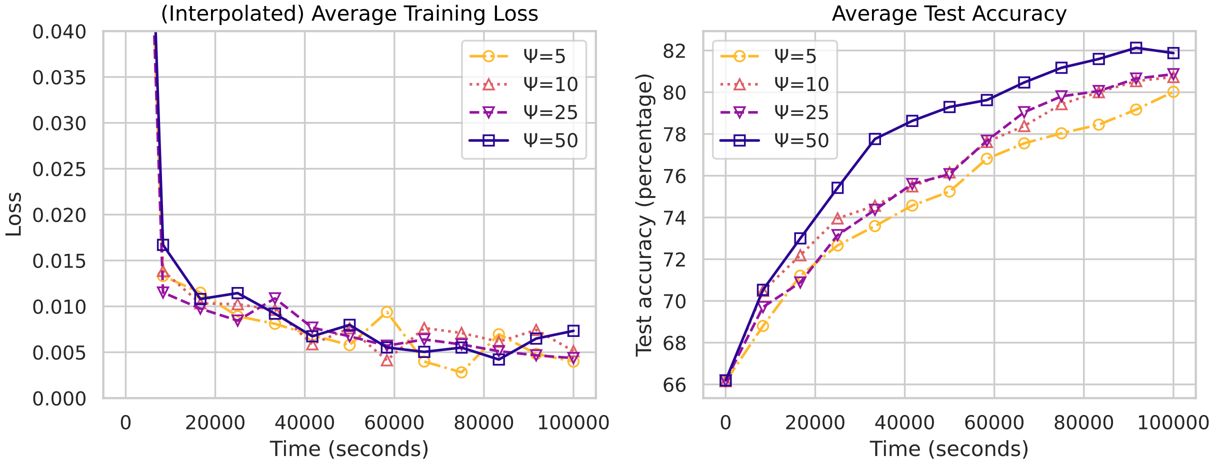


Figure B.4. Results with respect to different upper bounds on the number of received messages per user. ($P = 500$, $\Gamma_{\max} = 10$)

B.2.2 Dynamic connectivity and fading channels

In time-invariant $\mathcal{Q}$ cases, wireless channels can vary depending on the physical locations (coordinates) randomly generated. To avoid any bias by specific channel conditions and connectivity, we use the same coordinates for all simulation trials while changing only the value of the particular variable that impacts the performance. To minimize the influence of specific connectivity graphs in wireless networks, we repeat the experiments multiple times with different coordinates and draw the averaged results for each figure. Each experiment in this subsection is performed with $N = 15$ participants.

We vary the transmission duration deadline from 0.1 to 30 seconds in Fig. B.5. As $\Gamma_{\max}$ gets higher, the number of communication events increases since the number of stale updates that a user permits to receive also increases. This relaxation of acceptance for successful transmissions enhances the convergence rate by providing increased opportunities for message exchanges. Yet, the improvement saturates at the point around $\Gamma_{\max} = 10$, implying that stale updates that are included in model aggregation degrade the performance and lead to extended overall learning time.

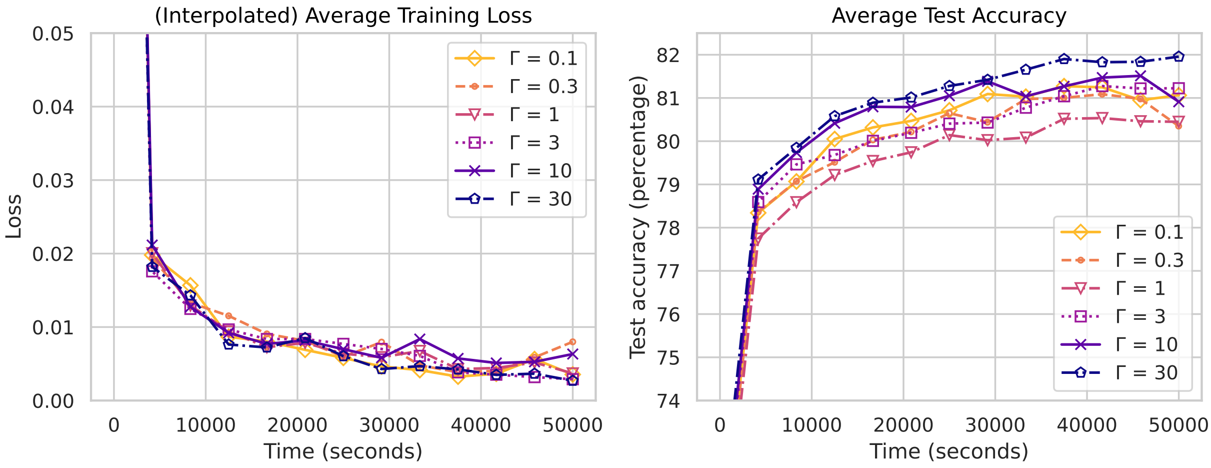


Figure B.5. Results with respect to different transmission duration deadline ($P = 50$, window $= 0$)

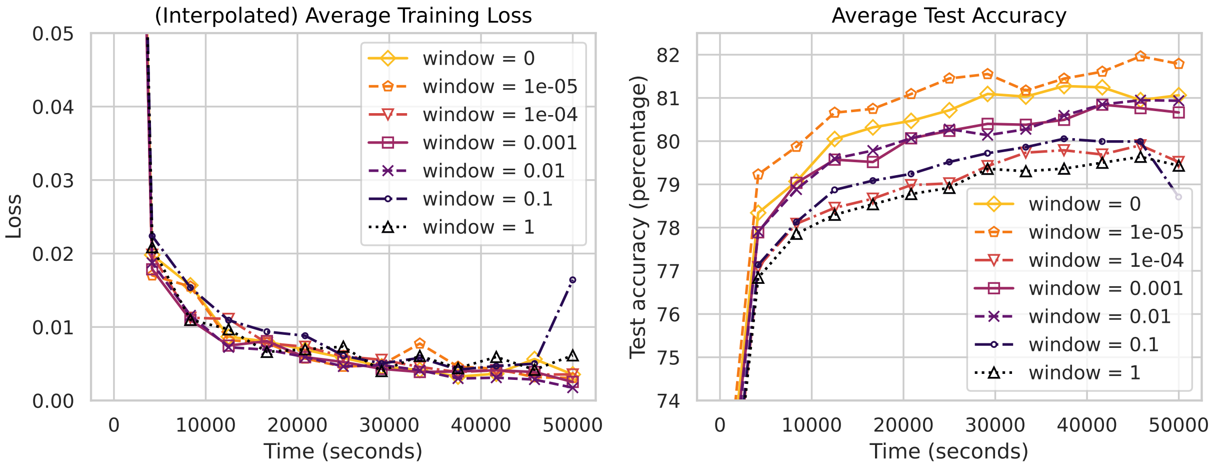


Figure B.6. Results with respect to different superposition windows. ($P = 50$, $\Gamma_{\max} = 1$)

We assess the performance of the proposed algorithm with different strictness in the superposition decision (superposition window), i.e., for how long each user lets the receiving session open. With a wider window, the participants renew their local reference models less frequently because they stack their neighbors' updates until the aggregation time comes. However, a wider window can also include stale gradients to the superposition phase, which can disturb the optimization direction of the reference models.

In Fig. B.7, we compare different lengths of unification periods. Even without controlling any other hyperparameter, $\psi_i(mP, (m+1)P)$ tends to increase as the network has longer P because a shorter unification period leads in general to more frequent reference model unification. Consequently, to remove the influence of bounding the number of incoming packets, we set a sufficiently large value for Ψ while looking at the impact of P on operating our algorithm. The results show that larger P is advantageous to get both high final accuracy and fast convergence. However, if the network topology (graph) is not vertex-transitive, the result can vary depending on which user becomes the hub. Meanwhile, smaller P implies more frequent broadcasting events, which come with increased communication overloads. In the case of asymmetric or

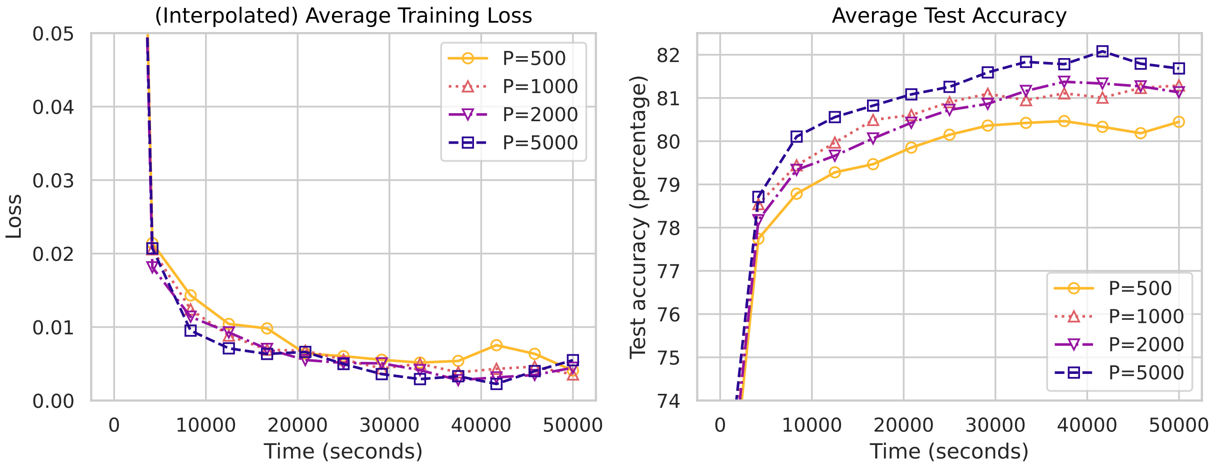


Figure B.7. Results with respect to different unification periods. ($\Gamma_{\max} = 1$, window $= 0$)

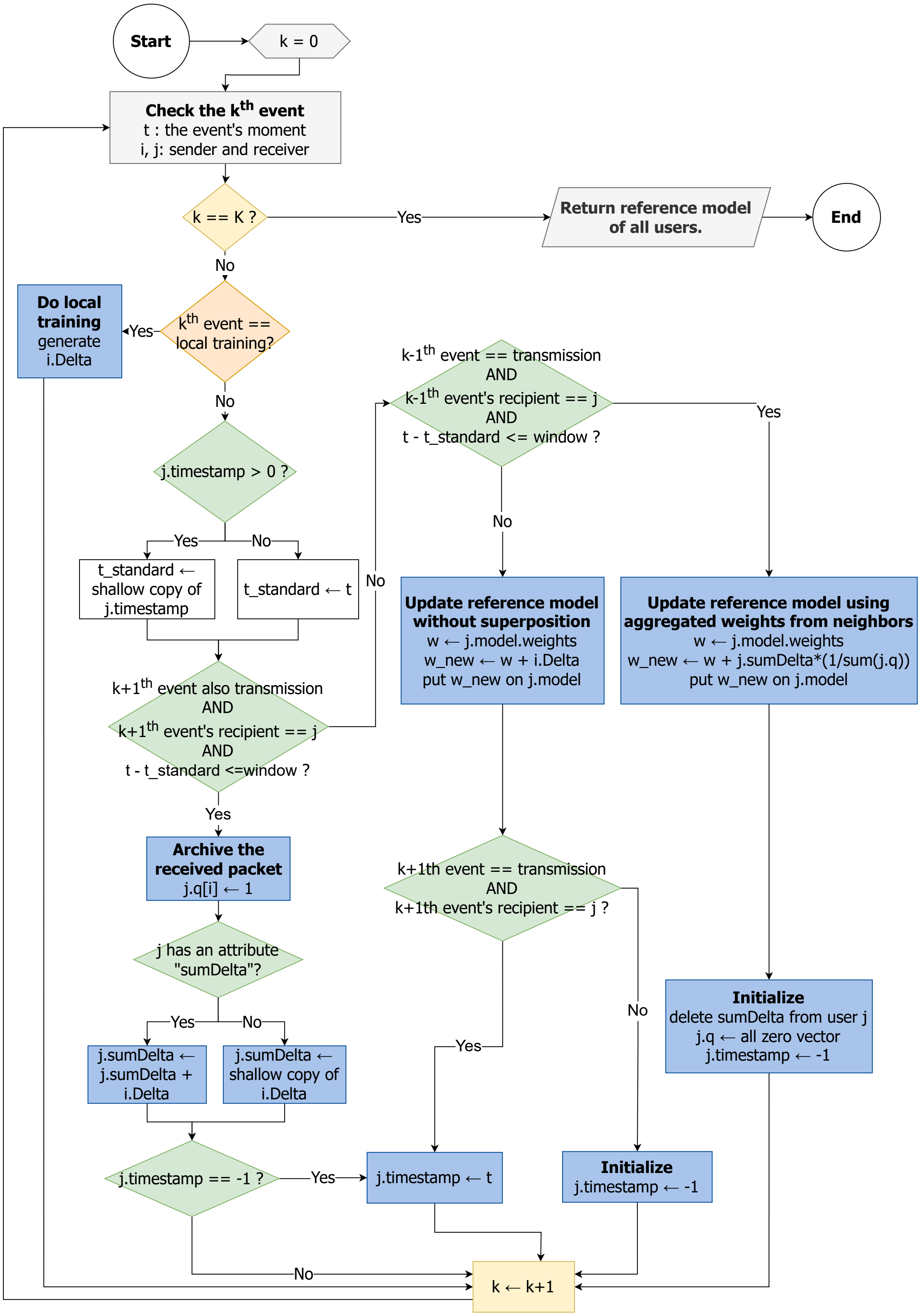
time-invariant topologies, the duration of the unification period should be chosen based on the communication resource budget. If a node with the most edges or the strongest channel is known or observable in a system, this node should take the role of a temporary hub, which will then broadcast its reference model.

B.3 Pseudo algorithm of DRACO

In this section, we provide a pseudo-algorithm and a flowchart for intuitive reproduction of source codes. The flowchart in Fig. B.8 includes only the transmission/reception procedure of DRACO, corresponding to line 21-39 (excluding periodic unification parts) of Algorithm 5.

Algorithm 5: Pseudo algorithm of Algorithm 3.

```
Input: $\eta, \lambda, \theta_0, B, T, P$
Output: $\{\theta_t : \forall t\}$
Initialize
    for $i = 1, \cdots, N$ do
        /* Generate ListEvents($i$)                                    */
        Generate $t \sim Exp(\lambda_i)$
        Append $[t, i]$ to ListEventsGrad($i$)
        for event in ListEventsGrad($i$) do
            for $j \in \mathcal{N}(i)$ do
                Generate $t \sim exp(\lambda_{ij})$ or $t \leftarrow$transmission delay
                Append $[t, j]$ to ListEventsComm($i$)
            end for
        end for
        ListEvents($i$) $\leftarrow$ ListEventsGrad($i$) + ListEventsComm($i$)
    end for
    /* Generate ListEvents over all clients                            */
    for $i = 1, \cdots, N$ do
        Stack ListEvents($i$) on ListEvents
    end for
    Sort ListEvents by $t$ in ascending order.
    Add the event indices $k$ in front of each element.
$K \leftarrow$ |ListEvents|
for $k = 1, \cdots, K$ do
    $(i, j) \leftarrow$ ListEvents($k$, 0), ListEvents($k$, 1)
    if $i == j$ then
        for $b = 0, \cdots, B-1$ do
            $\mathbf{y}_{b+1}^{(i)} \leftarrow \mathbf{y}_b^{(i)} - \eta g_i(\mathbf{y}_b^{(i)})$        // local batch training
        end for
        $\Delta_k^{(i)} \leftarrow \mathbf{y}_{k,B}^{(i)} - \theta_k^{(i)}$
    end if
    else
        for $j \in \mathcal{U} \setminus \{i\}$ do
            if event_code==“unification” then
                $\theta^{(j)} \leftarrow \tilde{\theta}^{(hub)}$
            end if
            else
                $\theta^{(j)} \leftarrow \theta^{(j)} + q_k^{i \to j} \tilde{\Delta}^{(i)}$                // aggregation
            end if
        end for
    end if
end for
if $t \equiv 0$ (mod P) and $t > 0$ and i is the hub then
    $\theta^{(hub)} \leftarrow \theta^{(i)}$
end if
return $\theta^{(i)}$ for $i \in \mathcal{U}$
```



Start
k = 0
Check the kth event
t : the event's moment
i, j: sender and receiver
k == K ?
Yes
Return reference model of all users.
End
No
kth event == local training?
Yes
Do local training
generate i.Delta
No
j.timestamp > 0 ?
Yes
No
t_standard ← shallow copy of j.timestamp
t_standard ← t
k+1th event also transmission AND k+1th event's recipient == j AND t - t_standard <=window ?
No
Yes
Archive the received packet
j.q[i] ← 1
j has an attribute "sumDelta"?
Yes
No
j.sumDelta ← j.sumDelta + i.Delta
j.sumDelta ← shallow copy of i.Delta
j.timestamp == -1 ?
Yes
No
j.timestamp ← t
k-1th event == transmission AND k-1th event's recipient == j AND t - t_standard <= window ?
Yes
No
Update reference model without superposition
w ← j.model.weights
w_new ← w + i.Delta
put w_new on j.model
Update reference model using aggregated weights from neighbors
w ← j.model.weights
w_new ← w + j.sumDelta*(1/sum(j.q))
put w_new on j.model
k+1th event == transmission AND k+1th event's recipient == j ?
Yes
No
Initialize
delete sumDelta from user j
j.q ← all zero vector
j.timestamp ← -1
Initialize
j.timestamp ← -1
k ← k+1

Figure B.8. Flowchart of DRACO after initialization

Bibliography

[1] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, *Federated learning: strategies for improving communication efficiency*, 2017. arXiv: 1610.05492 [cs.LG].

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data", in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, A. Singh and J. Zhu, Eds., ser. Proceedings of Machine Learning Research, vol. 54, PMLR, 2017, pp. 1273–1282.

[3] G. Neglia, C. Xu, D. Towsley, and G. Calbi, "Decentralized gradient methods: does topology matter?", in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, S. Chiappa and R. Calandra, Eds., ser. Proceedings of Machine Learning Research, vol. 108, PMLR, 2020, pp. 2348–2358.

[4] Z. Chen, M. Dahl, and E. G. Larsson, "Decentralized learning over wireless networks: the effect of broadcast with random access", in *2023 IEEE 24th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2023, pp. 316–320. DOI: 10.1109/SPAWC53906.2023.10304514.

[5] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized SGD with changing topology and local updates", in *International Conference on Machine Learning*, PMLR, 2020, pp. 5381–5393.

[6] V. Zantedeschi, A. Bellet, and M. Tommasi, "Fully decentralized joint learning of personalized models and collaboration graphs", in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, pp. 864–874.

[7] L. Yuan, Z. Wang, L. Sun, P. S. Yu, and C. G. Brinton, "Decentralized federated learning: a survey and perspective", *IEEE Internet of Things Journal*, pp. 1–1, 2024. DOI: 10.1109/JIOT.2024.3407584.

[8] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication", in *International Conference on Machine Learning*, PMLR, 2019, pp. 3478–3487.

[9] C. Hu, J. Jiang, and Z. Wang, *Decentralized federated learning: a segmented gossip approach*, 2019. arXiv: 1908.07782 [cs.LG].

[10] H. Cho, S. Mukherjee, D. Kim, T. Noh, and J. Lee, "Facing to wireless network densification in 6g: challenges and opportunities", *ICT Express*, vol. 9, no. 3, pp. 517–524, 2023, ISSN: 2405-9595. DOI: https://doi.org/10.1016/j.icte.2022.10.001.

[11] M. Al-Quraan, L. Mohjazi, L. Bariah, A. Centeno, A. Zoha, K. Arshad, K. Assaleh, S. Muhaidat, M. Debbah, and M. A. Imran, "Edge-native intelligence for 6g communications driven by federated learning: a survey of trends and challenges", *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 3, pp. 957–979, 2023. DOI: 10.1109/TETCI.2023.3251404.

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

[13] A. Karras, C. Karras, K. C. Giotopoulos, D. Tsolis, K. Oikonomou, and S. Sioutas, "Peer to peer federated learning: towards decentralized machine learning on edge devices", in *2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, 2022, pp. 1–9. DOI: 10.1109/SEEDA-CECNSM57760.2022.9932980.

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

[15] R. Pathak and M. J. Wainwright, "Fedsplit: an algorithmic framework for fast federated optimization", in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 7057–7066. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/4ebd440d99504722d80de606ea8507da-Paper.pdf.

[16] A. Taya, T. Nishio, M. Morikura, and K. Yamamoto, "Decentralized and model-free federated learning: consensus-based distillation in function space", *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 799–814, 2022. DOI: 10.1109/TSIPN.2022.3205549.

[17] L. Fang, P. Antsaklis, and A. Tzimas, "Asynchronous consensus protocols: preliminary results, simulations and open questions", in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 2194–2199. DOI: 10.1109/CDC.2005.1582487.

[18] C. Xie, S. Koyejo, and I. Gupta, *Asynchronous federated optimization*, 2020. arXiv: 1903.03934 [cs.DC].

[19] F. Wilhelmi, E. Guerra, and P. Dini, "On the decentralization of blockchain-enabled asynchronous federated learning", in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*, 2023, pp. 408–413. DOI: 10.1109/NetSoft57336.2023.10175411.

[20] C.-H. Hu, Z. Chen, and E. G. Larsson, "Scheduling and aggregation design for asynchronous federated learning over wireless networks", *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 4, pp. 874–886, 2023. DOI: 10.1109/JSAC.2023.3242719.

[21] M. Song, H. H. Yang, H. Shan, J. Lee, and T. Q. S. Quek, "Age of information in wireless networks: spatiotemporal analysis and locally adaptive power control", *IEEE Transactions on Mobile Computing*, vol. 22, no. 6, pp. 3123–3136, 2023. DOI: 10.1109/TMC.2021.3139666.

[22] B. Buyukates and S. Ulukus, “Timely communication in federated learning”, in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–6. DOI: 10.1109/INFOCOMWKSHPS51825.2021.9484497.

[23] H. H. Yang, A. Arafa, T. Q. S. Quek, and H. Vincent Poor, “Age-based scheduling policy for federated learning in mobile edge networks”, in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8743–8747. DOI: 10.1109/ICASSP40776.2020.9053740.

[24] S. Agarwal, H. Wang, S. Venkataraman, and D. Papailiopoulos, “On the utility of gradient compression in distributed training systems”, in *Proceedings of Machine Learning and Systems*, D. Marculescu, Y. Chi, and C. Wu, Eds., vol. 4, 2022, pp. 652–672. [Online]. Available: https://proceedings.mlsys.org/paper_files/paper/2022/file/773862fcc2e29f650d68960ba5bd1101-Paper.pdf.

[25] J. Wangni, J. Wang, J. Liu, and T. Zhang, “Gradient sparsification for communication-efficient distributed optimization”, in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/3328bdf9a4b9504b9398284244fe97c2-Paper.pdf.

[26] D. Grishchenko, F. Iutzeler, J. Malick, and M.-R. Amini, “Distributed learning with sparse communications by identification”, *SIAM Journal on Mathematics of Data Science*, vol. 3, no. 2, pp. 715–735, 2021. DOI: 10.1137/20M1347772. [Online]. Available: https://doi.org/10.1137/20M1347772.

[27] W. Xie, H. Li, J. Ma, Y. Li, J. Lei, D. Liu, and L. Fang, “JointSQ: joint sparsification-quantization for distributed learning”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 5778–5787.

[28] J. Sun, T. Chen, G. Giannakis, and Z. Yang, “Communication-efficient distributed learning via lazily aggregated quantized gradients”, in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/4e87337f366f72daa424dae
Paper.pdf.

[29] A. Danaee, R. C. de Lamare, and V. H. Nascimento, “Energy-efficient distributed learning with coarsely quantized signals”, *IEEE Signal Processing Letters*, vol. 28, pp. 329–333, 2021. DOI: 10.1109/LSP.2021.3051522.

[30] O. A. Hanna, Y. H. Ezzeldin, C. Fragouli, and S. Diggavi, *Quantizing data for distributed learning*, 2021. arXiv: 2012.07913 [cs.LG].

[31] N. Guha, A. Talwalkar, and V. Smith, *One-shot federated learning*, 2019. arXiv: 1902.11175 [cs.LG].

[32] S. Salehkaleybar, A. Sharifnassab, and S. J. Golestani, “One-shot federated learning: theoretical limits and algorithms to achieve them”, *Journal of Machine Learning Research*, vol. 22, no. 189, pp. 1–47, 2021.

[33] C. E. Heinbaugh, E. Luz-Ricca, and H. Shao, “Data-free one-shot federated learning under very high statistical heterogeneity”, in *The Eleventh International Conference on Learning Representations*, 2023.

[34] Y. Zhou, G. Pu, X. Ma, X. Li, and D. Wu, *Distilled one-shot federated learning*, 2021. arXiv: 2009.07999 [cs.LG].

[35] Y. Park, D.-J. Han, D.-Y. Kim, J. Seo, and J. Moon, “Few-round learning for federated learning”, in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 28 612–28 622.

[36] A. Rosato, M. Panella, E. Osipov, and D. Kleyko, “Few-shot federated learning in randomized neural networks via hyperdimensional computing”, in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–8. DOI: 10.1109/IJCNN55064.2022.9892007.

[37] J. M. B. da Silva, K. Ntougias, I. Krikidis, G. Fodor, and C. Fischione, “Simultaneous wireless information and power transfer for federated learning”, in *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2021, pp. 296–300. DOI: 10.1109/SPAWC51858.2021.9593160.

[38] A. Mahmoudi, H. S. Ghadikolaei, J. M. Barros Da Silva Júnior, and C. Fischione, “Fed-Cau: a proactive stop policy for communication and computation efficient federated learning”, *IEEE Transactions on Wireless Communications*, pp. 1–1, 2024. DOI: 10.1109/TWC.2024.3378351.

[39] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, “Terngrad: ternary gradients to reduce communication in distributed deep learning”, in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc., 2017.

[40] A. Panda, S. Mahloujifar, A. Nitin Bhagoji, S. Chakraborty, and P. Mittal, “SparseFed: mitigating model poisoning attacks in federated learning with sparsification”, in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, G. Camps-Valls, F. J. R. Ruiz, and I. Valera, Eds., ser. Proceedings of Machine Learning Research, vol. 151, PMLR, 2022, pp. 7587–7624.

[41] M. Beitollahi, M. Liu, and N. Lu, “Dsfl: dynamic sparsification for federated learning”, in *2022 5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, 2022, pp. 1–6. DOI: 10.1109/ICCSPA55860.2022.10019204.

[42] Y. Sun, S. Zhou, Z. Niu, and D. Gündüz, “Time-correlated sparsification for efficient over-the-air model aggregation in wireless federated learning”, in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 3388–3393. DOI: 10.1109/ICC45855.2022.9839279.

[43] K. Sun, H. Xu, K. Hua, X. Lin, G. Li, T. Jiang, and J. Li, “Joint top-k sparsification and shuffle model for communication-privacy-accuracy tradeoffs in federated-learning-based iov”, *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19 721–19 735, 2024. DOI: 10.1109/JIOT.2024.3370991.

[44] W. Xu, W. Fang, Y. Ding, M. Zou, and N. Xiong, “Accelerating federated learning for iot in big data analytics with pruning, quantization and selective updating”, *IEEE Access*, vol. 9, pp. 38 457–38 466, 2021. DOI: 10.1109/ACCESS.2021.3063291.

[45] P. Prakash, J. Ding, R. Chen, X. Qin, M. Shu, Q. Cui, Y. Guo, and M. Pan, “Iot device friendly and communication-efficient federated learning via joint model pruning and quantization”, *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 638–13 650, 2022. DOI: 10.1109/JIOT.2022.3145865.

[46] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, “Model pruning enables efficient federated learning on edge devices”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 10 374–10 386, 2023. DOI: 10.1109/TNNLS.2022.3166101.

[47] A. Albasyoni, M. Safaryan, L. Condat, and P. Richtárik, *Optimal gradient compression for distributed and federated learning*, 2020. arXiv: 2010.03246 [cs.LG].

[48] D. Leith and P. Clifford, “Convergence of distributed learning algorithms for optimal wireless channel allocation”, in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 2980–2985. DOI: 10.1109/CDC.2006.376821.

[49] T. Li, M. Sanjabi, A. Beirami, and V. Smith, *Fair resource allocation in federated learning*, 2020. arXiv: 1905.10497 [cs.LG].

[50] J. Xu, H. Wang, and L. Chen, “Bandwidth allocation for multiple federated learning services in wireless edge networks”, *IEEE Transactions on Wireless Communications*, vol. 21, no. 4, pp. 2534–2546, 2022. DOI: 10.1109/TWC.2021.3113346.

[51] E. Altman, G. Neglia, F. De Pellegrini, and D. Miorandi, “Decentralized stochastic control of delay tolerant networks”, in *IEEE INFOCOM 2009*, 2009, pp. 1134–1142. DOI: 10.1109/INFCOM.2009.5062026.

[52] E. Jeong, M. Zecchin, and M. Kountouris, “Asynchronous decentralized learning over unreliable wireless networks”, in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 607–612. DOI: 10.1109/ICC45855.2022.9838891.

[53] E. Jeong and M. Kountouris, *DRACO: a framework for decentralized asynchronous learning over continuous row-stochastic networks*, 2024.

[54] E. Jeong and M. Kountouris, “Personalized decentralized federated learning with knowledge distillation”, in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 1982–1987. DOI: 10.1109/ICC45041.2023.10279714.

[55] F. Yan, S. Sundaram, S. Vishwanathan, and Y. Qi, “Distributed autonomous online learning: regrets and intrinsic privacy-preserving properties”, *IEEE Trans. on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2483–2493, 2012.

[56] J. Tsitsiklis, D. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms”, *IEEE Trans. on Autom. Control*, vol. 31, no. 9, pp. 803–812, 1986.

[57] H. Xing, O. Simeone, and S. Bi, “Federated learning over wireless device-to-device networks: algorithms and convergence analysis”, *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3723–3741, 2021. DOI: 10.1109/JSAC.2021.3118400.

[58] E. Ozfatura, S. Rini, and D. Gündüz, "Decentralized SGD with over-the-air computation", in *IEEE Global Communications Conference*, 2020.

[59] Y. Shi, Y. Zhou, and Y. Shi, "Over-the-air decentralized federated learning", *arXiv preprint arXiv:2106.08011*, 2021.

[60] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: distributed stochastic gradient descent over-the-air", *IEEE Trans. on Signal Processing*, vol. 68, pp. 2155–2169, 2020.

[61] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, *Three approaches for personalization with applications to federated learning*, 2020. arXiv: 2002.10619 [cs.LG].

[62] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, *Improving federated learning personalization via model agnostic meta learning*, 2019. arXiv: 1909.12488 [cs.LG].

[63] S. Divi, H. Farrukh, and B. Celik, *Unifying distillation with personalization in federated learning*, 2021. arXiv: 2105.15191 [cs.LG].

[64] L. Xiao, S. Boyd, and S. Lall, "Distributed average consensus with time-varying Metropolis weights", *Automatica*, vol. 1, 2006.

[65] A. Koloskova*, T. Lin*, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression", in *ICLR 2020 - International Conference on Learning Representations*, 2020.

[66] S. Dutta, J. Wang, and G. Joshi, "Slow and stale gradients can win the race", *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 1012–1024, 2021.

[67] H. Xiao, K. Rasul, and R. Vollgraf. "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms". arXiv: cs.LG/1708.07747 [cs.LG]. (2017).

[68] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization", *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009. DOI: 10.1109/TAC.2008.2009515.

[69] A. Lalitha, S. Shekhar, T. Javidi, and F. Koushanfar, "Fully decentralized federated learning", in *Third Workshop on bayesian deep learning (in Conjunction with NeurIPS 2018)*, vol. 2, 2018.

[70] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, *Braintorrent: a peer-to-peer environment for decentralized federated learning*, 2019. arXiv: 1905.06731 [cs.LG].

[71] T. Qin, S. R. Etesami, and C. A. Uribe, "Decentralized federated learning for overparameterized models", in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 5200–5205. DOI: 10.1109/CDC51059.2022.9992924.

[72] G. Nadiradze, A. Sabour, P. Davies, S. Li, and D. Alistarh, "Asynchronous decentralized sgd with quantized and local updates", in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 6829–6842.

[73] R. Dai, L. Shen, F. He, X. Tian, and D. Tao, “DisPFL: towards communication-efficient personalized federated learning via decentralized sparse training”, in *Proceedings of the 39th International Conference on Machine Learning*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., ser. Proceedings of Machine Learning Research, vol. 162, PMLR, 2022, pp. 4587–4604.

[74] Y. Esfandiari, S. Y. Tan, Z. Jiang, A. Balu, E. Herron, C. Hegde, and S. Sarkar, “Cross-gradient aggregation for decentralized learning from non-iid data”, in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 3036–3046.

[75] J. Jiang, W. Zhang, J. GU, and W. Zhu, “Asynchronous decentralized online learning”, in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 20 185–20 196.

[76] X. Liang, A. M. Javid, M. Skoglund, and S. Chatterjee, “Asynchrounous decentralized learning of a neural network”, in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3947–3951. DOI: 10.1109/ICASSP40776.2020.9053996.

[77] Y. Kanamori, Y. Yamasaki, S. Hosoai, H. Nakamura, and H. Takase, “An asynchronous federated learning focusing on updated models for decentralized systems with a practical framework”, in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2023, pp. 1147–1154. DOI: 10.1109/COMPSAC57700.2023.00173.

[78] E. T. Martínez Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, “Decentralized federated learning: fundamentals, state of the art, frameworks, trends, and challenges”, *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2983–3013, 2023. DOI: 10.1109/COMST.2023.3315746.

[79] X. Zhang, X. Zhu, W. Bao, L. T. Yang, J. Wang, H. Yan, and H. Chen, “Distributed learning on mobile devices: a new approach to data mining in the internet of things”, *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 264–10 279, 2021. DOI: 10.1109/JIOT.2020.3030783.

[80] I. Hegedűs, G. Danner, and M. Jelasity, “Decentralized learning works: an empirical comparison of gossip learning and federated learning”, *Journal of Parallel and Distributed Computing*, vol. 148, pp. 109–124, 2021, ISSN: 0743-7315. DOI: 10.1016/j.jpdc.2020.10.006.

[81] L. Wulfert, N. Asadi, W.-Y. Chung, C. Wiede, and A. Grabmaier, “Adaptive decentralized federated gossip learning for resource-constrained iot devices”, in *Proceedings of the 4th International Workshop on Distributed Machine Learning*, ser. DistributedML ’23, Paris, France: Association for Computing Machinery, 2023, 27–33, ISBN: 9798400704475. DOI: 10.1145/3630048.3630181.

[82] M. Even, A. Koloskova, and L. Massoulié, “Asynchronous SGD on graphs: a unified framework for asynchronous decentralized and federated optimization”, in *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, S. Dasgupta, S. Mandt, and Y. Li, Eds., ser. Proceedings of Machine Learning Research, vol. 238, PMLR, 2024, pp. 64–72.

[83] M. Blot, D. Picard, M. Cord, and N. Thome, *Gossip training for deep learning*, 2016. arXiv: `1611.09726 [cs.CV]`.

[84] D. T. A. Nguyen, D. T. Nguyen, and A. Nedić, “Accelerated *AB*/push-pull methods for distributed optimization over time-varying directed networks”, *IEEE Transactions on Control of Network Systems*, pp. 1–12, 2023. DOI: `10.1109/TCNS.2023.3338236`.

[85] R. Xin, C. Xi, and U. A. Khan, “FROST—fast row-stochastic optimization with uncoordinated step-sizes”, *EURASIP Journal on Advances in Signal Processing*, vol. 2019, pp. 1–14, 2019.

[86] A. Nedić and A. Olshevsky, “Distributed optimization over time-varying directed graphs”, *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015. DOI: `10.1109/TAC.2014.2364096`.

[87] M. Akbari, B. Gharesifard, and T. Linder, “Distributed online convex optimization on time-varying directed graphs”, *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 417–428, 2017. DOI: `10.1109/TCNS.2015.2505149`.

[88] Z. Li, Z. Ding, J. Sun, and Z. Li, “Distributed adaptive convex optimization on directed graphs via continuous-time algorithms”, *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1434–1441, 2018. DOI: `10.1109/TAC.2017.2750103`.

[89] D. Ghaderyan, N. S. Aybat, A. P. Aguiar, and F. L. Pereira, “A fast row-stochastic decentralized method for distributed optimization over directed graphs”, *IEEE Transactions on Automatic Control*, vol. 69, no. 1, pp. 275–289, 2024. DOI: `10.1109/TAC.2023.3275927`.

[90] V. S. Mai and E. H. Abed, “Distributed optimization over weighted directed graphs using row stochastic matrix”, in *2016 American Control Conference (ACC)*, 2016, pp. 7165–7170. DOI: `10.1109/ACC.2016.7526803`.

[91] D. T. A. Nguyen, S. Wang, D. T. Nguyen, A. Nedich, and H. V. Poor, *Decentralized federated learning with gradient tracking over time-varying directed networks*, 2024. arXiv: `2409.17189 [math.OC]`. [Online]. Available: `https://arxiv.org/abs/2409.17189`.

[92] Y. Liu, Y. Shi, Q. Li, B. Wu, X. Wang, and L. Shen, “Decentralized directed collaboration for personalized federated learning”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 23 168–23 178.

[93] Z. Song, W. Li, K. Jin, L. Shi, M. Yan, W. Yin, and K. Yuan, “Communication-efficient topologies for decentralized learning with O(1) consensus rate”, in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35, Curran Associates, Inc., 2022, pp. 1073–1085.

[94] S. Wang and M. Ji, “A unified analysis of federated learning with arbitrary client participation”, in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35, Curran Associates, Inc., 2022, pp. 19 124–19 137.

[95] D. Jhunjhunwala, P. Sharma, A. Nagarkatti, and G. Joshi, “Fedvarp: tackling the variance due to partial client participation in federated learning”, in *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, J. Cussens and K. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 180, PMLR, 2022, pp. 906–916.

[96] B. Li, M. N. Schmidt, T. S. Alstrøm, and S. U. Stich, “On the effectiveness of partial variance reduction in federated learning with heterogeneous data”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 3964–3973.

[97] T. Qin, J. Yevale, and S. R. Etesami, “Communication-efficient local sgd for over-parametrized models with partial participation”, in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 2098–2103. DOI: 10.1109/CDC49753.2023.10383908.

[98] Y. J. Cho, P. Sharma, G. Joshi, Z. Xu, S. Kale, and T. Zhang, “On the convergence of federated averaging with cyclic client participation”, in *Proceedings of the 40th International Conference on Machine Learning*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., ser. Proceedings of Machine Learning Research, vol. 202, PMLR, 2023, pp. 5677–5721.

[99] M. Even, H. Hendrikx, and L. Massoulié, “Asynchronous speedup in decentralized optimization”, in *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

[100] M. G. Rabbat and K. I. Tsianos, “Asynchronous decentralized optimization in heterogeneous systems”, in *53rd IEEE Conference on Decision and Control*, 2014, pp. 1125–1130. DOI: 10.1109/CDC.2014.7039532.

[101] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, “Push-sum distributed dual averaging for convex optimization”, in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 5453–5458. DOI: 10.1109/CDC.2012.6426375.

[102] H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, “Quantized decentralized stochastic learning over directed graphs”, in *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, Eds., ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 9324–9333.

[103] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, “Push-saga: a decentralized stochastic algorithm with variance reduction over directed graphs”, *IEEE Control Systems Letters*, vol. 6, pp. 1202–1207, 2022. DOI: 10.1109/LCSYS.2021.3090652.

[104] M. T. Toghani, S. Lee, and C. A. Uribe, “Pars-push: personalized, asynchronous and robust decentralized optimization”, *IEEE Control Systems Letters*, vol. 7, pp. 361–366, 2023. DOI: 10.1109/LCSYS.2022.3189317.

[105] M. Assran, N. Loizou, N. Ballas, and M. Rabbat, “Stochastic gradient push for distributed deep learning”, in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 344–353.

[106] M. S. Assran and M. G. Rabbat, “Asynchronous gradient push”, *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 168–183, 2021. DOI: 10.1109/TAC.2020.2981035.

[107] A. Nedić and A. Olshevsky, “Stochastic gradient-push for strongly convex functions on time-varying directed graphs”, *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3936–3947, 2016. DOI: 10.1109/TAC.2016.2529285.

[108] Y.-G. Hsieh, Y. Laguel, F. Iutzeler, and J. Malick, “Push–pull with device sampling”, *IEEE Transactions on Automatic Control*, vol. 68, no. 12, pp. 7179–7194, 2023. DOI: 10.1109/TAC.2023.3250339.

[109] X. Mao, K. Yuan, Y. Hu, Y. Gu, A. H. Sayed, and W. Yin, “Walkman: a communication-efficient random-walk algorithm for decentralized optimization”, *IEEE Transactions on Signal Processing*, vol. 68, pp. 2513–2528, 2020. DOI: 10.1109/TSP.2020.2983167.

[110] G. Ayache and S. El Rouayheb, “Random walk gradient descent for decentralized learning on graphs”, in *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2019, pp. 926–931. DOI: 10.1109/IPDPSW.2019.00157.

[111] H. Hendrikx, “A principled framework for the design and analysis of token algorithms”, in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, F. Ruiz, J. Dy, and J.-W. van de Meent, Eds., ser. Proceedings of Machine Learning Research, vol. 206, PMLR, 2023, pp. 470–489.

[112] A. Nedić, “Distributed gradient methods for convex machine learning problems in networks: distributed optimization”, *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 92–101, 2020. DOI: 10.1109/MSP.2020.2975210.

[113] L. Giaretta and v. Girdzijauskas, “Gossip learning: off the beaten path”, in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 1117–1124. DOI: 10.1109/BigData47090.2019.9006216.

[114] P. Gholami and H. Seferoglu, “Digest: fast and communication efficient decentralized learning with local updates”, *IEEE Transactions on Machine Learning in Communications and Networking*, pp. 1–1, 2024. DOI: 10.1109/TMLCN.2024.3354236.

[115] A. Nabli and E. Oyallon, “DADAO: decoupled accelerated decentralized asynchronous optimization”, in *International Conference on Machine Learning*, PMLR, 2023, pp. 25 604–25 626.

[116] A. Nabli, E. Belilovsky, and E. Oyallon, “**A^{2}CiD^2**: accelerating asynchronous communication in decentralized deep learning”, in *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36, Curran Associates, Inc., 2023, pp. 47 451–47 474.

[117] E. Belilovsky, M. Eickenberg, and E. Oyallon, “Decoupled greedy learning of CNNs”, in *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, Eds., ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 736–745.

[118] J. Postel, *Rfc0793: transmission control protocol*, 1981.

[119] J. F. C. Kingman, *Poisson processes*. Clarendon Press, 1992, vol. 3.

[120] A. Hashemi, A. Acharya, R. Das, H. Vikalo, S. Sanghavi, and I. Dhillon, *On the benefits of multiple gossip steps in communication-constrained decentralized optimization*, 2020. arXiv: 2011.10643 [cs.LG].

[121] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, “EMNIST: extending mnist to handwritten letters”, in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2921–2926. DOI: 10.1109/IJCNN.2017.7966217.

[122] R. Cattral and F. Oppacher, *Poker Hand*, UCI Machine Learning Repository, DOI: https://doi.org/10. 2002.

[123] M. Salehi and E. Hossain, “Federated learning in unreliable and resource-constrained cellular wireless networks”, *IEEE Transactions on Communications*, vol. 69, no. 8, pp. 5136–5151, 2021. DOI: 10.1109/TCOMM.2021.3081746.

[124] H. Xie, M. Xia, P. Wu, S. Wang, and K. Huang, “Decentralized federated learning with asynchronous parameter sharing for large-scale iot networks”, *IEEE Internet of Things Journal*, pp. 1–1, 2024. DOI: 10.1109/JIOT.2024.3354869.

[125] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, “A survey on federated learning for resource-constrained iot devices”, *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 1–24, 2021.

[126] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, “Inverting gradients - how easy is it to break privacy in federated learning?”, *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 937–16 947, 2020.

[127] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, “Robust and communication-efficient federated learning from non-iid data”, *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3400–3413, 2019.

[128] S. Niknam, H. S. Dhillon, and J. H. Reed, “Federated learning for wireless communications: motivation, opportunities, and challenges”, *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[129] O. Marfoq, G. Neglia, R. Vidal, and L. Kameni, “Personalized federated learning through local memorization”, in *Proceedings of the 39th International Conference on Machine Learning*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., ser. Proceedings of Machine Learning Research, vol. 162, PMLR, 2022, pp. 15 070–15 092.

[130] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, “Personalized cross-silo federated learning on non-iid data”, in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 7865–7873.

[131] A. Fallah, A. Mokhtari, and A. Ozdaglar, *Personalized federated learning: a meta-learning approach*, 2020. arXiv: 2002.07948 [cs.LG].

[132] S. Wu, T. Li, Z. Charles, Y. Xiao, K. Liu, Z. Xu, and V. Smith, “Motley: benchmarking heterogeneity and personalization in federated learning”, in *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

[133] P. Vanhaesebrouck, A. Bellet, and M. Tommasi, “Decentralized collaborative learning of personalized models over networks”, in *Artificial Intelligence and Statistics*, PMLR, 2017, pp. 509–517.

[134] G. Ye, T. Chen, Y. Li, L. Cui, Q. V. H. Nguyen, and H. Yin, “Heterogeneous collaborative learning for personalized healthcare analytics via messenger distillation”, *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 11, pp. 5249–5259, 2023. DOI: 10.1109/JBHI.2023.3247463.

[135] S. Li, T. Zhou, X. Tian, and D. Tao, “Learning to collaborate in decentralized learning of personalized models”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9766–9775.

[136] E. Borodich, A. Beznosikov, A. Sadiev, V. Sushko, N. Savelyev, M. Takáč, and A. Gasnikov, *Decentralized personalized federated learning for min-max problems*, 2021. arXiv: 2106.07289 [cs.LG].

[137] A. Sadiev, E. Borodich, A. Beznosikov, D. Dvinskikh, S. Chezhegov, R. Tappenden, M. Takáč, and A. Gasnikov, “Decentralized personalized federated learning: lower bounds and optimal algorithm for all personalization modes”, *EURO Journal on Computational Optimization*, vol. 10, p. 100 041, 2022.

[138] A. Bellet, R. Guerraoui, M. Taziki, and M. Tommasi, “Personalized and private peer-to-peer machine learning”, in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2018, pp. 473–481.

[139] O. Marfoq, G. Neglia, A. Bellet, L. Kameni, and R. Vidal, “Federated multi-task learning under a mixture of distributions”, in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 15 434–15 447.

[140] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, “Communication-efficient on-device machine learning: federated distillation and augmentation under non-iid private data”, in *Workshop on Machine Learning on the Phone and other Consumer Devices (in Conjunction with NeurIPS 2018)*, 2018.

[141] G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*, 2015. arXiv: 1503.02531 [stat.ML].

[142] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton, *Large scale distributed neural network training through online distillation*, 2020. arXiv: 1804.03235 [cs.LG].

[143] A. Nichol, J. Achiam, and J. Schulman, *On first-order meta-learning algorithms*, 2018. arXiv: 1803.02999 [cs.LG].

[144] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, *Federated learning with matched averaging*, 2020. arXiv: 2002.06440 [cs.LG].

[145] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, “On the convergence of fedavg on non-iid data”, in *International Conference on Learning Representations*, 2020.

[146] B. Le Bars, A. Bellet, M. Tommasi, E. Lavoie, and A. Kermarrec, “Refined convergence and topology learning for decentralized optimization with heterogeneous data”, in *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

[147] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, “Federated multi-task learning”, in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc., 2017.

[148] M. Karnaugh, “The map method for synthesis of combinational logic circuits”, *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, vol. 72, no. 5, pp. 593–599, 1953. DOI: 10.1109/TCE.1953.6371932.

[149] Y. LeCun, C. Cortes, and C. Burges, “Mnist handwritten digit database”, *ATT Labs*, vol. 2, 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist.